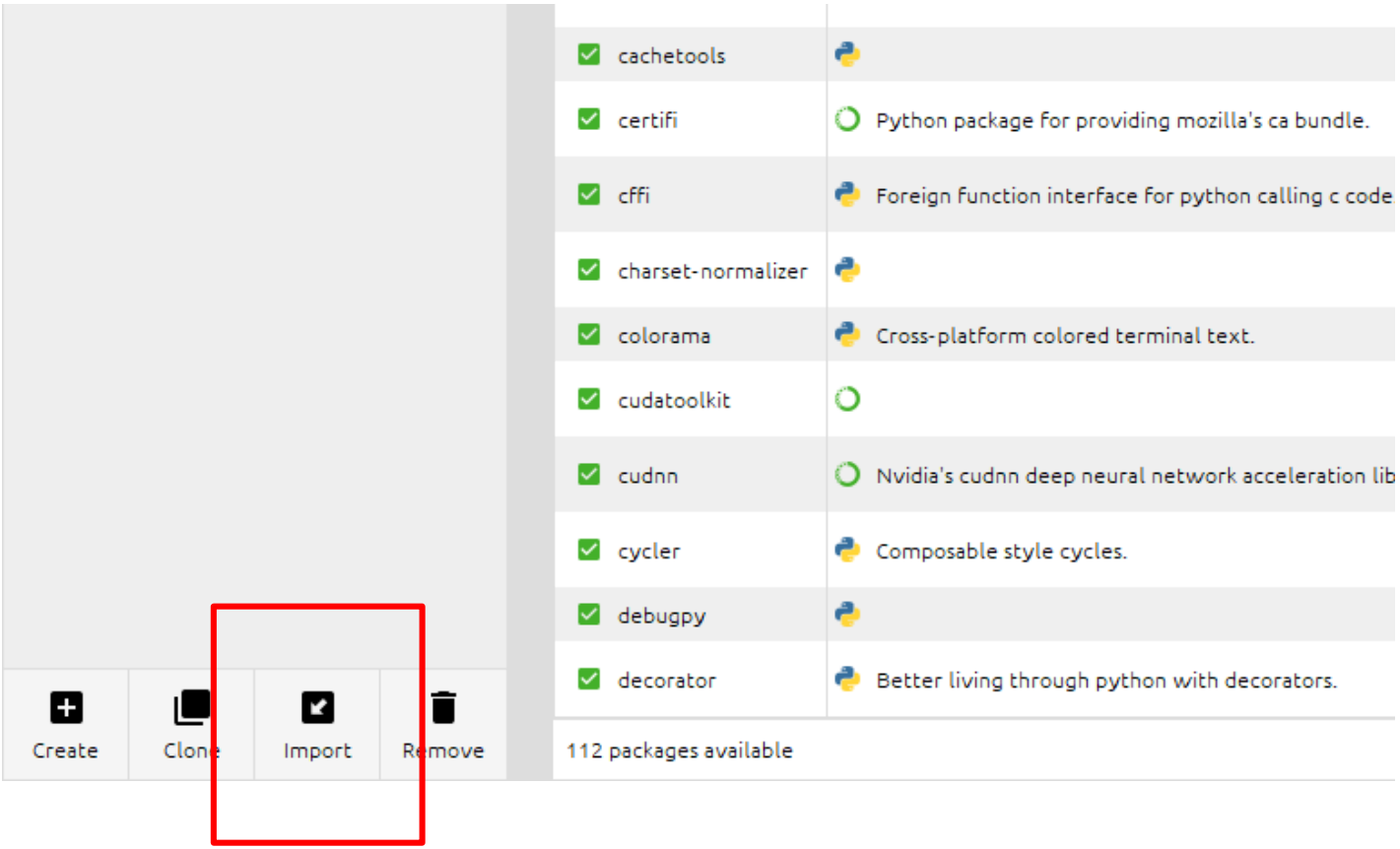
LAB4-Lenet、EdgeTPU

國立雲林科技大學
夏世昌 特聘教授 / 電子工程系
王斯弘 助理教授 / 前瞻學位學士學程

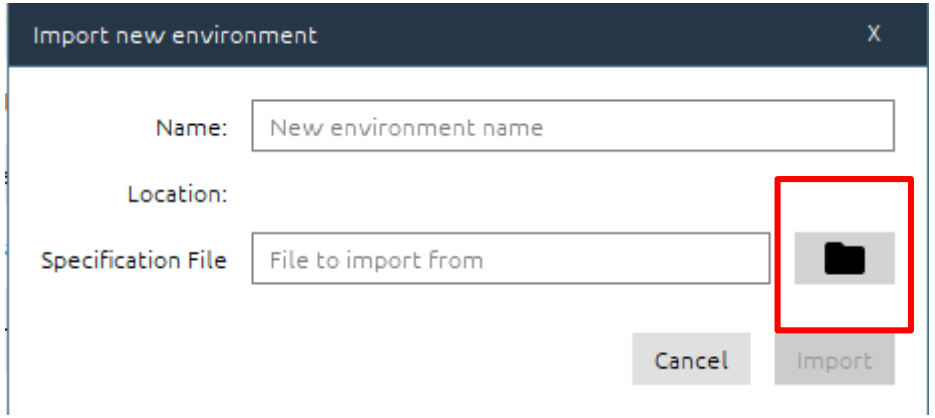
2022, Fall Semester

Import library



- 匯入環境

Import library



lenet.yml

- 匯入環境，至雲端硬碟下載

Import library

```
import tensorflow as tf
import numpy as np

tf.__version__
```

```
'1.15.0'
```

- cudnn==7.6.5
- cudatoolkit==10.0.130

參數設定

```
num_class = 10
batch_size = 2048
epochs = 100
iterations = 30
```

iterations

- iterations（迭代）：每一次迭代都是一次權重更新，每一 次權重更新需要batch_size個數據進行Forward運算得到損 失函式，再BP演算法更新引數。1個iteration等於使用 batchsize個樣本訓練一次。

- 具體的計算公式為：
  one **epoch** = numbers of **iterations** = N = 訓練樣本的數 量 **/batch_size**

dataset

```
(x_train, y_train), (x_test, y_test) = tf.keras.datasets.mnist.load_data()
```

Loads the MNIST dataset

https://www.tensorflow.org/api_docs/python/tf/keras/datasets/mnist/load_data

資料處理

```
x_train = x_train / 255
x_test = x_test / 255
x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], x_train.shape[2], 1))  #(60000,28,28,1)
x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], x_test.shape[2], 1))  #(10000, 28, 28, 1)
y_train = tf.keras.utils.to_categorical(y_train, 10)
y_test = tf.keras.utils.to_categorical(y_test, 10)
```

函式

- **astype()**

對資料類別進行轉換

- **np.reshape()**

將資料原來的尺寸更改為我們想要的尺寸

定義convolution

```
def conv(x, filters, size):
    return tf.keras.layers.Conv2D(filters=filters, kernel_size=size)(x)
```

函式

- **tf.keras.layers.Conv2D()**

2D convolution layer (e.g. spatial convolution over images)

https://www.tensorflow.org/api_docs/python/tf/keras/layers/C onv2D

定義pooling層

```
def maxpooling(x):
    return tf.keras.layers.MaxPooling2D(padding='same', strides=2)(x)
```

函式

- **tf.keras.layers.MaxPooling2D**

Max pooling operation for 2D spatial data.

https://www.tensorflow.org/api_docs/python/tf/keras/layers/MaxPool2D

定義
Model

```
def lenet(x):
    x = conv(x, 6, (5, 5))
    x = maxpooling(x)
    x = conv(x, 16, (5, 5))
    x = maxpooling(x)
    x = tf.keras.layers.Flatten()(x)
    x = tf.keras.layers.Dense(120)(x)
    x = tf.keras.layers.Dense(84)(x)
    x = tf.keras.layers.Dense(10, activation='softmax')(x)
    return x
```

函式

- **tf.keras.layers.Flatten**

Flattens the input. Does not affect the batch size.

https://www.tensorflow.org/api_docs/python/tf/keras/layers/Flatten

定義Model架構

```
img_input = tf.keras.Input(shape=(28,28,1))
output = lenet(img_input)
model = tf.keras.Model(img_input,output)
```

可視化Model架構

```
model.summary()
```

```
Model: "functional_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 28, 28, 1)]       0
_________________________________________________________________
conv2d (Conv2D)              (None, 24, 24, 6)         156
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 12, 12, 6)         0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 8, 8, 16)          2416
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 4, 4, 16)          0
_________________________________________________________________
flatten (Flatten)            (None, 256)               0
_________________________________________________________________
dense (Dense)                (None, 120)               30840
_________________________________________________________________
dense_1 (Dense)              (None, 84)                10164
_________________________________________________________________
dense_2 (Dense)              (None, 10)                850
=================================================================
Total params: 44,426
Trainable params: 44,426
Non-trainable params: 0
_________________________________________________________________
```

