

Made_By_Chun.py



[NQU] 手把手深度學習基礎教學

運用Colab雲端開發環境及PyTorch深度學習套件開發

=====

講義連結

<https://reurl.cc/VXgGbN>

技術總攬

DeepLearning、PyTorch、PythonSkill

PreTrainedModel、TransferLearning

TeachableMachine、Colab、GoogleDriveTool



近千行程式碼、百多頁投影片、數十個檔案、約五個小時、就問你怕不怕



Copy 到自己的雲端硬碟 – 安裝工具

Google Workspace Marketplace

Copy, URL to Google Drive™

以下查詢的搜尋結果： Copy, URL to Google Drive™

Cloud Copy 已安裝

URL to Google Drive

Copy, URL to Google...
3.8 ★★★★☆ (446)
53,922

Google Workspace Marketplace

搜尋「apps」

Copy, URL to Googl...
softgateon.herokuapp.com/urltodrive
雲端硬碟外掛程式

★★★★☆ (446) · 53,922

安装



Copy 到自己的雲端硬碟 – 使用工具

與我共用 > 工作坊 > 1091120_NQU_AI

垃圾桶功能有所異動。

名稱	操作
figure	
images	
labels	
tm	
transfer	
00_StreamVideo.ipynb	
01_TM2.ipynb	
02_Classification.ipynb	
03_CompareModel.ipynb	

選擇開啟工具

- 新資料夾
- 共用
- 取得連結
- 在雲端硬碟中新增捷徑
- 移至
- 新增至「已加星號」專區
- 重新命名
- 變更顏色
- 在「1091120_NQU_AI」中搜尋
- 下載
- 移除

Copy, URL to Google Drive

+ 連結更多應用程式



Copy 到自己的雲端硬碟 – 認證

Copy, URL to Google Drive



This is a tool that allows you to save or unzip, uncompress any URL (document, file..) directly to Google Drive. This app use a Google Apps Script hosting on Google clouds. It's fast and reliable. Bypass your slow internet connection. This apps allow you to do an URL uploading to Google Drive. In first you have to accept the policy access. This is the standard procedure by google for google script.

Provides connect with Google Drive. If you select a file from Google Drive, you can use it just like "Make a copy", "Copy Folder" or Unzip, Un7z, Untar to any folder.

Select a file, folder from Google Drive

Ready...

File URL

Maximum fetch size for external file URL: 50 M (except Drive files)

Target Folder Root Folder

FileName
to be saved

Use Uncompress, Save to Google Drive Uncompress Format zip (default)

Folder Name to extract a archive file. Subfolder Name of Target Folder(above)

Login & Authorize

To use this app, Please login to the Google Drive and authorize
(Note: If your browser block or disable the third-party cookie)

Copy, URL to Google Drive

This is a tool that allows you to save or unzip, uncompress any URL (document, file..) directly to Google Drive. This app use a Google Apps Script hosting on Google clouds. It's fast and reliable. Bypass your slow internet connection. This apps allow you to do an URL uploading to Google Drive. In first you have to accept the policy access. This is the standard procedure by google for google script.

Provides connect with Google Drive. If you select a file from Google Drive, you can use it just like "Make a copy", "Copy Folder" or Unzip, Un7z, Untar to any folder.

Select a file, folder from Google Drive

(Copy Folder) Show this folder

1091120_NQU_AI

Target Folder images

Save, Copy to Google Drive



選擇Root Folder & Save...

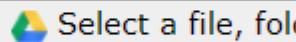
耗時：1分鐘 · 1451 → 1452

Copy, URL to Google Drive



This is a tool that allows you to save or unzip, uncompress any URL (document, file..) directly to Google Drive. This app use a Google Apps Script hosting on Google clouds. It's fast and reliable. Bypass your slow internet connection. This apps allow you to do an URL uploading to Google Drive. In first you have to accept the policy access. This is the standard procedure by google for google script.

Provides connect with Google Drive. If you select a file from Google Drive, you can use it just like "Make a copy", "Copy Folder" or Unzip, Un7z, Untar to any folder.



Select a file, folder from Google Drive

New URL

(Copy Folder) Show this folder

1091120_NQU_AI

Target Folder Root Folder



Save, Copy to Google Drive

Select Target Folder

Root Folder



更換成自己愛的名稱

名稱 ↑

Copy of 1091120_NQU_AI

custom

old

yolov5

重新命名

Blabla

取消

確定

lecturer.py



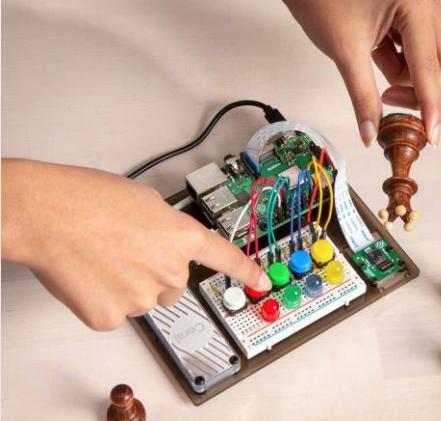
張嘉鈞 (Chun)

