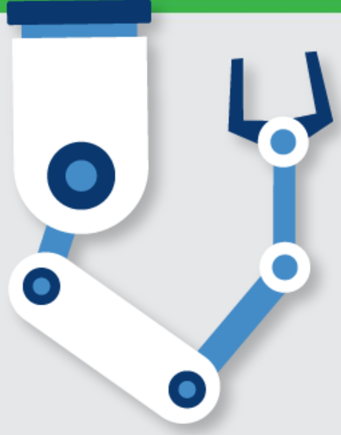
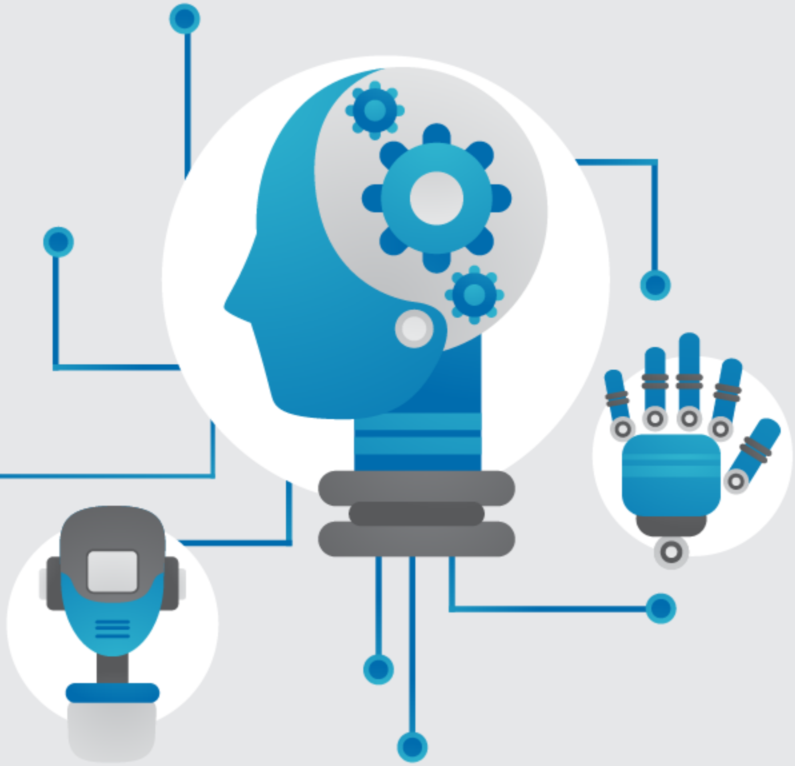
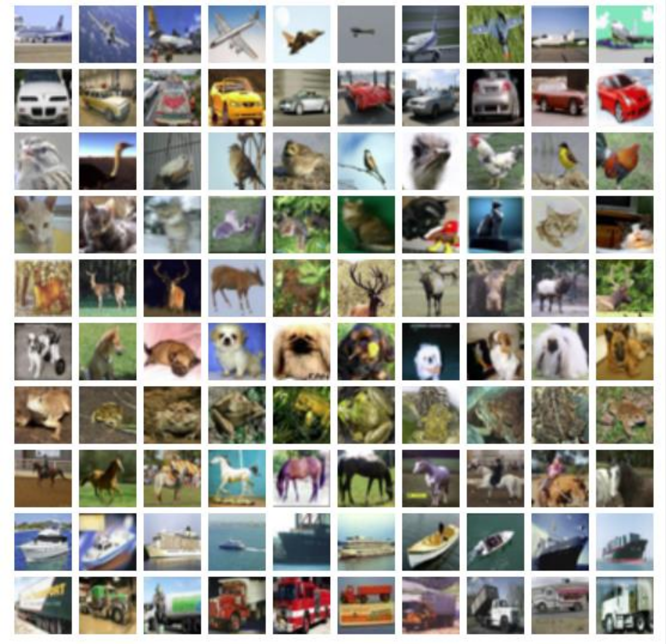