LOSS與優化器

```
sgd = tf.keras.optimizers.SGD(learning_rate=0.01)
model.compile(optimizer=sgd, loss='categorical_crossentropy', metrics=['acc'])
```

函式

- **tf.keras.optimizers.SGD**

Gradient descent (with momentum) optimizer.

https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/SGD

訓練

```
history = model.fit(x=x_train,y=y_train,batch_size=batch_size,epochs=epochs,steps_per_epoch=iterations,validation_data=(x_test, y_test))
```

成果

```
Epoch 91/100
30/30 [==============================] - 0s 10ms/step - loss: 0.1116 - accuracy: 0.9666 - val_loss: 0.1056 - val_accuracy: 0.9670
Epoch 92/100
30/30 [==============================] - 0s 8ms/step - loss: 0.1107 - accuracy: 0.9666 - val_loss: 0.1052 - val_accuracy: 0.9681
Epoch 93/100
30/30 [==============================] - 0s 10ms/step - loss: 0.1102 - accuracy: 0.9672 - val_loss: 0.1059 - val_accuracy: 0.9680
Epoch 94/100
30/30 [==============================] - 0s 9ms/step - loss: 0.1093 - accuracy: 0.9670 - val_loss: 0.1051 - val_accuracy: 0.9681
Epoch 95/100
30/30 [==============================] - 0s 9ms/step - loss: 0.1087 - accuracy: 0.9677 - val_loss: 0.1026 - val_accuracy: 0.9687
Epoch 96/100
30/30 [==============================] - 0s 9ms/step - loss: 0.1077 - accuracy: 0.9680 - val_loss: 0.1022 - val_accuracy: 0.9700
Epoch 97/100
30/30 [==============================] - 0s 9ms/step - loss: 0.1072 - accuracy: 0.9682 - val_loss: 0.1011 - val_accuracy: 0.9681
Epoch 98/100
30/30 [==============================] - 0s 9ms/step - loss: 0.1066 - accuracy: 0.9681 - val_loss: 0.1027 - val_accuracy: 0.9678
Epoch 99/100
30/30 [==============================] - 0s 9ms/step - loss: 0.1056 - accuracy: 0.9683 - val_loss: 0.1017 - val_accuracy: 0.9692
Epoch 100/100
30/30 [==============================] - 0s 9ms/step - loss: 0.1050 - accuracy: 0.9682 - val_loss: 0.0981 - val_accuracy: 0.9703
```

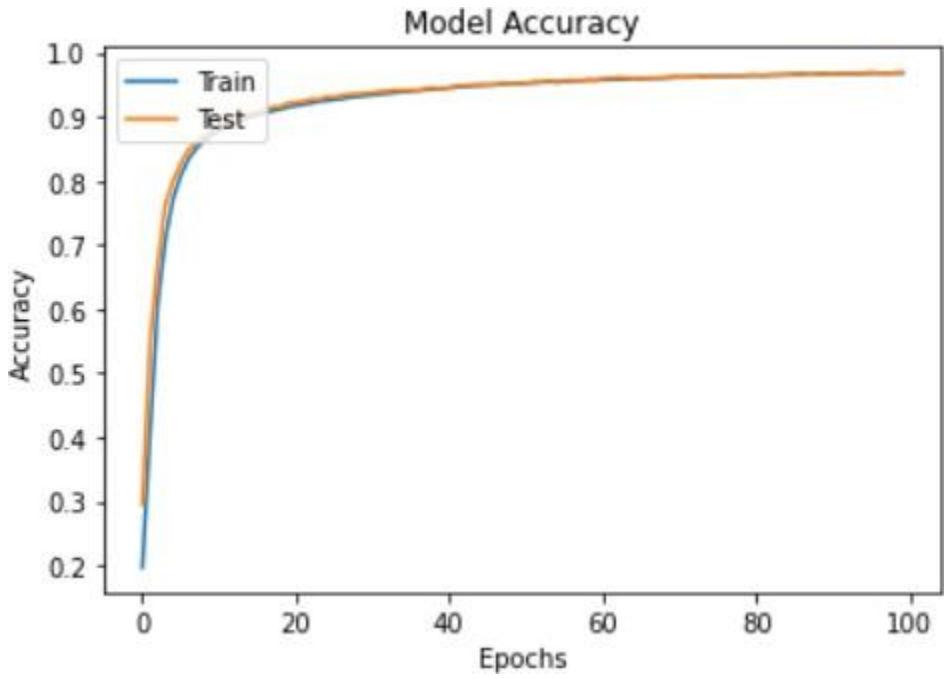
成果圖

```
import matplotlib.pyplot as plt

plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('Model Accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epochs')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()

plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model Accuracy')
plt.ylabel('Loss')
plt.xlabel('Epochs')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```

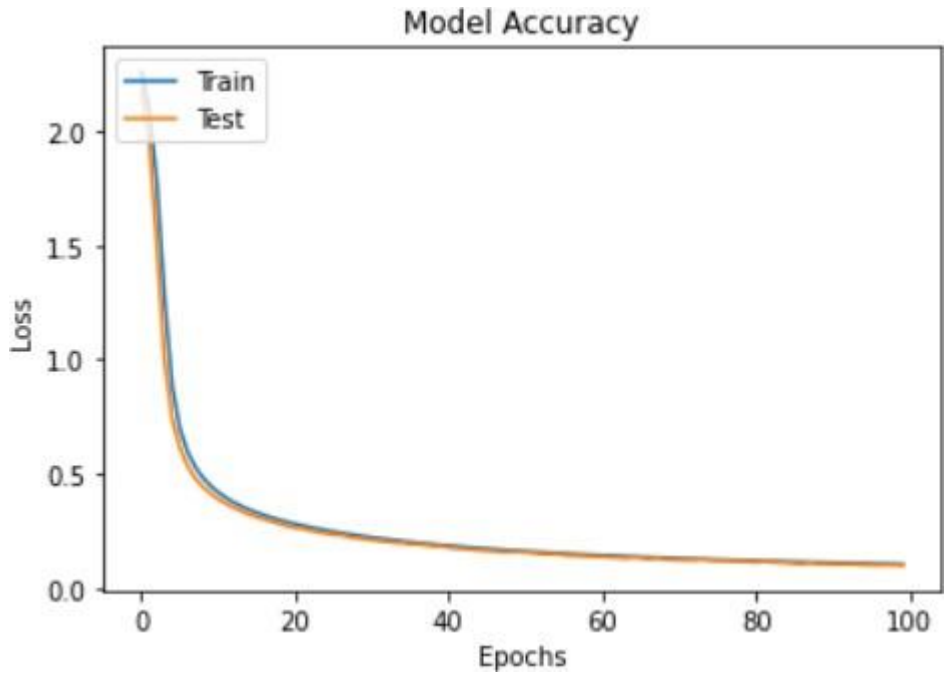
成果圖


Model Accuracy
Train
Test
Accuracy
Epochs
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0
20
40
60
80
100

成果圖

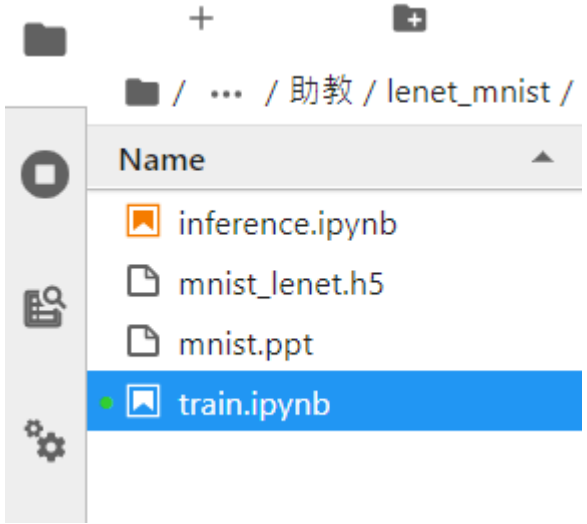
```
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```

成果圖


Model Accuracy
Train
Test
Loss
2.0
1.5
1.0
0.5
0.0
0
20
40
60
80
100
Epochs

儲存model

```
model.save('./mnist_lenet.h5')
```



預測

import

```
import tensorflow as tf
tf.__version__
```

```
'2.0.0'
```

```
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
```

```
import tensorflow as tf
import numpy as np

tf.__version__
```

```
'1.15.0'
```

```
(lenet) C:\Users\es402>pip install pillow
```

```
(lenet) C:\Users\es402>pip install matplotlib
```

PIL(Pillow)

The Python Imaging Library adds image processing capabilities to your Python interpreter.

This library provides extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities.