CAVEDU教育團隊 講師

p513817@cavedu.com



AI教育可以做什麼



**不論你想要多邊緣，
我們
都會支持**

Robot 單機器人王國





Outline

- AI 入門應該怎麼做？ (20 min
 - Trained from scratch 、 Use Pre-Trained Model 、 Transfer Learning
- 從頭開始訓練 – 更簡單的方式 (60 min
 - Teachable Machine 、 Colab
- 用別人訓練好的也可以 - 運用 Pre-Trained Model (90 min
 - Classification 、 Object Detection 、 Segmentation
- 站在巨人的肩膀上 – Transfer Learning (90 min
 - Transfer Learning base on YOLOv5

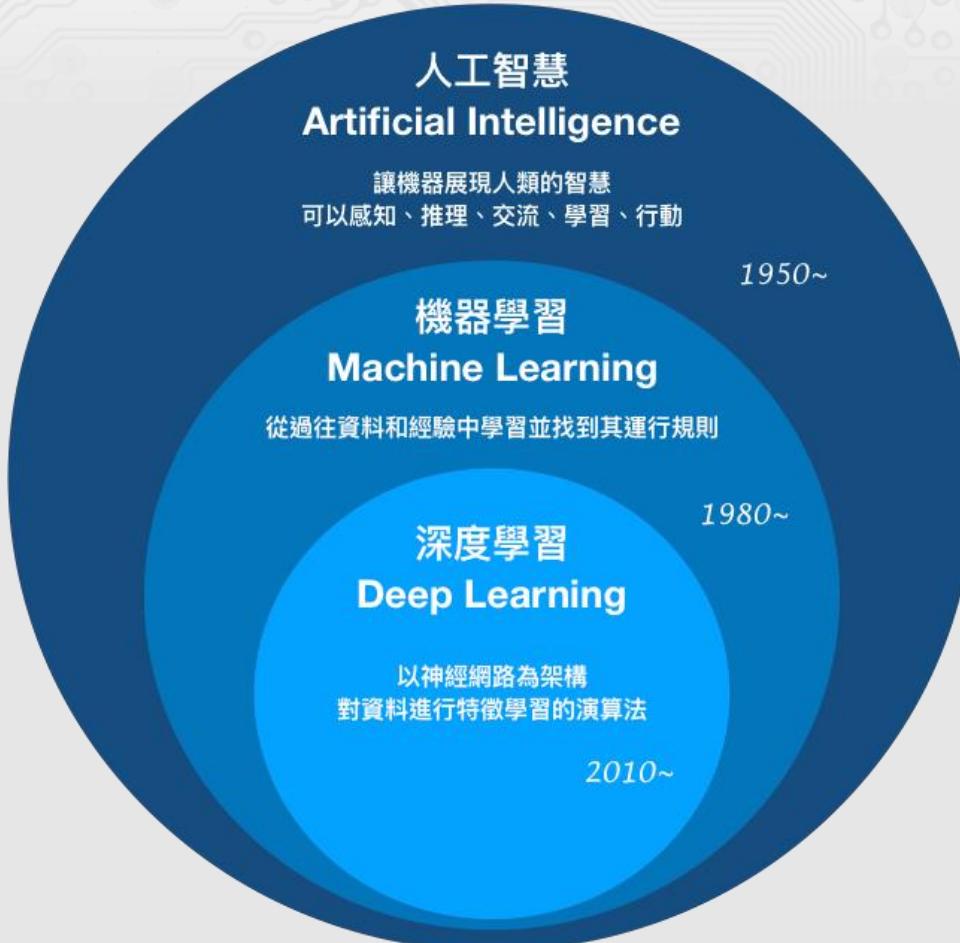
CAVEDU

AI 入門應該怎麼做？

很多種玩法可以嘗試



人工智慧、機器學習及深度學習





AI 與影像





AI 與影像 – 競賽

IM²GENET Large Scale Visual Recognition Challenge (ILSVRC)

Competition

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) evaluates algorithms for object detection and image classification at large scale. One high level motivation is to allow researchers to compare progress in detection across a wider variety of objects -- taking advantage of the quite expensive labeling effort. Another motivation is to measure the progress of computer vision for large scale image indexing for retrieval and annotation.

For details about each challenge please refer to the corresponding page.

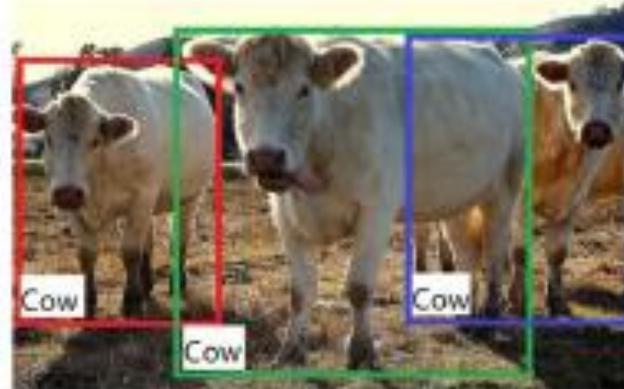
- [ILSVRC 2017](#)
- [ILSVRC 2016](#)
- [ILSVRC 2015](#)
- [ILSVRC 2014](#)
- [ILSVRC 2013](#)
- [ILSVRC 2012](#)
- [ILSVRC 2011](#)
- [ILSVRC 2010](#)