Cifar-10資料集介紹



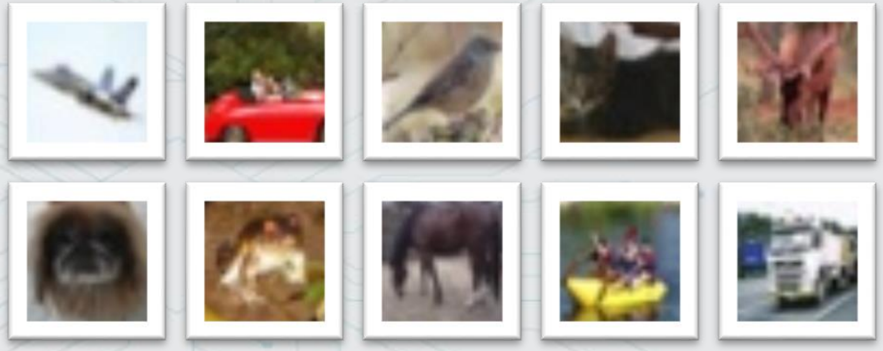
CIFAR-10資料集

- CIFAR-10圖片資料集（Canadian Institute For Advanced Research）是取自於80 million tiny images資料集中的10種類別
- 是由Alex Krizhevsky, Vinod Nair和Geoffrey Hinton所蒐集



CIFAR-10資料集

- 總共60,000張32x32的RGB彩色影像
- 分為10個類別，每類6000張
- 訓練資料佔50,000張，測試資料佔10,000張
- 類別有airplane、automobile、bird、cat、deer、dog、frog、horse、ship、truck



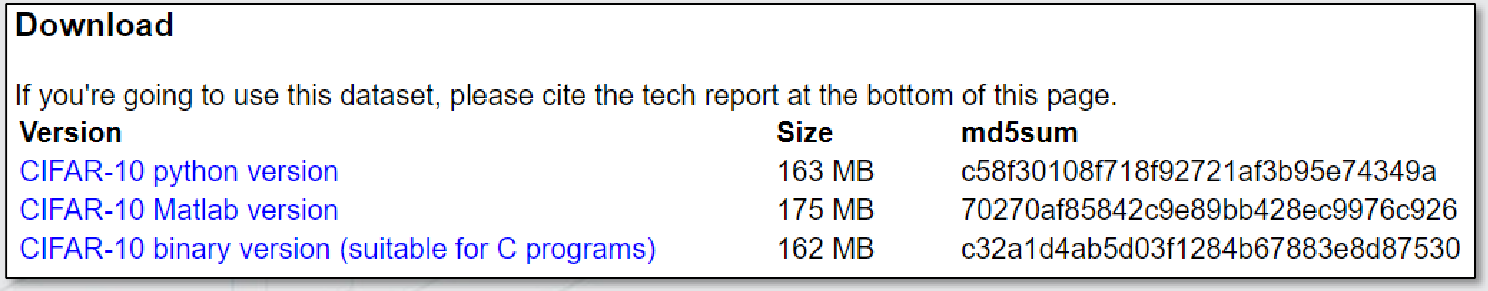
CIFAR-100資料集

- 同Cifar-10，取自於80 million tiny images資料集中的100種類別。
- 每類有600張影像，其中500張訓練，100張測試，且整理成20個群組，將每5種類別歸為一群。

Superclass	Classes
aquatic mammals	beaver,dolphin,otter,seal,whale
fish	aquarium fish,flatfish,ray,shark,trout
flowers	orchids,poppies,rose,sunflowers,tulips
food containers	bottles,bowls,cans,cups,plates
fruit and vegetables	apples,mushrooms,oranges,pears,sweet peppers
household electrical devices	clock,computer keyboard,lamp,telephone,television
household furniture	bed,chair,couch,table,wardrobe
insects	bee,beetle,butterfly,caterpillar,cockroach
large carnivores	bear,leopard,lion,tiger,wolf
large man-made outdoor things	bridge,castle,house,road,skyscraper
large natural outdoor scenes	cloud,forest,mountain,plain,sea
large omnivores and herbivores	camel,cattle,chimpanzee,elephant,kangaroo
medium-sized mammals	fox,porcupine,possum,raccoon,skunk
non-insect invertebrates	crab,lobster,snail,spider,worm
people	baby,boy,girl,man,woman
reptiles	crocodile,dinosaur,lizard,snake,turtle
small mammals	hamster,mouse,rabbit,shrew,squirrel
trees	maple,oak,palm,pine,willow
vehicles 1	bicycle,bus,motorcycle,pickup truck,train
vehicles 2	lawn-mower,rocket,streetcar,tank,tractor