The core image library is designed for fast access to data stored in a few basic pixel formats. It should provide a solid foundation for a general image processing tool.

https://pypi.org/project/Pillow/

預測

載入model

```
model = tf.keras.models.load_model('./mnist_lenet.h5')
```

如果有錯執行:

```
(lenet) C:\Users\es402>pip install h5py==2.10
```

函式

- **tf.keras.models.load_model()**

Loads a model saved via model.save().

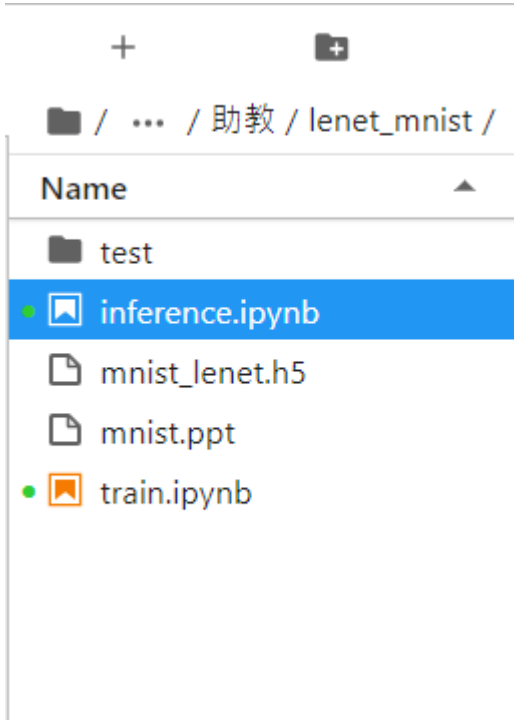
https://www.tensorflow.org/api_docs/python/tf/keras/models/load_model

預測

載入預測圖片

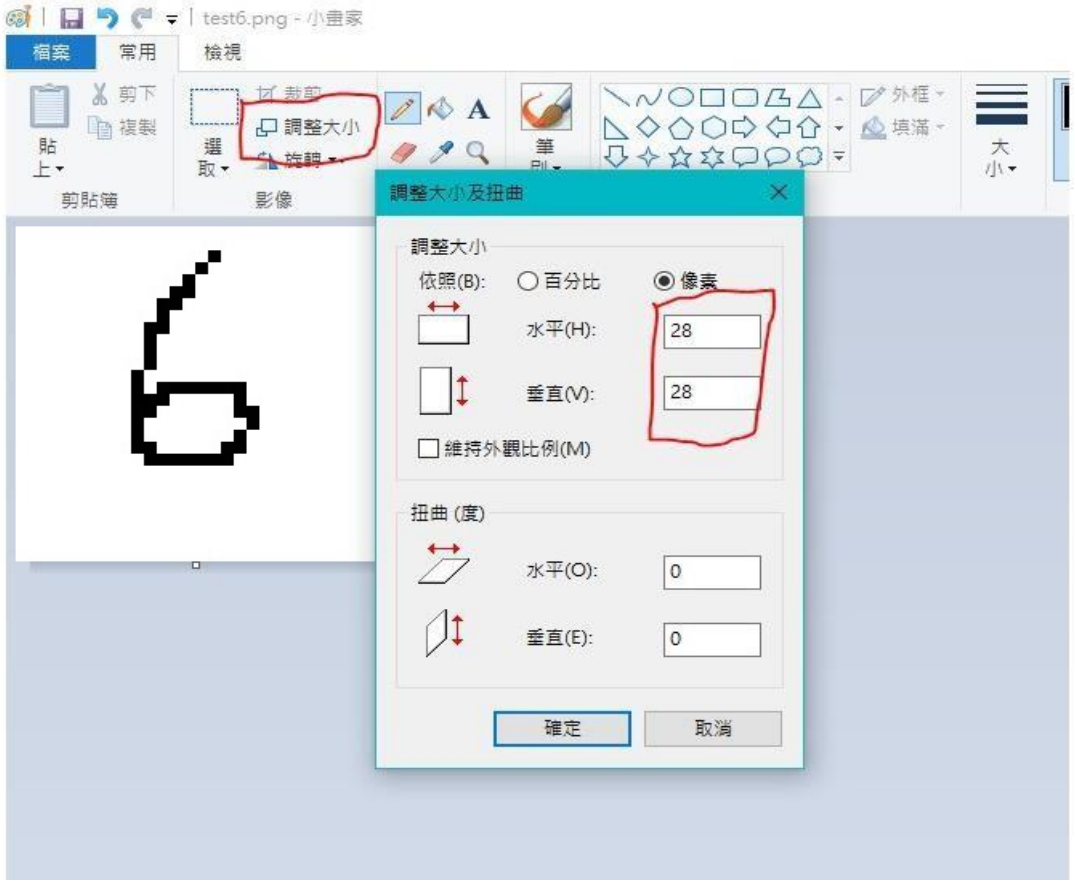
```
img = Image.open('./test/3.png')

plt.figure(figsize=(1,1))
plt.imshow(img)
plt.show()
```



預測圖片請各位開啟小畫家手寫

預測


test6.png - 小畫家
檔案
常用
檢視
剪下
複製
貼上
剪貼簿
選取
裁剪
調整大小
旋轉
影像
筆刷
外框
填滿
大小
調整大小及扭曲
調整大小
依照(B):
百分比
像素
水平(H):
28
垂直(V):
28
維持外觀比例(M)
扭曲 (度)
水平(O):
0
垂直(E):
0
確定
取消

預測

預測圖片前處理

```
img = np.array(img)
```

```
img = np.dot(img[...,:3], [0.299, 0.587, 0.114])
```

```
img = 255-img
img = img / 255.
```

```
img = np.reshape(img,(1,28,28,1))
```

函式

```
img = np.dot(img[...,:3], [0.299, 0.587, 0.114])
```

基於matplotlib和公式將 RGB 圖轉換為灰度圖，只對文字內容感興趣

$$Y' = 0.299\ R + 0.587\ G + 0.114\ B$$

函式

```
img = 255-img
img = img / 255.
```

這裡讀入的圖片陣列是 float32 型的，範圍是 0-1，而 PIL.Image 資料是 uinit8 型的，範圍是0-255，所以要進 行轉換

預測

預測圖片

```
model.predict(img)
```

```
array([[7.4919909e-03, 2.3622868e-04, 1.3560286e-02, 9.0231025e-01,
        1.0235576e-03, 3.6447242e-04, 5.5334434e-02, 2.8058883e-05,
        1.8490007e-02, 1.1606632e-03]], dtype=float32)
```

預測

預測多張圖片

```
file = ['./test/0.png',
        './test/1.png',
        './test/2.png',
        './test/3.png',
       './test/4.png',
       './test/5.png',
       './test/6.png',
       './test/7.png',
       './test/8.png',
       './test/9.png']

for f in file:
    img = Image.open(f)
    plt.figure(figsize=(1,1))
    plt.imshow(img)
    plt.show()
    img = np.array(img)
    img = np.dot(img[...,:3], [0.299, 0.587, 0.114])
    img = 255-img
    img = img / 255.
    img = np.reshape(img,(1,28,28,1))
    model.predict(img)
    print(np.argmax(model.predict(img)))
    print("-------------------------------------------------")
```

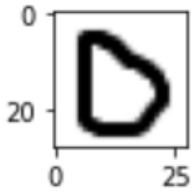
- 將手寫圖片放入”test” 資料夾

函式

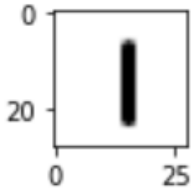
- **plt.figure(figsize=(float,float))** 寬度,高度（以英寸為單位）

https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.figure.html

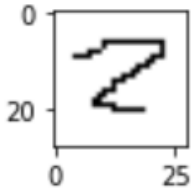
預測



0



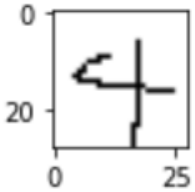
1



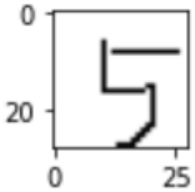
2

預測

3



4

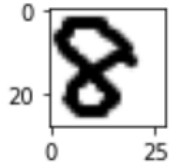


5

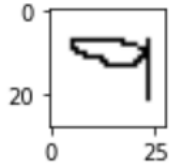
預測

6

7



8



9

下載win32 燒入程式

http://downloads.raspberrypi.org/raspbian/images/

下載img檔

raspbian-2017-06-23/	2017-06-23 07:14	-
raspbian-2017-07-05/	2017-07-05 17:43	-
raspbian-2017-08-17/	2017-08-17 09:17	-
raspbian-2017-09-08/	2017-09-08 12:13	-
raspbian-2017-12-01/	2017-12-01 10:22	-
raspbian-2018-03-14/	2018-03-16 18:07	-
raspbian-2018-04-19/	2018-04-19 15:40	-
raspbian-2018-06-29/	2018-06-29 03:28	-
raspbian-2018-10-11/	2018-10-11 11:46	-
raspbian-2018-11-15/	2018-11-15 21:06	-
raspbian-2019-04-09/	2019-04-09 23:46	-
raspbian-2019-06-24/	2019-06-24 07:20	-
raspbian-2019-07-12/	2019-07-12 14:55	-
raspbian-2019-09-30/	2019-09-30 15:52	-
raspbian-2020-02-07/	2020-02-07 08:23	-
raspbian-2020-02-14/	2020-02-14 15:33	-