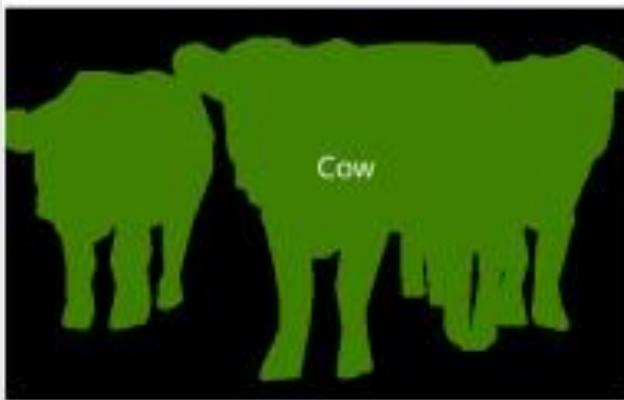
AI 在圖像上的應用



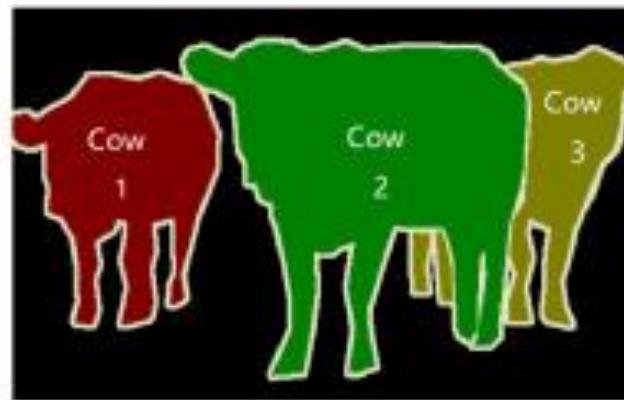
(a) Image Classification



(b) Object Detection



(c) Semantic Segmentation

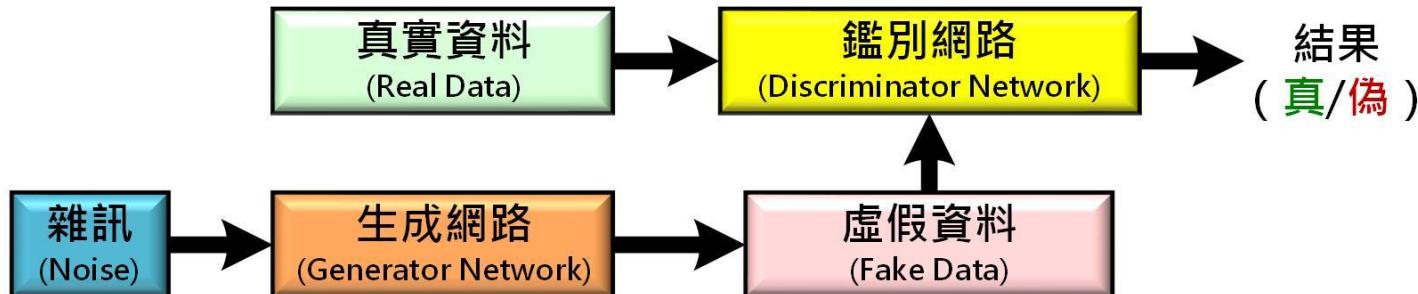


(d) Instance Segmentation



AI 在圖像上的應用

生成對抗網路 (Generative Adversarial Network)



成年→兒童



女生→男生

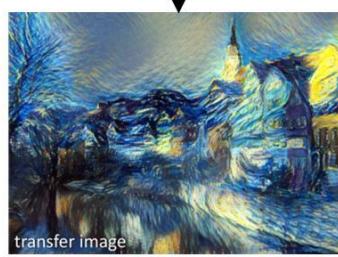


成年→老年



真實→卡通

人臉生成(變臉APP)



風格轉換



圖像修復(填補/上色/去糊)

我們從簡單的開始

Classification、Object Detection



那我應該要怎麼入門比較好？

基礎

數學觀念、大部分都是用 Python 跟 C 、熟悉 AI 框架、數據處理的方法

建構神經網路模型

理解數學公式、了解不同的模型差異、稍微會算微積分、看到數學公式不會暈



那我應該要怎麼入門比較好？

- <https://www.coursera.org/specializations/deep-learning>

深度学习 專項課程

Become a Deep Learning experts. Master Deep Learning and Break into AI

★★★★★ 4.8 260,817 個評分



Andrew Ng 另外 +2 位授課教師

頂尖授課教師

- <https://www.coursera.org/specializations/mathematics-machine-learning>

数学在机器学习领域的应用 專項課程

数学在机器学习领域的应用. Learn about the prerequisite mathematics for applications in data science and machine learning

★★★★★ 4.5 14,793 個評分



David Dye 另外 +3 位授課教師

我自己都覺得我還在入門階段

你要我建構一個獨特的神經網路模型
我可能沒辦法，這可會牽扯到各種技術
但至少我知道怎麼去應用它、修改它



使用AI模型常見的方式有三種

- 從頭開始訓練吧？Train from scratch
 - 程式建構模型、理解數學公式、蒐集資料、反覆訓練權重以及驗證
- 太難了？也可以用別人的模型。Pre-Trained Model
 - 導入模型架構跟權重、應用
 - 缺點是只能用別人的數據集
- 想要應用在自己的數據上有什麼好辦法呢？Transfer Learning
 - 導入模型架構、取得權重再微調