CIFAR-10資料集下載

- 下載網頁
  https://www.cs.toronto.edu/~kriz/cifar.html



Download

If you're going to use this dataset, please cite the tech report at the bottom of this page.

Version	Size	md5sum
CIFAR-10 python version	163 MB	c58f30108f718f92721af3b95e74349a
CIFAR-10 Matlab version	175 MB	70270af85842c9e89bb428ec9976c926
CIFAR-10 binary version (suitable for C programs)	162 MB	c32a1d4ab5d03f1284b67883e8d87530

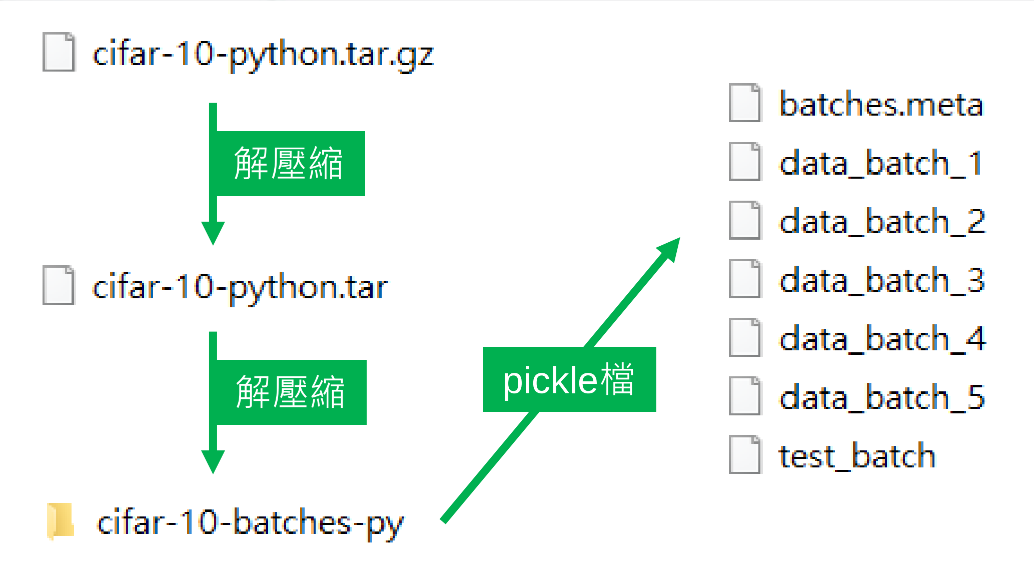
- Keras套件內建資料集

```
from keras.datasets import cifar10
(x_train, y_train), (x_test, y_test) = cifar10.load_data()
```

CIFAR-10資料集內容

- 作者網頁有提供python的pickle、Matlab的mat和純二進位檔，檔案的規劃如下：

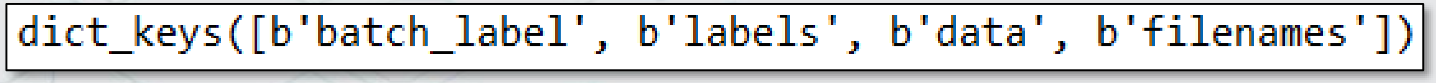
  batchs.meta內容是該資料集的摘要
  data_batch_1~data_batch_5是訓練資料
  test_batch是測試資料



CIFAR-10資料集內容

- pickle是用來保存python物件的套件，讀取CIFAR-10程式碼如下：

```
def unpickle(file):
    import pickle
    with open(file, 'rb') as fo:
        dict = pickle.load(fo, encoding='bytes')
    return dict
data_batch_1=unpickle('cifar-10-batches-py/data_batch_1')
print(data_batch_1.keys())
```



```
dict_keys([b'batch_label', b'labels', b'data', b'filenames'])
```