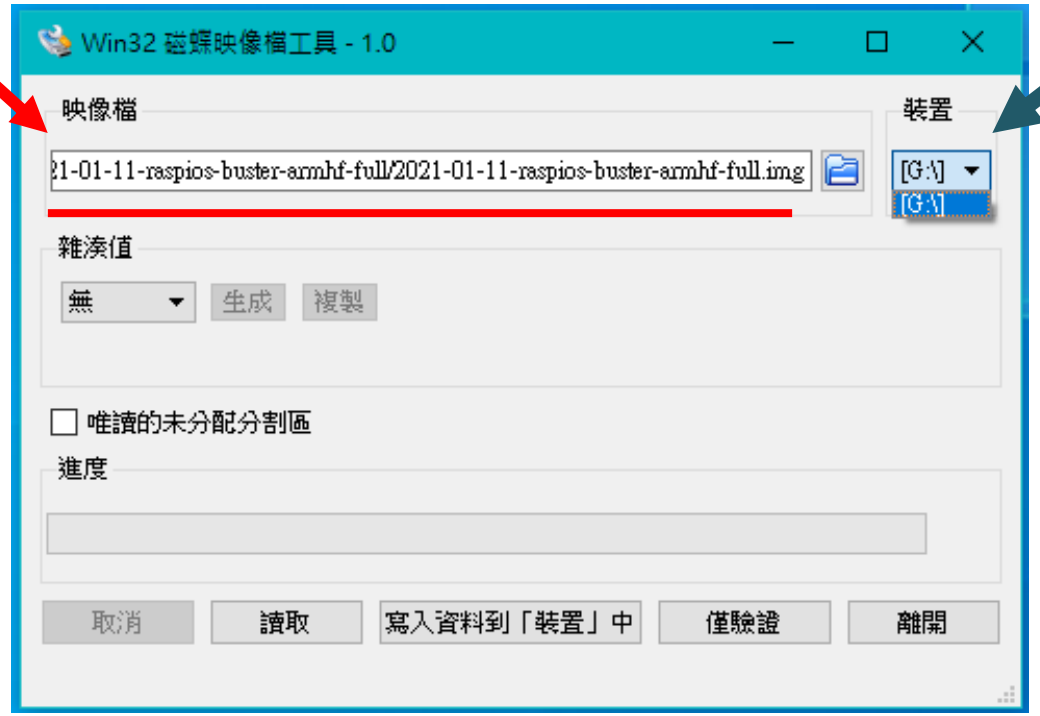
點開後 進到這畫面 點第三個 下載

Index of /raspbian/images/raspbian-2020-02-14

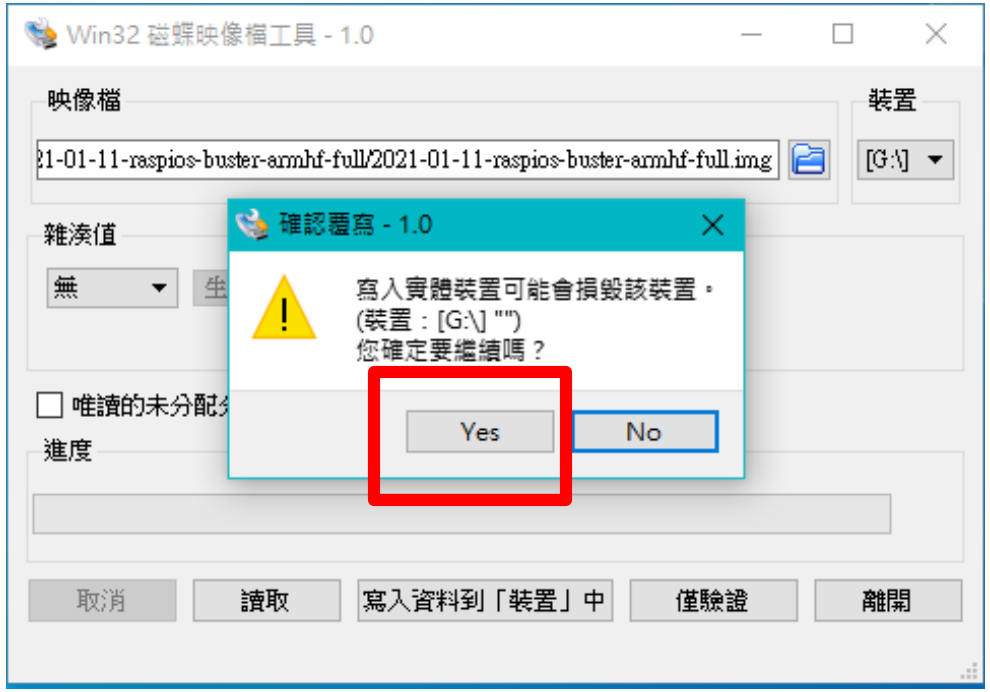
Name	Last modified	Size	Description
Parent Directory		-	
2020-02-13-raspbian-buster.info	2020-02-13 16:21	170K	
2020-02-13-raspbian-buster.zip	2020-02-13 16:21	1.1G	
2020-02-13-raspbian-buster.zip.sha1	2020-02-14 13:47	73	
2020-02-13-raspbian-buster.zip.sha256	2020-02-14 13:48	97	
2020-02-13-raspbian-buster.zip.sig	2020-02-14 12:55	488	
2020-02-13-raspbian-buster.zip.torrent	2020-02-14 13:48	22K	

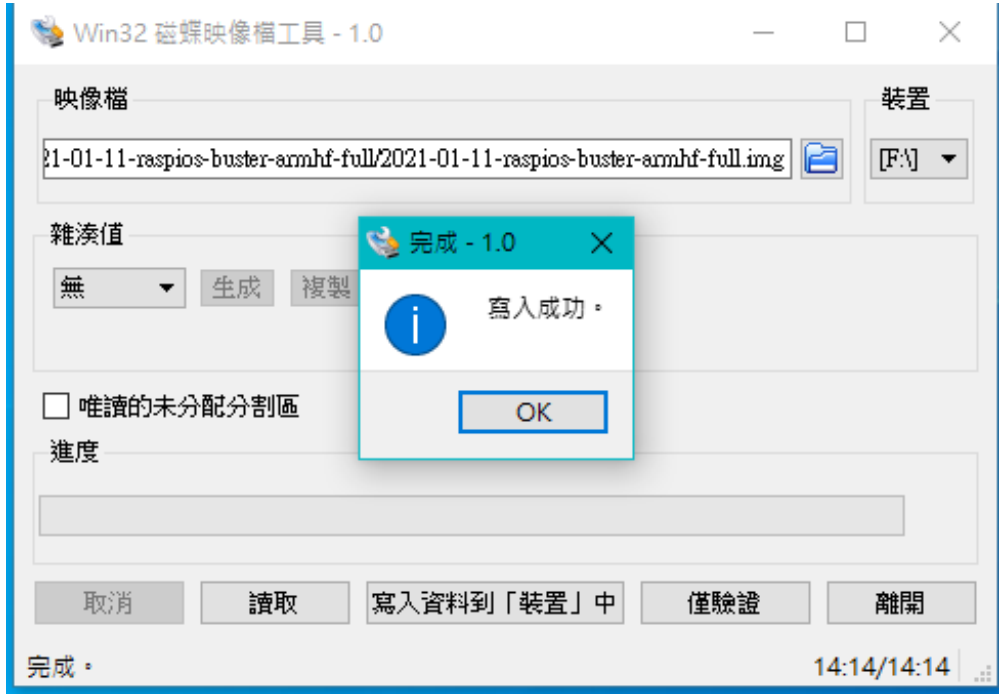
2021-01-11-raspios-buster-armhf-full

名稱	修改日期
2021-01-11-raspios-buster-armhf-full.i...	2022/10/20 下



SD卡


Win32 磁碟映像檔工具 - 1.0
映像檔
21-01-11-raspios-buster-armhf-full/2021-01-11-raspios-buster-armhf-full.img
裝置
[G:\]
雜湊值
無
唯讀的未分配
進度
確認覆寫 - 1.0
寫入實體裝置可能會損毀該裝置。
(裝置：[G:\] "")
您確定要繼續嗎？
Yes
No
取消
讀取
寫入資料到「裝置」中
僅驗證
離開