CAVEDU

要從頭開始訓練嗎？

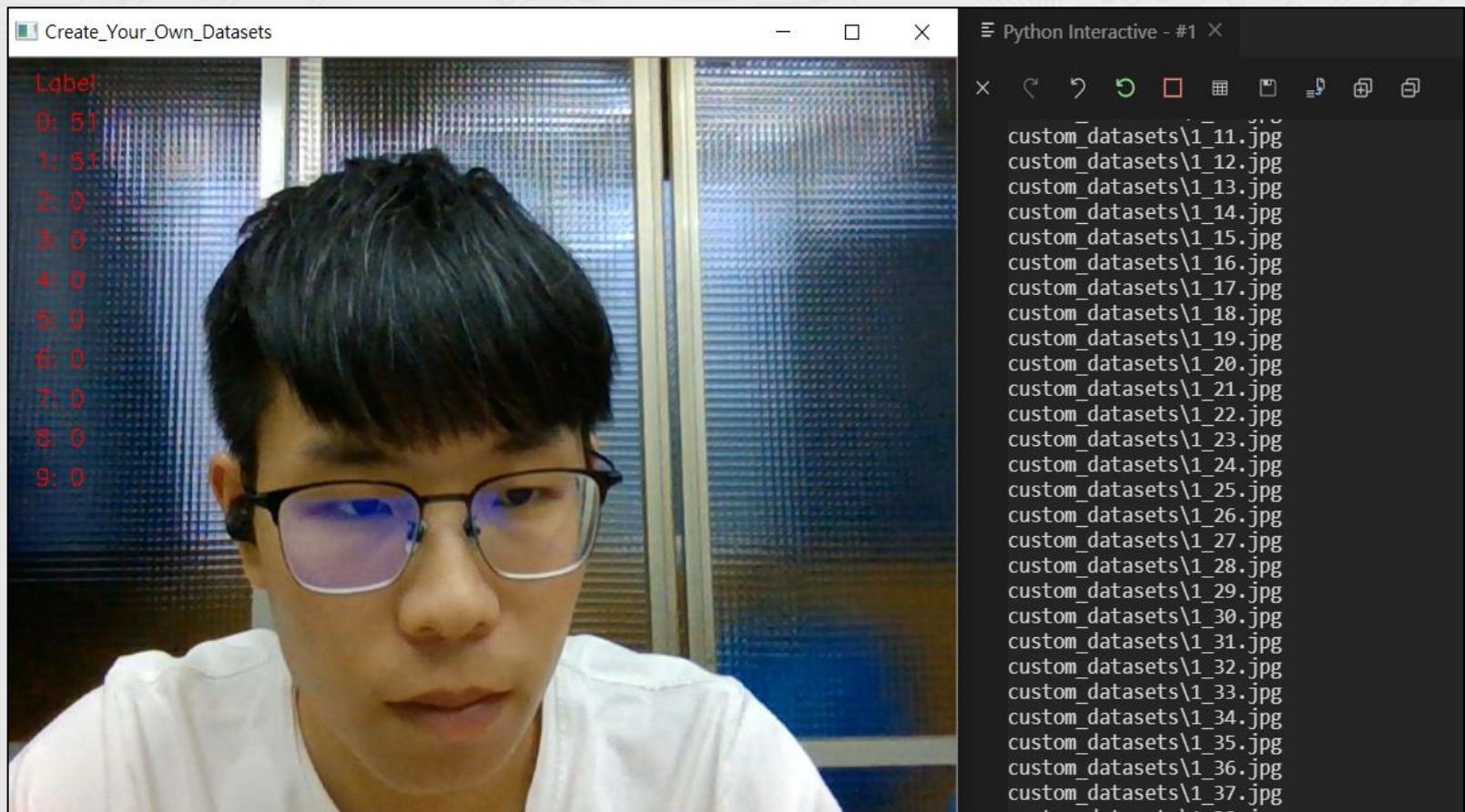


從頭開始訓練

- 蒐集資料 (拍照、下載...)
- 準備標籤 (檔名、框物件...)
- 建立神經網路 (MLP、CNN、RNN ...)
- 數據前處理(正規化、資料擴增、轉換成張量 ...)
- 訓練 (調整疊代次數、超參數 ...)
- 驗證評估 (準確度、IOU ...)
- 實際測試 (測試資料、現實中的資料 ...)
- 應用 (整合網頁、硬體裝置 ...)