- labels為0~9的數字，代表類別
- data為長度3072的整數陣列，對應到影像資料32*32*3

CIFAR-10資料集內容

- print(data_batch_1[b'data'])

10000筆長度為3072的陣列

```
[[  59    43    50  …  140  84  72  ]
 [154  126  105  … 139  142  144  ]
 [255  253  253  …      83  83  84  ]
…
[71   60   74   …   68   69   68]
[250  254  211  …  215  255  254]
[62   61   60   …  130  130  131]]
```

CIFAR-10資料集內容

- 3072個整數：

依序為**紅**通道、**綠**通道和**藍**通道各1024個，皆為row major

- 以最後一筆資料為例

```
import cv2
last_img=data_batch_1[b'data' ][-1]  # 取得最後一筆影像
last_img=last_img.reshape((3,32,32)) # row major
last_img=last_img.transpose(1, 2, 0) # 轉置成column, row, depth
last_img=cv2.cvtColor(last_img, cv2.COLOR_RGB2BGR) #opencv為BGR
cv2.imwrite('a.jpg', last_img)  # 存成檔案
```

CIFAR-10資料集內容

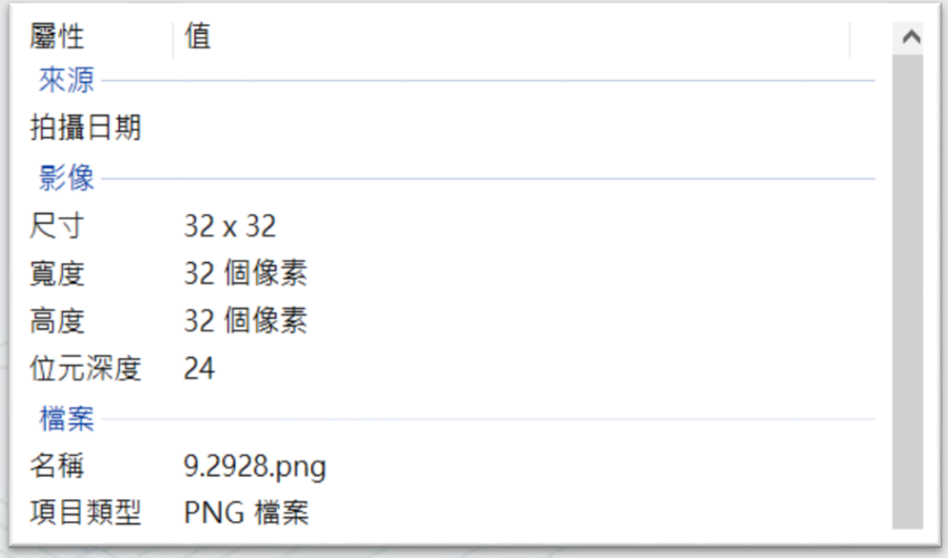
- 直接使用Keras內建資料集，內容為32*32*3的陣列

```
from keras.datasets import cifar10
(x_train, y_train), (x_test, y_test) = cifar10.load_data()
print(x_train.shape, y_train.shape)
```

```
（ 50000, 32, 32, 3 ）（ 50000,1 ）
```

CIFAR-10圖檔處理

- 圖檔格式：60,000張32X32X3 png圖檔



屬性	值
來源	
拍攝日期	
影像	
尺寸	32 x 32
寬度	32 個像素
高度	32 個像素
位元深度	24
檔案	
名稱	9.2928.png
項目類型	PNG 檔案