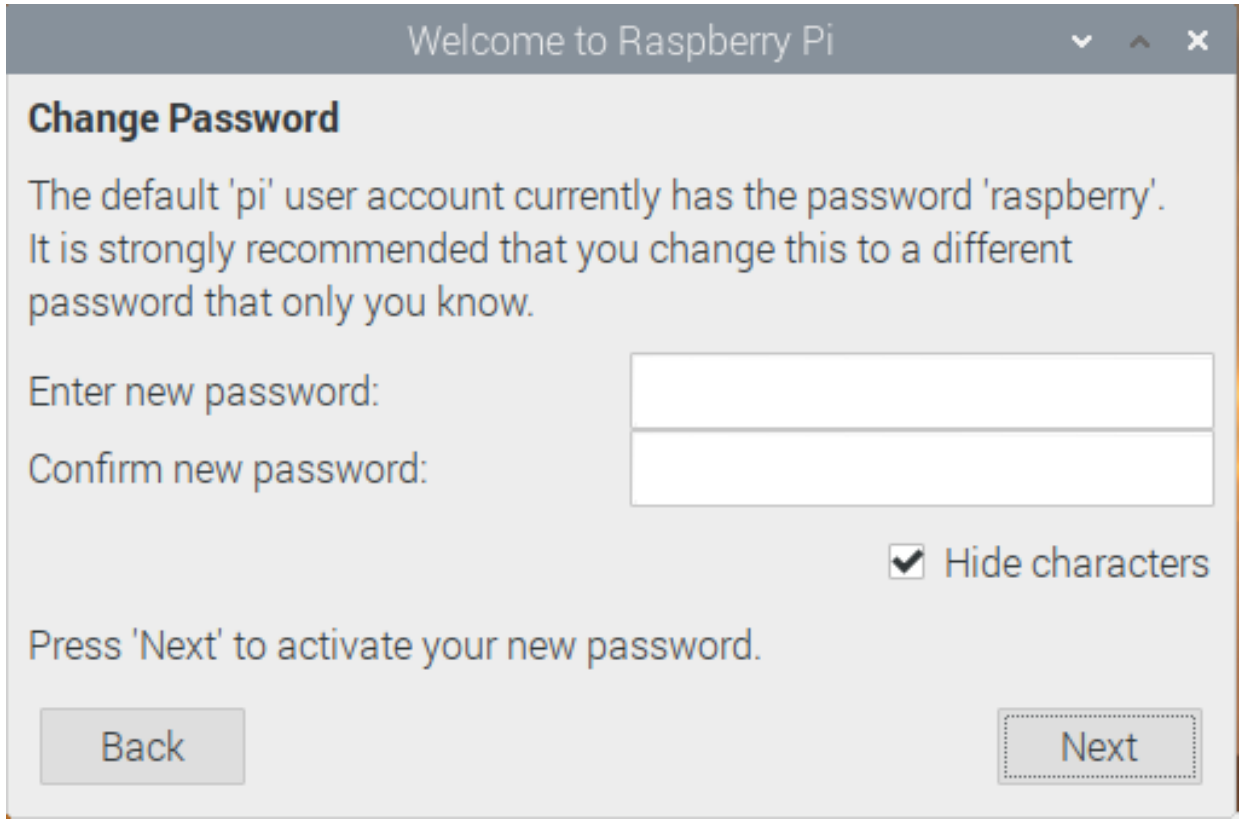
有出現 格式化視窗
直接關閉即可

把SD卡放到 樹梅派

設定


Welcome to Raspberry Pi
Set Country
Enter the details of your location. This is used to set the language, time zone, keyboard and other international settings.
Country:
Taiwan
Language:
Chinese
Timezone:
Taipei
Use English language
Use US keyboard
Press 'Next' when you have made your selection.
Back
Next

設定



預設密碼為raspberry，可以不用填

WiFi

在樹梅派上設定安裝包

以下指令 可到官網 https://coral.ai/docs/accelerator/get-started/#1-install-the-edge-tpu-runtime 複製指令

安裝套件階段 不要插上TPU!!!!!!!!!!!!
安裝套件階段 不要插上TPU!!!!!!!!!!!!
安裝套件階段 不要插上TPU!!!!!!!!!!!!

在樹梅派上設定安裝指令

- echo "deb https://packages.cloud.google.com/apt coral-edgetpu-stable main" | sudo tee /etc/apt/sources.list.d/coral-edgetpu.list
- curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
- sudo apt-get update

- sudo apt-get install libedgetpu1-std
- sudo apt-get install python3-pycoral

以下測試指令 請插上TPU

```
mkdir coral && cd coral
```

```
git clone https://github.com/google-coral/pycoral.git
```

```
cd pycoral
```

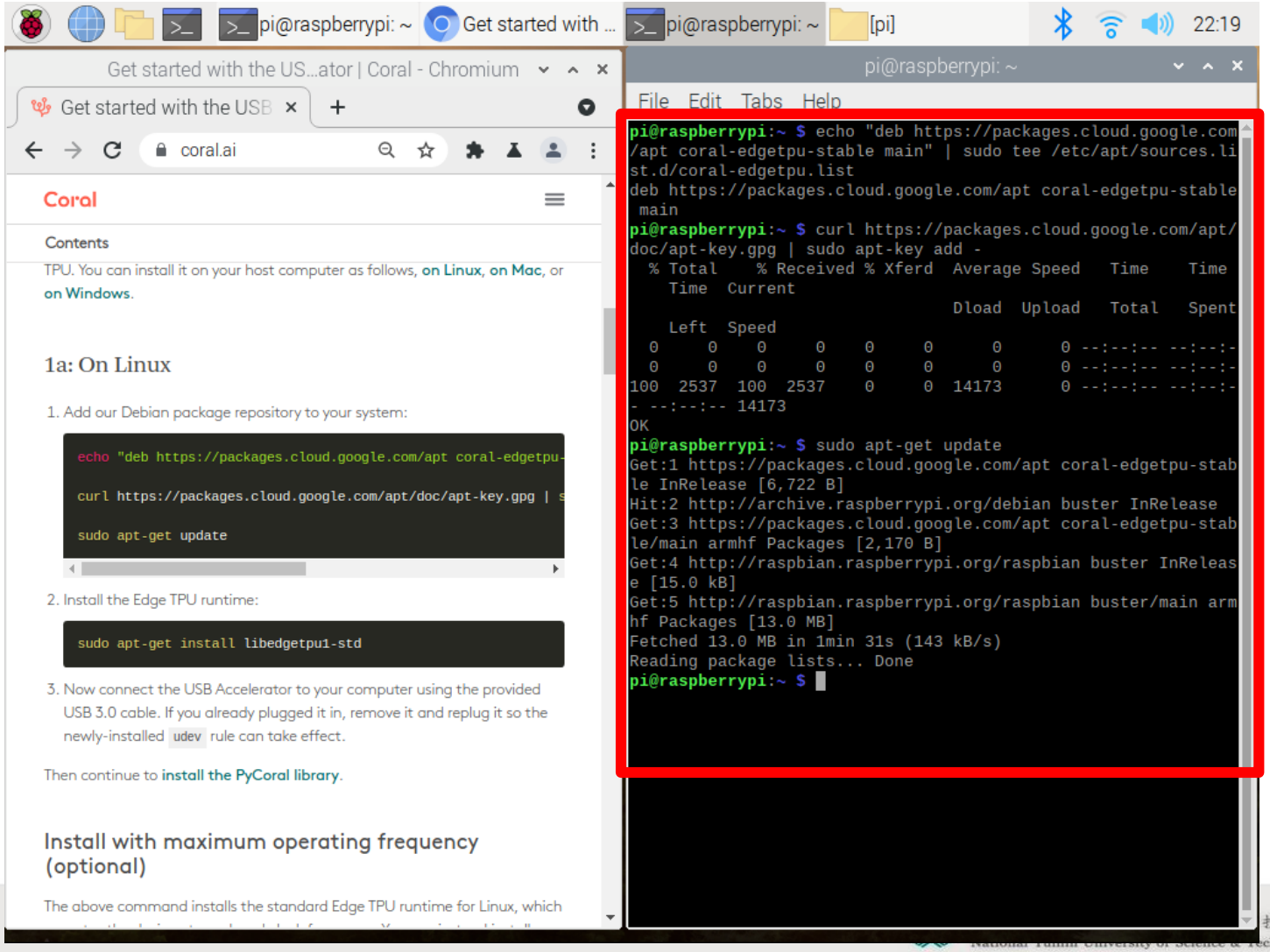
```
bash examples/install_requirements.sh classify_image.py
```

```
python3 examples/classify_image.py \
 --model test_data/mobilenet_v2_1.0_224_inat_bird_quant_edgetpu.tflite \
 --labels test_data/inat_bird_labels.txt \ --input test_data/parrot.jpg
```