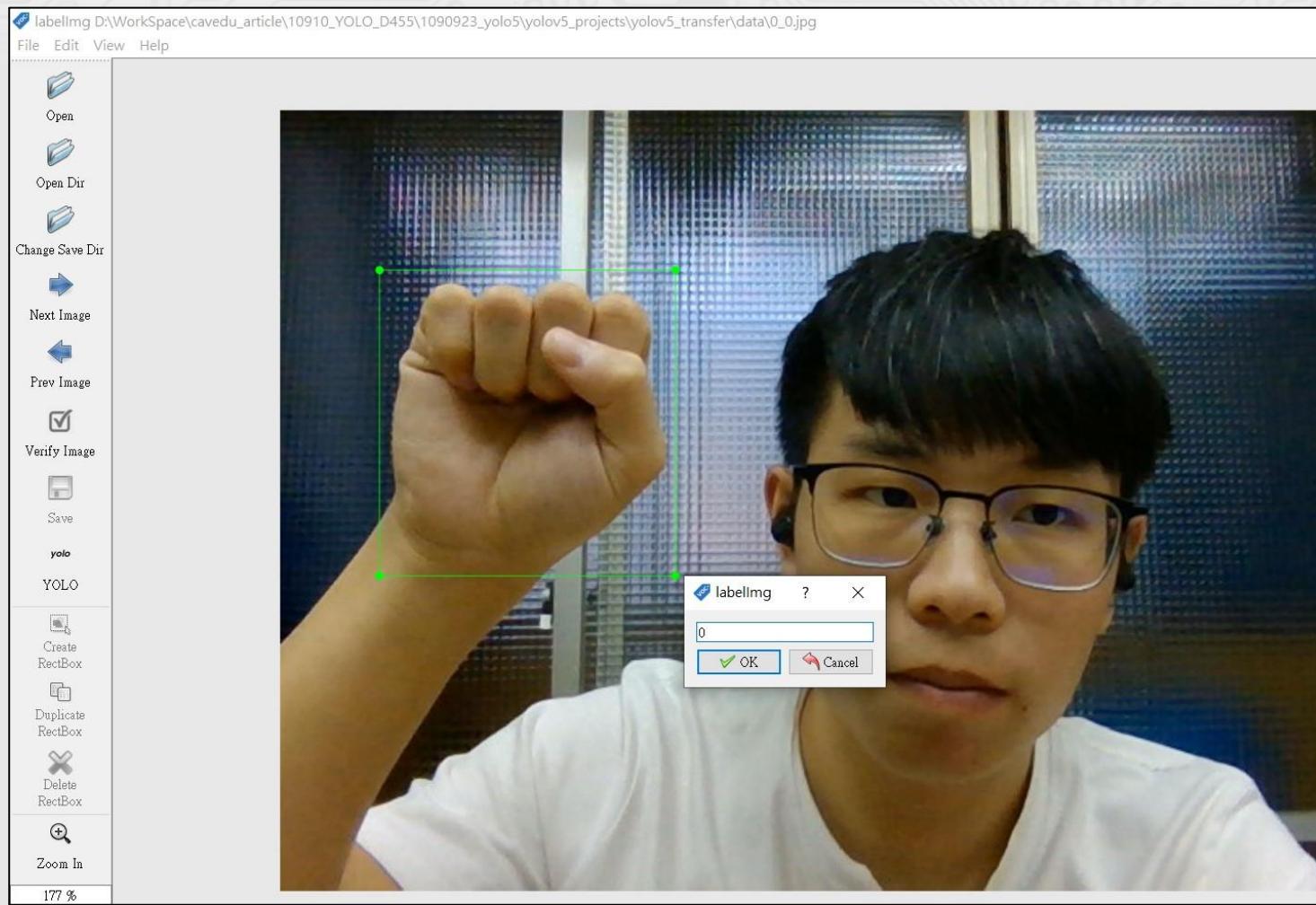


蒐集數據





製作標籤





建立模型、套用模型 (TF、Torch...)

```
  yolo.py  x
D: > WorkSpace > cavedu_article > 10910_YOLO_D455 > 1090923_yolo5 > yolov5_projects > yolov5 > models > yolo.py ...
11  from models.experimental import MixConv2d, CrossConv, C3
12  from utils.general import check_anchor_order, make_divisible, check_file, set_logging
13  from utils.torch_utils import (
14      time_synchronized, fuse_conv_and_bn, model_info, scale_img, initialize_weights, select_device)
15
16  logger = logging.getLogger(__name__)
17
18
19  class Detect(nn.Module):
20      stride = None # strides computed during build
21      export = False # onnx export
22
23      def __init__(self, nc=80, anchors=(), ch=()): # detection layer
24          super(Detect, self).__init__()
25          self.nc = nc # number of classes
26          self.no = nc + 5 # number of outputs per anchor
27          self.nl = len(anchors) # number of detection layers
28          self.na = len(anchors[0]) // 2 # number of anchors
29          self.grid = [torch.zeros(1)] * self.nl # init grid
30          a = torch.tensor(anchors).float().view(self.nl, -1, 2)
31          self.register_buffer('anchors', a) # shape(nl,na,2)
32          self.register_buffer('anchor_grid', a.clone().view(self.nl, 1, -1, 1, 1, 2)) # shape(nl,1,na,1,1,2)
33          self.m = nn.ModuleList(nn.Conv2d(x, self.no * self.na, 1) for x in ch) # output conv
34
35      def forward(self, x):
36          # x = x.copy() # for profiling
37          z = [] # inference output
38          self.training |= self.export
39          for i in range(self.nl):
40              x[i] = self.m[i](x[i]) # conv
41              bs, _, ny, nx = x[i].shape # x(bs,255,20,20) to x(bs,3,20,20,85)
42              x[i] = x[i].view(bs, self.na, self.no, ny, nx).permute(0, 1, 3, 4, 2).contiguous()
43
44              if not self.training: # inference
45                  if self.grid[i].shape[2:4] != x[i].shape[2:4]:
46                      self.grid[i] = self._make_grid(nx, ny).to(x[i].device)
```



訓練、驗證、評估

我的雲端硬碟 > ... > runs > exp17 >

資料夾

名稱 ↑

weights

檔案

事件	hyp.yaml	labels_correlogram.png	labels.png	opt.yaml	results.png
events.out.tfevents.16...					