CIFAR-10圖檔處理

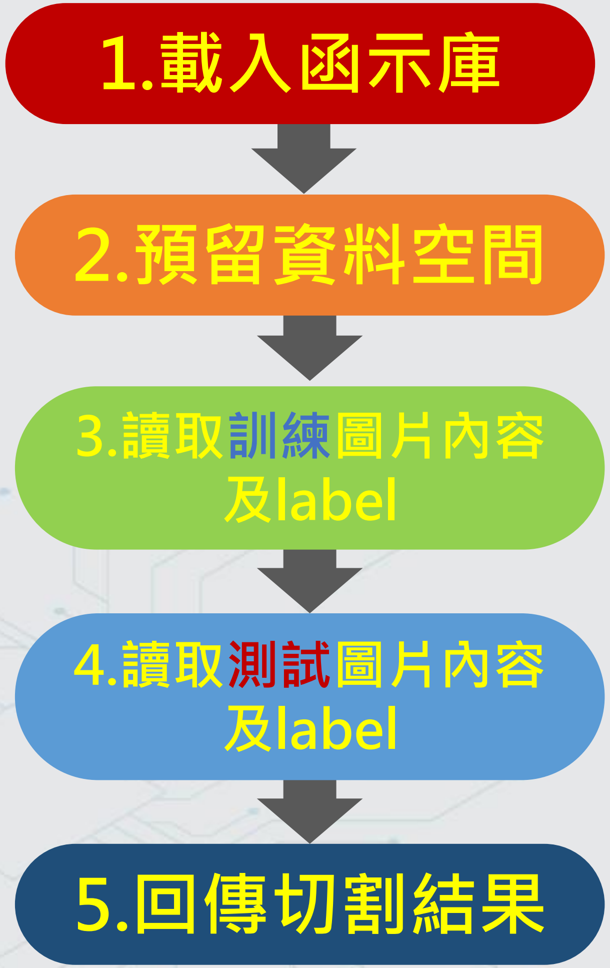
- 資料切割：訓練資料50,000張，測試資料10,000張
- 類別：0 ~ 9
- 圖檔檔名編碼：
  0.0.jpg, 0.1.jpg, 0.2.jpg, ..., 0.4999.jpg,