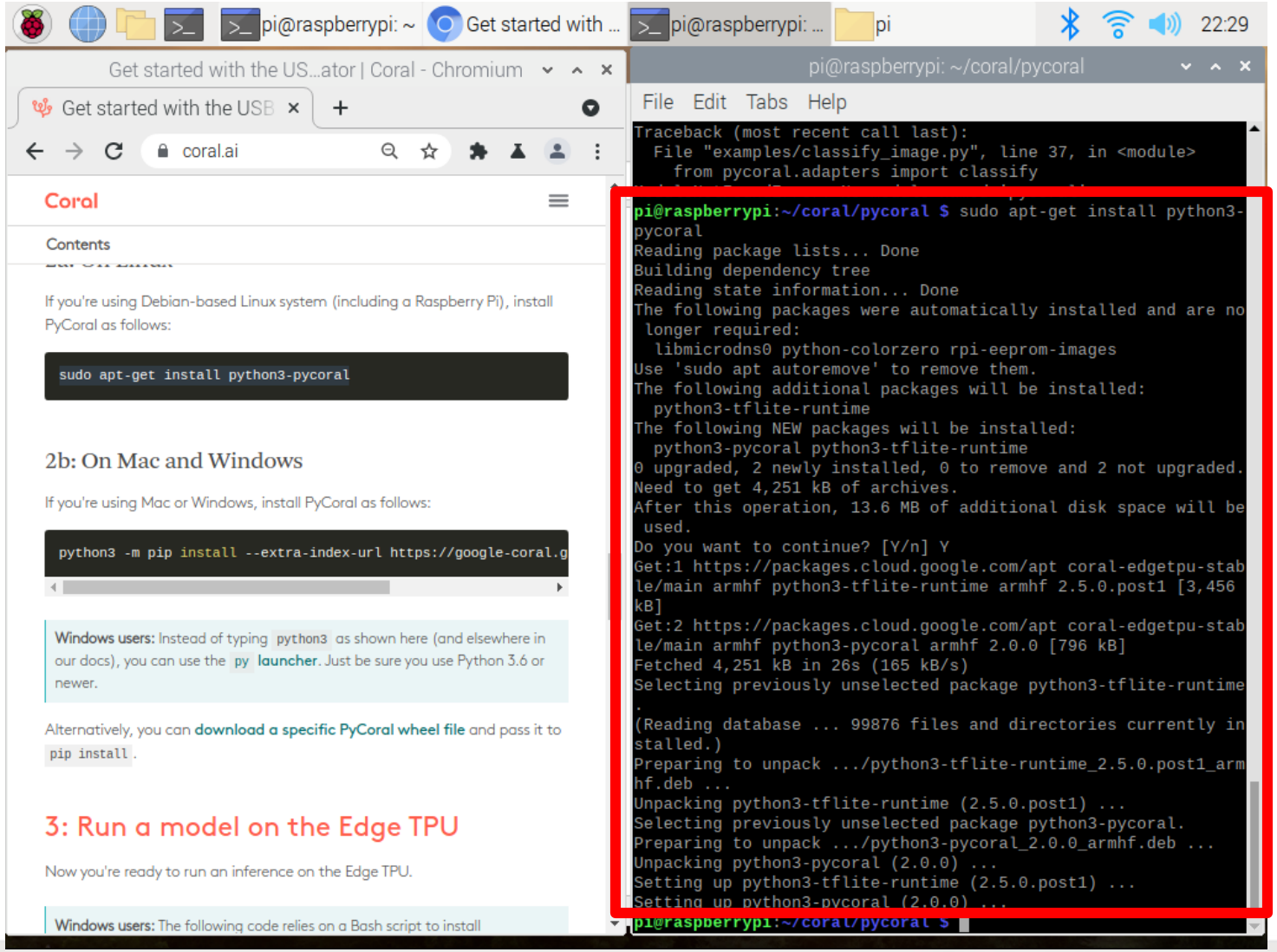
這測試指令 簡而言之 載測試的模型 程式 圖片

最後一行去執行

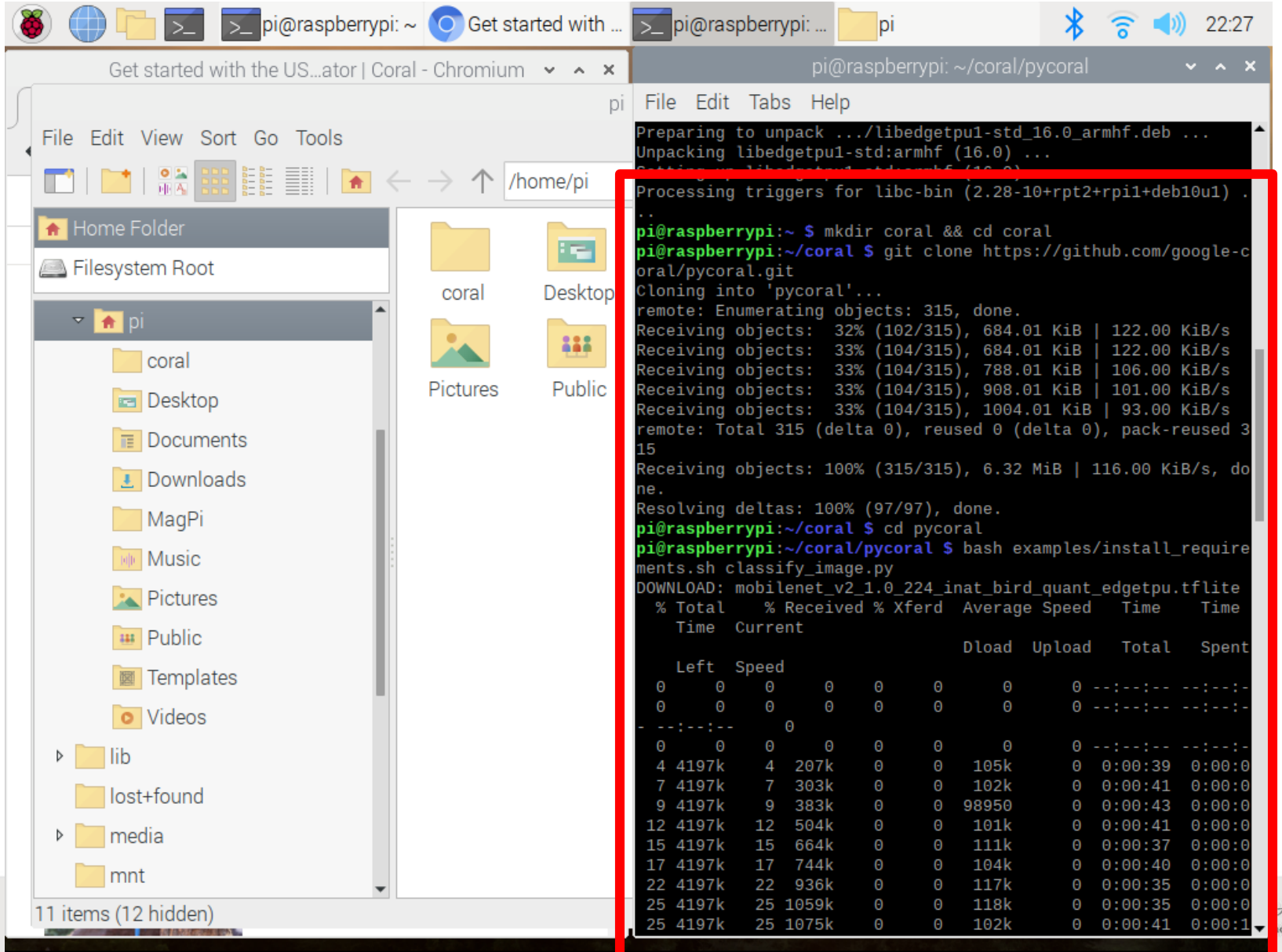
套件成功畫面


pi@raspberrypi: ~
Get started with ...
pi@raspberrypi: ~
[pi]
22:19
Get started with the US...ator | Coral - Chromium
Get started with the USB
coral.ai
Coral
Contents
TPU. You can install it on your host computer as follows, on Linux, on Mac, or on Windows.
1a: On Linux
1. Add our Debian package repository to your system:
echo "deb https://packages.cloud.google.com/apt coral-edgetpu-
curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | s
sudo apt-get update
2. Install the Edge TPU runtime:
sudo apt-get install libedgetpu1-std
3. Now connect the USB Accelerator to your computer using the provided USB 3.0 cable. If you already plugged it in, remove it and replug it so the newly-installed udev rule can take effect.
Then continue to install the PyCoral library.
Install with maximum operating frequency (optional)
The above command installs the standard Edge TPU runtime for Linux, which
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ echo "deb https://packages.cloud.google.com/apt coral-edgetpu-stable main" | sudo tee /etc/apt/sources.list.d/coral-edgetpu.list
deb https://packages.cloud.google.com/apt coral-edgetpu-stable main
pi@raspberrypi:~ $ curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
0 0 0 0 0 0 0 0 --:--:-- --:--:--
0 0 0 0 0 0 0 0 --:--:-- --:--:--
100 2537 100 2537 0 0 14173 0 --:--:-- --:--:-- --:--:-- 14173
OK
pi@raspberrypi:~ $ sudo apt-get update
Get:1 https://packages.cloud.google.com/apt coral-edgetpu-stable InRelease [6,722 B]
Hit:2 http://archive.raspberrypi.org/debian buster InRelease
Get:3 https://packages.cloud.google.com/apt coral-edgetpu-stable/main armhf Packages [2,170 B]
Get:4 http://raspbian.raspberrypi.org/raspbian buster InRelease [15.0 kB]
Get:5 http://raspbian.raspberrypi.org/raspbian buster/main armhf Packages [13.0 MB]
Fetched 13.0 MB in 1min 31s (143 kB/s)
Reading package lists... Done
pi@raspberrypi:~ $

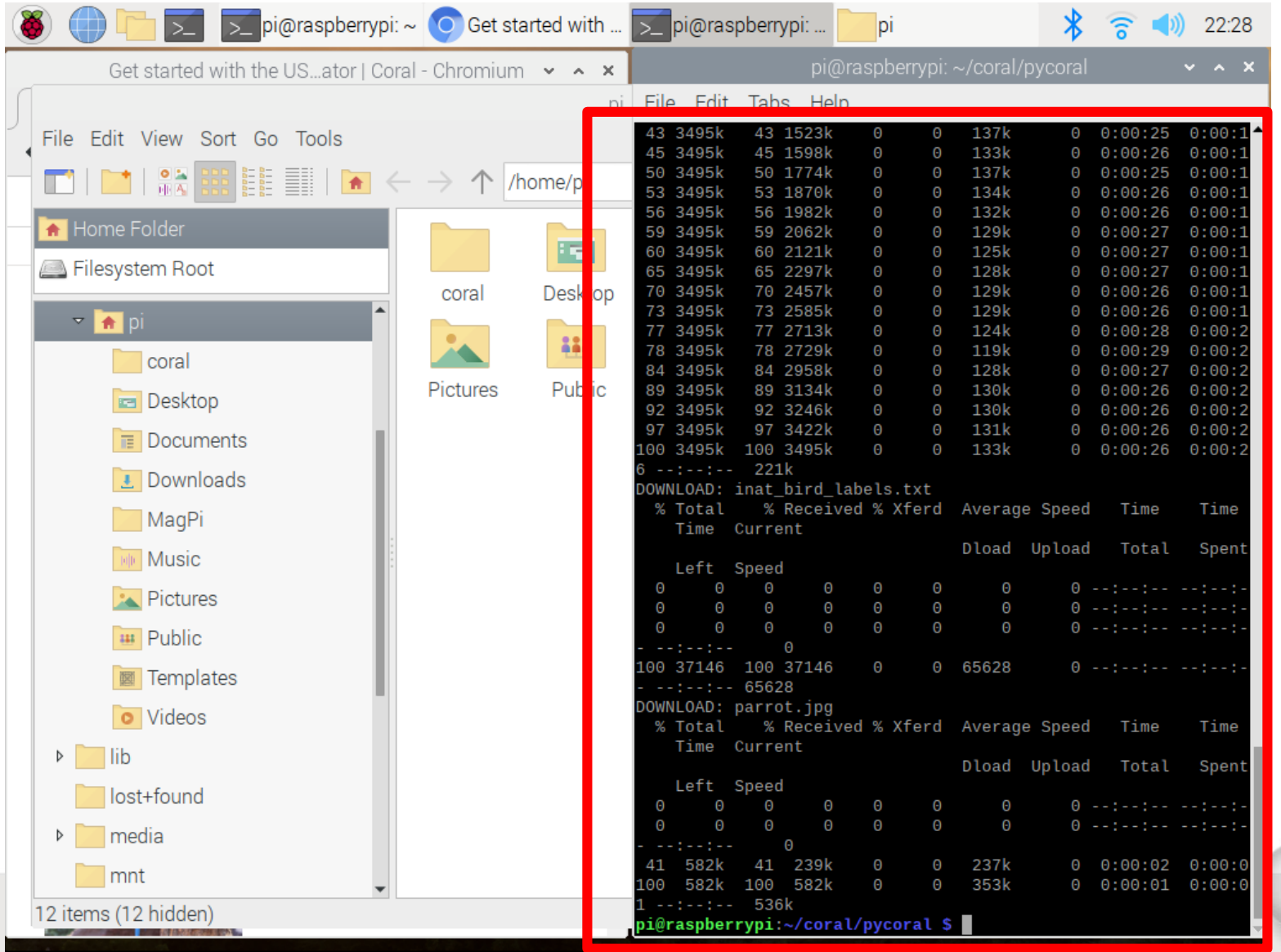
套件成功畫面


pi@raspberrypi: ~
Get started with ...
pi@raspberrypi: ...
pi
22:29
Get started with the US...ator | Coral - Chromium
Get started with the USB