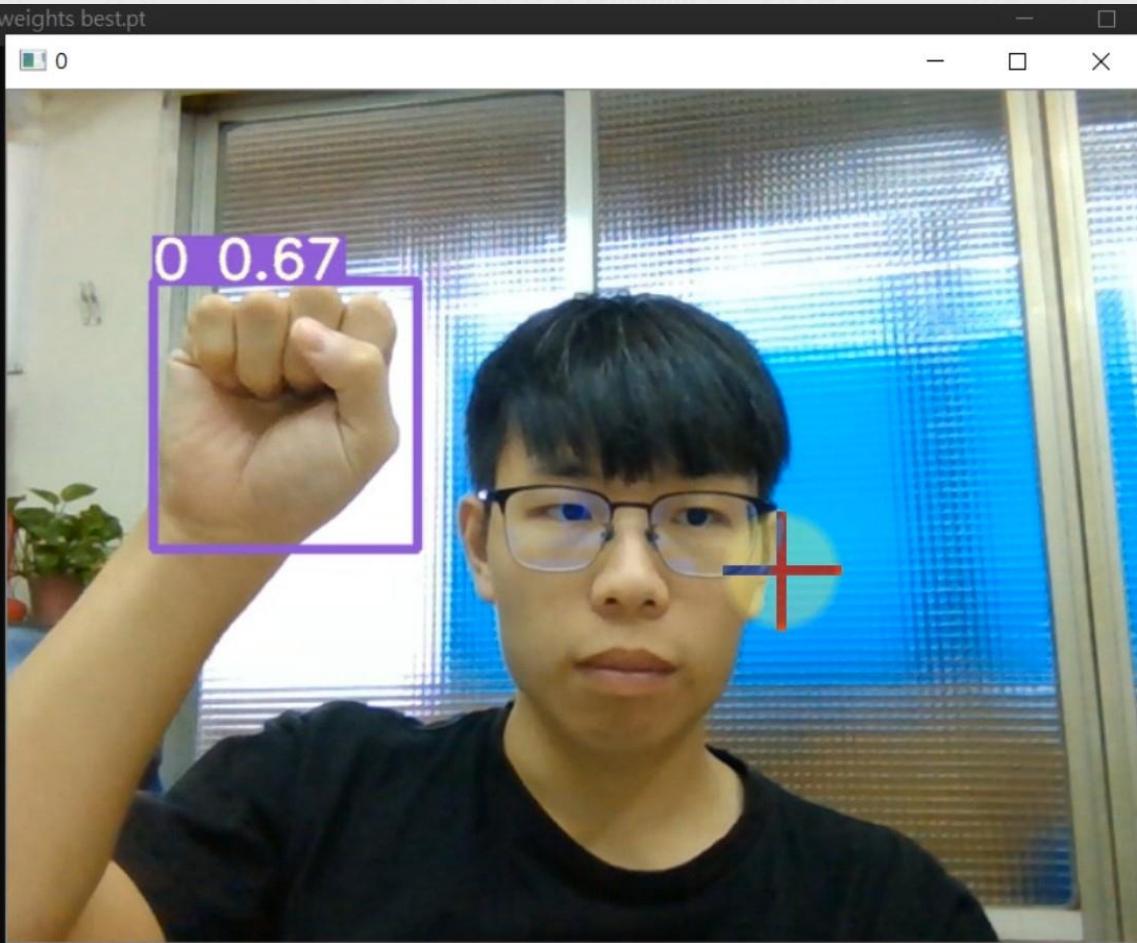
結果	test_batch0_gt.jpg	test_batch0_pred.jpg	train_batch0.jpg	train_batch1.jpg	train_batch2.jpg
results.txt					

Figure 1: A screenshot of a cloud storage interface showing the contents of a 'runs/exp17' folder. The 'weights' folder is highlighted. The interface displays various files and visualizations related to training and evaluation, including event logs, configuration files (hyp.yaml, opt.yaml), and results files (results.txt). It also shows generated images (test and train batches) and performance metrics (Objectness, Classification, Precision, mAP@0.5) plotted against validation steps.



實際測試 & 應用

```
[1]: 命令提示字元 - python detect.py --source 0 --weights best.pt
0: 512x640 1 0s, Done. (0.027s)
0: 512x640 1 0s, Done. (0.026s)
0: 512x640 1 0s, Done. (0.027s)
```



CAVEDU

有沒有更簡單的方法

Teachable Machine



Teachable Machine

- 蒐集資料 (拍照、下載...)
- 準備標籤 (檔名、框物件...)
- 建立神經網路 (MLP、CNN、RNN ...)
- 數據前處理(正規化、資料擴增、轉換成張量 ...)
- 訓練 (調整疊代次數、超參數 ...)
- 驗證評估 (準確度、IOU ...)
- 實際測試 (測試資料、現實中的資料 ...)
- 應用 (整合網頁、硬體裝置 ...)



Teachable Machine

- 利用 Teachable Machine 訓練AI模型
- 下載並上傳至雲端硬碟
- Colab 基本操作（安裝、設定、開啟相機）
- 透過 Colab 運行 AI 模型（複製程式碼）
- 排除小BUG（修改程式碼）



Teachable Machine 2

工程師

一般使用者

≡

About FAQ **Get Started**

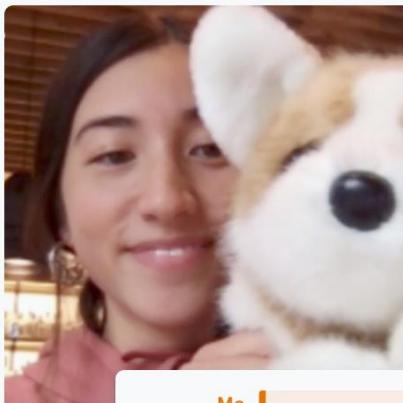
Teachable Machine

Train a computer to recognize your own images, sounds, & poses.