  1.0.jpg, 1.1.jpg, 1.2.jpg, ..., 1.4999.jpg,

  ……

  9.0.jpg, 9.1.jpg, 9.2.jpg, ..., 9.4999.jpg

讀取圖檔



1.載入函示庫

2.預留資料空間

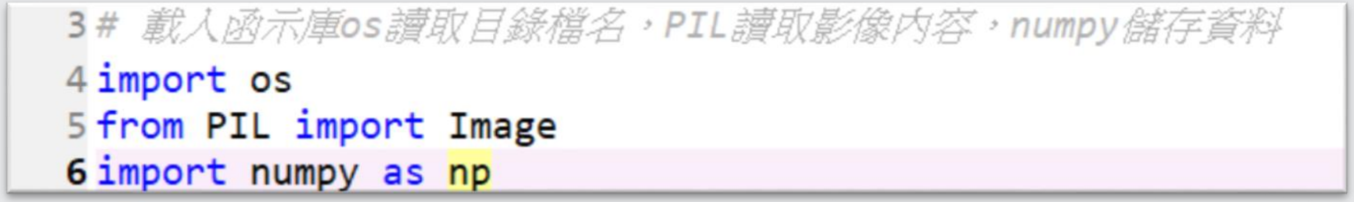
3.讀取訓練圖片內容及label

4.讀取測試圖片內容及label

5.回傳切割結果

讀取圖檔

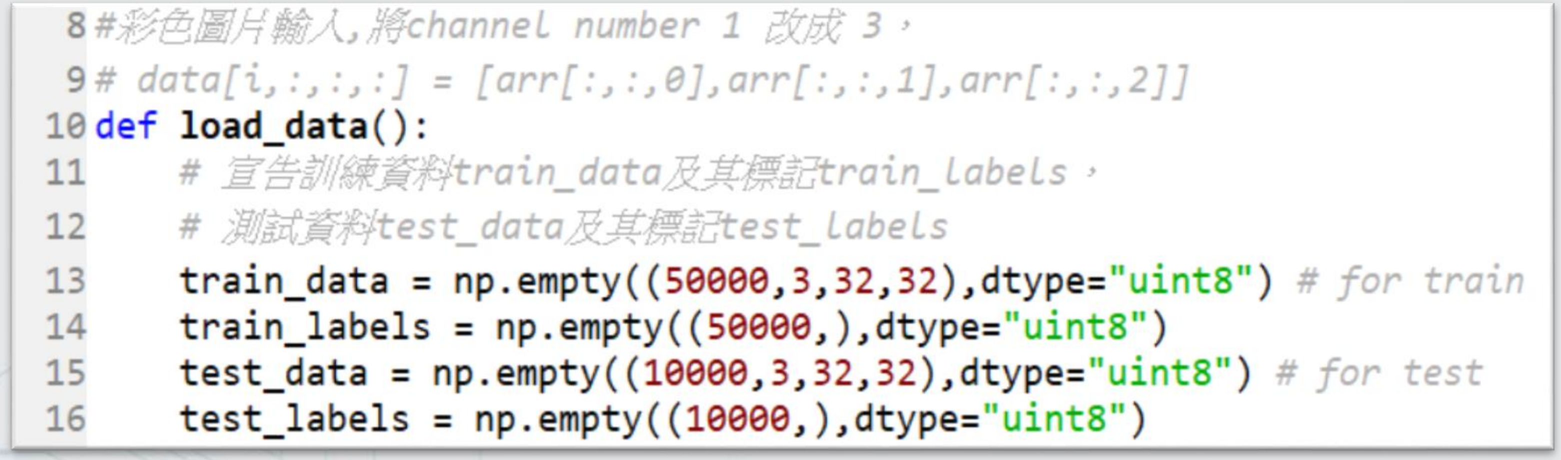
- 載入函示庫



```python
# 載入函示庫os讀取目錄檔名，PIL讀取影像內容，numpy儲存資料
import os
from PIL import Image
import numpy as np
```

讀取圖檔

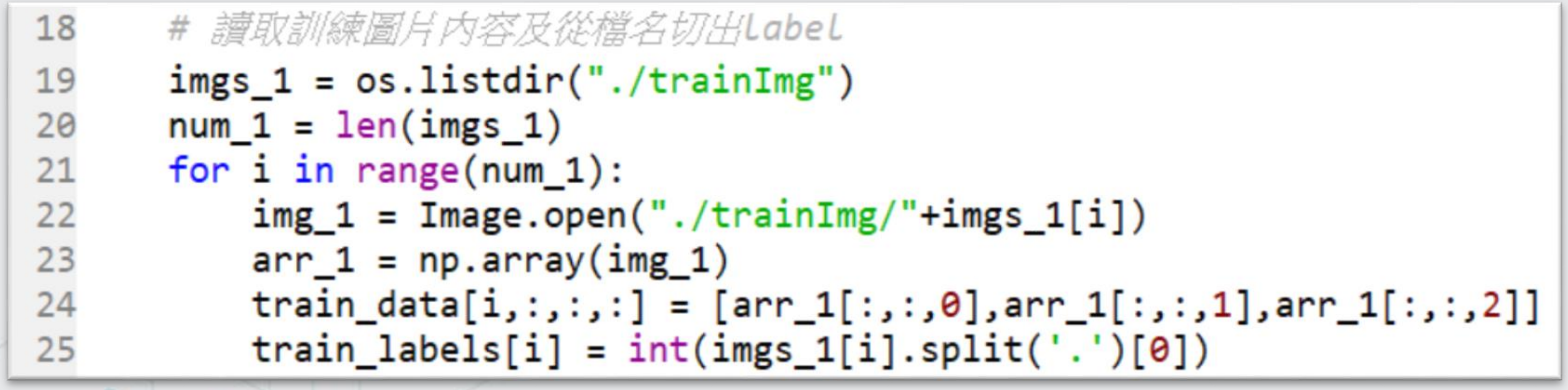
- 預留資料空間



```python
#彩色圖片輸入,將channel number 1 改成 3，
# data[i,:,:,:] = [arr[:,:,0],arr[:,:,1],arr[:,:,2]]
def load_data():
    # 宣告訓練資料train_data及其標記train_labels，
    # 測試資料test_data及其標記test_labels
    train_data = np.empty((50000,3,32,32),dtype="uint8") # for train
    train_labels = np.empty((50000,),dtype="uint8")
    test_data = np.empty((10000,3,32,32),dtype="uint8") # for test
    test_labels = np.empty((10000,),dtype="uint8")
```