coral.ai
Coral
Contents
If you're using Debian-based Linux system (including a Raspberry Pi), install PyCoral as follows:
sudo apt-get install python3-pycoral
2b: On Mac and Windows
If you're using Mac or Windows, install PyCoral as follows:
python3 -m pip install --extra-index-url https://google-coral.g
Windows users: Instead of typing python3 as shown here (and elsewhere in our docs), you can use the py launcher. Just be sure you use Python 3.6 or newer.
Alternatively, you can download a specific PyCoral wheel file and pass it to pip install.
3: Run a model on the Edge TPU
Now you're ready to run an inference on the Edge TPU.
Windows users: The following code relies on a Bash script to install
pi@raspberrypi: ~/coral/pycoral
File Edit Tabs Help
Traceback (most recent call last):
File "examples/classify_image.py", line 37, in <module>
from pycoral.adapters import classify
pi@raspberrypi:~/coral/pycoral $ sudo apt-get install python3-pycoral
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
libmicrodns0 python-colorzero rpi-eeprom-images
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
python3-tflite-runtime
The following NEW packages will be installed:
python3-pycoral python3-tflite-runtime
0 upgraded, 2 newly installed, 0 to remove and 2 not upgraded.
Need to get 4,251 kB of archives.
After this operation, 13.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 https://packages.cloud.google.com/apt coral-edgetpu-stable/main armhf python3-tflite-runtime armhf 2.5.0.post1 [3,456 kB]
Get:2 https://packages.cloud.google.com/apt coral-edgetpu-stable/main armhf python3-pycoral armhf 2.0.0 [796 kB]
Fetched 4,251 kB in 26s (165 kB/s)
Selecting previously unselected package python3-tflite-runtime.
(Reading database ... 99876 files and directories currently installed.)
Preparing to unpack .../python3-tflite-runtime_2.5.0.post1_armhf.deb ...
Unpacking python3-tflite-runtime (2.5.0.post1) ...
Selecting previously unselected package python3-pycoral.
Preparing to unpack .../python3-pycoral_2.0.0_armhf.deb ...
Unpacking python3-pycoral (2.0.0) ...
Setting up python3-tflite-runtime (2.5.0.post1) ...
Setting up python3-pycoral (2.0.0) ...
pi@raspberrypi:~/coral/pycoral $