A fast, easy way to create machine learning models for your sites, apps, and more – no expertise or coding required.

Get Started

↑ ml5.js p5.js Coral ← node.js



Me 1%
Me + Dog <3 98%

What is Teachable Machine?



Play with Teachable Machine

New Project

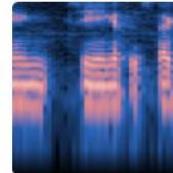
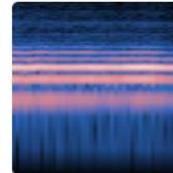
 Open an existing project from Drive.

 Open an existing project from a file.



Image Project

Teach based on images, from files or your webcam.



Audio Project

Teach based on one-second-long sounds, from files or your microphone.



Pose Project

Teach based on images, from files or your webcam.



Take a pic

≡ Teachable Machine

Chun 🖊

81 Image Samples

Webcam Upload



Coffee 🖊

Webcam X



Hold to Record ⚙

80 Image Samples



Training

Train Model

Advanced

Preview ⌂ Export Model

You must train a model on the left before you can preview it here.



Train Model

Training

Train Model

Advanced ^

Epochs: 50

Batch Size: 16

Learning Rate:
0.001

Reset Defaults

Under the hood

Preview ↑ E

You must train a model before you can preview.

Training

Train Model

Advanced ^

Epochs: 30

Batch Size: 16

Learning Rate:
0.001

Reset Defaults

Under the hood



Preview

≡ Teachable Machine

Chun

81 Image Samples

Webcam Upload

Coffee

80 Image Samples

Webcam Upload

Add a class

Training

Model Trained

Advanced

Epochs: 30

Batch Size: 16

Learning Rate: 0.001

Reset Defaults

Under the hood

Preview

Input ON Webcam

Crop

Output

Chun 100%

Coffee



Download Model

≡ Teachable Machine

Chun

81 Image Samples

Webcam Upload

Coffee

80 Image Samples

Webcam Upload

Add a class

Export your model to use it in projects.

X

Tensorflow.js [i](#) Tensorflow [i](#) Tensorflow Lite [i](#)

Model conversion type:

Keras Savedmodel [Download my model](#)

Converts your model to a keras .h5 model. Note the conversion happens in the cloud, but your training data is not being uploaded, only your trained model.

Code snippets to use your model:

Keras

Contribute on Github [G](#)

```
import tensorflow.keras
from PIL import Image, ImageOps
import numpy as np

# Disable scientific notation for clarity
np.set_printoptions(suppress=True)

# Load the model
model = tensorflow.keras.models.load_model('keras_model.h5')

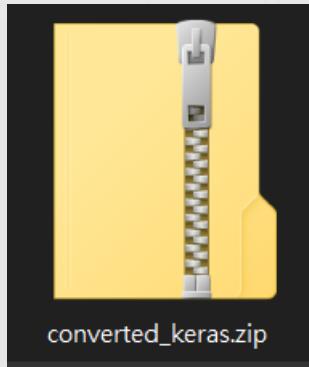
# Create the array of the right shape to feed into the keras model
# The 'length' or number of images you can put into the array is
# determined by the first position in the shape tuple, in this case 1.
data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)

# Replace this with the path to your image
```

Copy



將模型檔放進 Google Drive



我的雲端硬碟 > 人生KPI > 工作坊 > 1120_NQU_AI ▾

檔案

1_TM2.ipynb

converted_keras.zip



安裝 Colab

- 雲端硬碟任意處右鍵 > 更多 > 連結更多應用程式
- 在跳出的視窗輸入 colab 就會跳出 Colaboratory 的選項
- 點擊 icon 進去安裝

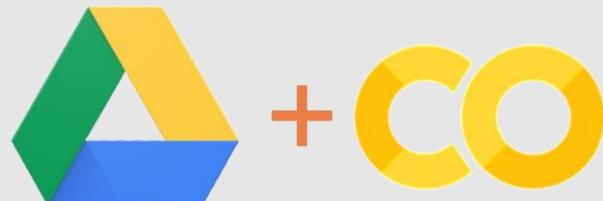
The image shows two screenshots illustrating the process of installing Colaboratory. On the left, a screenshot of the Google Drive context menu is shown, with the 'More' option expanded to reveal the 'Connect more apps' submenu. On the right, a screenshot of the Google Workspace Marketplace search results for 'Colaboratory' is displayed. The search bar at the top contains the text 'Colaboratory'. Below the search bar, the text '以下查詢的搜尋結果: Colaboratory' is visible. A card for 'Colaboratory' is shown, featuring a yellow circular logo with the letters 'CO', a 'Connected' status indicator, and a '4.7 ★★★★★ (2715)' rating. The card also displays the number '3,442,913'.



什麼是 Colab (Colaboratory) ?