讀取圖檔

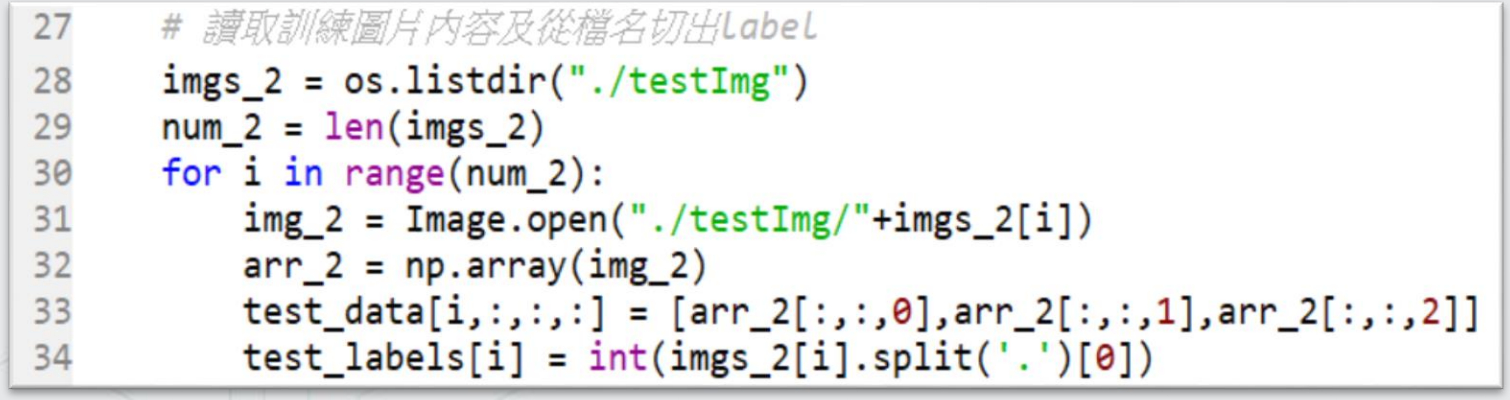
- 讀取訓練圖片內容及label



```python
# 讀取訓練圖片內容及從檔名切出label
imgs_1 = os.listdir("./trainImg")
num_1 = len(imgs_1)
for i in range(num_1):
    img_1 = Image.open("./trainImg/"+imgs_1[i])
    arr_1 = np.array(img_1)
    train_data[i,:,:,:] = [arr_1[:,:,0],arr_1[:,:,1],arr_1[:,:,2]]
    train_labels[i] = int(imgs_1[i].split('.')[0])
```

讀取圖檔

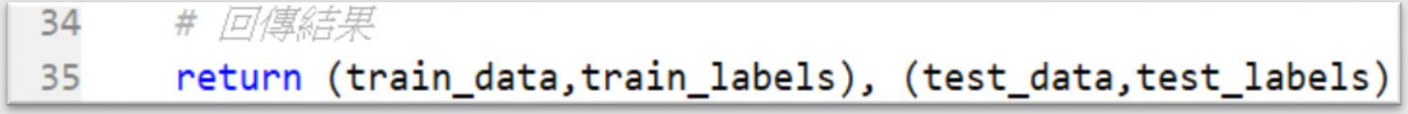
- 讀取測試圖片內容及label



```python
# 讀取訓練圖片內容及從檔名切出label
imgs_2 = os.listdir("./testImg")
num_2 = len(imgs_2)
for i in range(num_2):
    img_2 = Image.open("./testImg/"+imgs_2[i])
    arr_2 = np.array(img_2)
    test_data[i,:,:,:] = [arr_2[:,:,0],arr_2[:,:,1],arr_2[:,:,2]]
    test_labels[i] = int(imgs_2[i].split('.')[0])
```

讀取圖檔

- 回傳切割結果



```
    # 回傳結果
    return (train_data,train_labels), (test_data,test_labels)
```