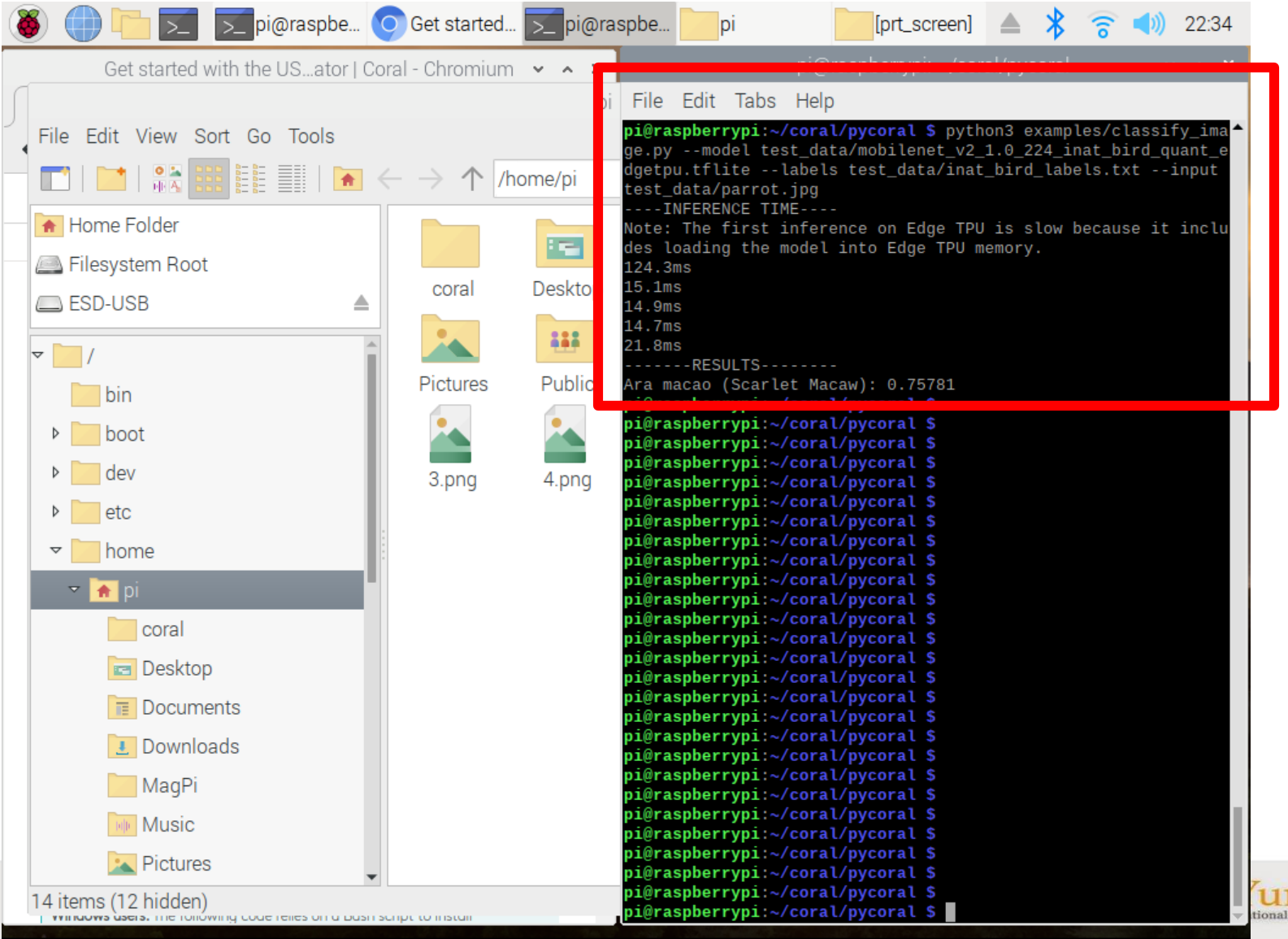
測試成功畫面


pi@raspberrypi: ~
Get started with ...
pi@raspberrypi: ...
pi
22:27
Get started with the US...ator | Coral - Chromium
File Edit View Sort Go Tools
/home/pi
Home Folder
Filesystem Root
pi
coral
Desktop
Documents
Downloads
MagPi
Music
Pictures
Public
Templates
Videos
lib
lost+found
media
mnt
11 items (12 hidden)
pi@raspberrypi: ~/coral/pycoral
File Edit Tabs Help
Preparing to unpack .../libedgetpu1-std_16.0_armhf.deb ...
Unpacking libedgetpu1-std:armhf (16.0) ...
Processing triggers for libc-bin (2.28-10+rpt2+rpi1+deb10u1) ...
pi@raspberrypi:~ $ mkdir coral && cd coral
pi@raspberrypi:~/coral $ git clone https://github.com/google-coral/pycoral.git
Cloning into 'pycoral'...
remote: Enumerating objects: 315, done.
Receiving objects: 32% (102/315), 684.01 KiB | 122.00 KiB/s
Receiving objects: 33% (104/315), 684.01 KiB | 122.00 KiB/s
Receiving objects: 33% (104/315), 788.01 KiB | 106.00 KiB/s
Receiving objects: 33% (104/315), 908.01 KiB | 101.00 KiB/s
Receiving objects: 33% (104/315), 1004.01 KiB | 93.00 KiB/s
remote: Total 315 (delta 0), reused 0 (delta 0), pack-reused 315
Receiving objects: 100% (315/315), 6.32 MiB | 116.00 KiB/s, done.
Resolving deltas: 100% (97/97), done.
pi@raspberrypi:~/coral $ cd pycoral
pi@raspberrypi:~/coral/pycoral $ bash examples/install_requirements.sh classify_image.py
DOWNLOAD: mobilenet_v2_1.0_224_inat_bird_quant_edgetpu.tflite
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
0 0 0 0 0 0 0 0 --:--:-- --:--:--
0 0 0 0 0 0 0 0 --:--:-- --:--:-- --:--:-- 0
0 0 0 0 0 0 0 0 --:--:-- --:--:--
4 4197k 4 207k 0 0 105k 0 0:00:39 0:00:0
7 4197k 7 303k 0 0 102k 0 0:00:41 0:00:0
9 4197k 9 383k 0 0 98950 0 0:00:43 0:00:0
12 4197k 12 504k 0 0 101k 0 0:00:41 0:00:0
15 4197k 15 664k 0 0 111k 0 0:00:37 0:00:0
17 4197k 17 744k 0 0 104k 0 0:00:40 0:00:0
22 4197k 22 936k 0 0 117k 0 0:00:35 0:00:0
25 4197k 25 1059k 0 0 118k 0 0:00:35 0:00:0
25 4197k 25 1075k 0 0 102k 0 0:00:41 0:00:1

測試成功畫面


pi@raspberrypi: ~
Get started with ...
pi@raspberrypi: ...
pi
22:28
Get started with the US...ator | Coral - Chromium
File Edit View Sort Go Tools
/home/p
Home Folder
Filesystem Root
pi
coral
Desktop
Documents
Downloads
MagPi
Music
Pictures
Public
Templates
Videos
lib
lost+found
media
mnt
12 items (12 hidden)
pi@raspberrypi: ~/coral/pycoral
File Edit Tabs Help
43 3495k 43 1523k 0 0 137k 0 0:00:25 0:00:1
45 3495k 45 1598k 0 0 133k 0 0:00:26 0:00:1
50 3495k 50 1774k 0 0 137k 0 0:00:25 0:00:1
53 3495k 53 1870k 0 0 134k 0 0:00:26 0:00:1
56 3495k 56 1982k 0 0 132k 0 0:00:26 0:00:1
59 3495k 59 2062k 0 0 129k 0 0:00:27 0:00:1
60 3495k 60 2121k 0 0 125k 0 0:00:27 0:00:1
65 3495k 65 2297k 0 0 128k 0 0:00:27 0:00:1
70 3495k 70 2457k 0 0 129k 0 0:00:26 0:00:1
73 3495k 73 2585k 0 0 129k 0 0:00:26 0:00:1
77 3495k 77 2713k 0 0 124k 0 0:00:28 0:00:2
78 3495k 78 2729k 0 0 119k 0 0:00:29 0:00:2
84 3495k 84 2958k 0 0 128k 0 0:00:27 0:00:2
89 3495k 89 3134k 0 0 130k 0 0:00:26 0:00:2
92 3495k 92 3246k 0 0 130k 0 0:00:26 0:00:2
97 3495k 97 3422k 0 0 131k 0 0:00:26 0:00:2
100 3495k 100 3495k 0 0 133k 0 0:00:26 0:00:2
6 --:--:-- 221k
DOWNLOAD: inat_bird_labels.txt
% Total % Received % Xferd Average Speed Time Time
Time Current
Dload Upload Total Spent
Left Speed
0 0 0 0 0 0 0 0 --:--:-- --:--:-
0 0 0 0 0 0 0 0 --:--:-- --:--:-
0 0 0 0 0 0 0 0 --:--:-- --:--:-
- --:--:-- 0
100 37146 100 37146 0 0 65628 0 --:--:-- --:--:-
- --:--:-- 65628
DOWNLOAD: parrot.jpg
% Total % Received % Xferd Average Speed Time Time
Time Current
Dload Upload Total Spent
Left Speed
0 0 0 0 0 0 0 0 --:--:-- --:--:-
0 0 0 0 0 0 0 0 --:--:-- --:--:-
- --:--:-- 0
41 582k 41 239k 0 0 237k 0 0:00:02 0:00:0
100 582k 100 582k 0 0 353k 0 0:00:01 0:00:0
1 --:--:-- 536k
pi@raspberrypi:~/coral/pycoral $

測試成功畫面



會跑出放進圖片的預測結果

登出

先登出等完全斷電再拔除電源

Reference

- https://www.raspberrypi.org/downloads/
- https://coral.ai/

END