Colaboratory (簡稱為「Colab」) 可讓你在瀏覽器上撰寫及執行 Python，且具備下列優點：

- 不必進行任何設定
- 免費使用 GPU
- 輕鬆共用
- 無論你是學生、數據資料學家或是 AI 研究人員，Colab 都能讓你的工作事半功倍。請觀看 [Colab 的簡介影片](#)瞭解詳情，或是直接瀏覽以下的新手入門說明！





新增 Colab 檔案

The screenshot shows the 'New' menu in Google Drive. The 'More' option is highlighted, revealing a dropdown menu with several options:

- 新資料夾
- 上傳檔案
- 上傳資料夾
- Google 文件
- Google 試算表
- Google 簡報
- Google 表單
- 更多 >
- Google 協作平台
- Copy, URL to Google Drive
- Google Apps Script
- Google Colaboratory
- Google Jamboard

The 'Google Colaboratory' option is also highlighted in the dropdown menu.

CAVEDU

實作一、TM2

01_TM2.ipynb



查看GPU使用狀況 - CPU

```
import torch
import torchvision
import numpy as np
import cv2

print("GPU IS WORKED ? " , torch.cuda.is_available())

if torch.cuda.is_available():
    info = torch.cuda.get_device_properties(0)
    print(info)
```

GPU IS WORKED ? False



設定筆記本

A screenshot of a Jupyter Notebook interface. The title bar shows "Untitled0.ipynb". The "Edit" menu is open, displaying various options like "Cut", "Copy", and "Paste". A sub-menu for "Cell" is also visible, containing options like "Run Cell" and "Cell Kernel". The main workspace shows some code cells and output.

CO Untitled0.ipynb ☆

檔案 編輯 檢視畫面 插入 執行階段 工具 說明

+ 程式碼 復原 Ctrl+M Z

重做 Ctrl+Shift+Y

選取所有儲存格 Ctrl+Shift+A

剪下儲存格或選取範圍

複製儲存格或選取範圍

貼上

刪除所選儲存格 Ctrl+M D

尋找與取代 Ctrl+H

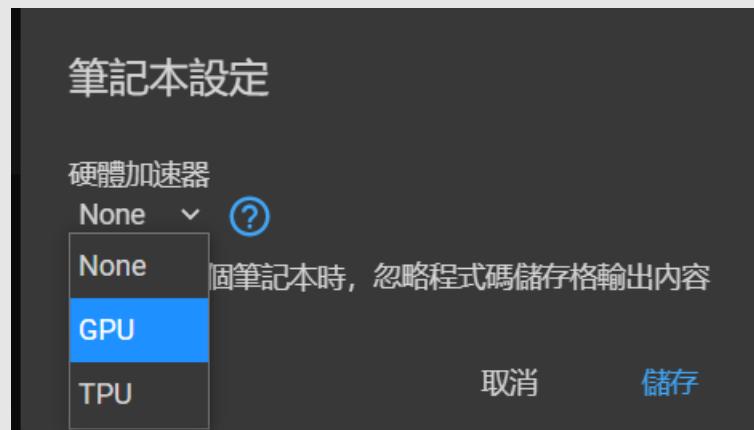
尋找下一個項目 Ctrl+G

尋找上一個項目 Ctrl+Shift+G

筆記本設定

顯示/隱藏程式碼

清除所有輸出內容





查看GPU使用狀況 - GPU

```
import torch
import torchvision
import numpy as np
import cv2

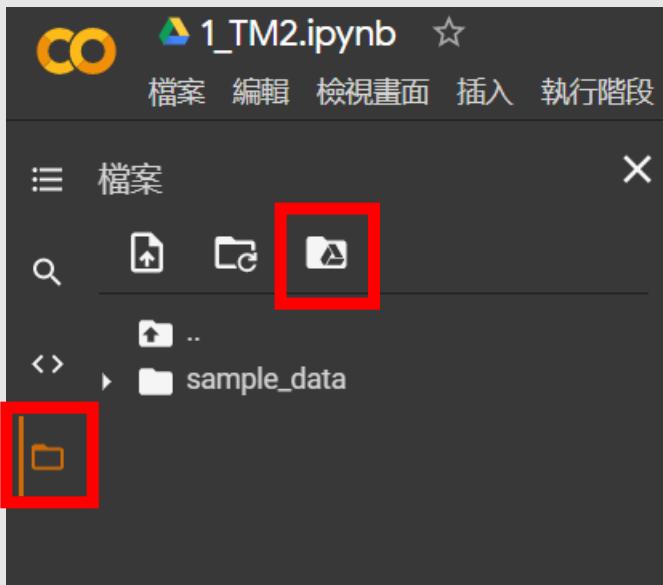
print("GPU IS WORKED ? " , torch.cuda.is_available())

if torch.cuda.is_available():
    info = torch.cuda.get_device_properties(0)
    print(info)
```

```
GPU IS WORKED ? True
_CudaDeviceProperties(name='Tesla P4', major=6, minor=1, total_memory=7611MB, multi_processor_count=20)
```



掛接到Google Drive





新建一個資料夾並複製該位置



新增完

記得重新整理

- ▶ 1091020_TKU_Line...
- ▶ 1091023_TKU_Jets...
- ▶ 1091024_AIoT_福...
- ▶ 1091104_NIU_玩轉...
- ▶ 1102_AI人工智慧應...
- ▶ 1120_NQU_AI
- ▶ AICV_Workshop
- ▶ AI自駕車程式開發...
- ▶ Ai-Self-Car 工作坊 (...)
- ▶ Python 深度學習研...
- ▶ python與tensorflow...
- ▶ 小夜燈工作坊
- ▶ 金門ai工作坊
- ▶ 比賽
- ▶ 演講

import os

%pwd

path = ''

os.chdir(path)

上傳

刪除資料夾

重新命名檔案

新增檔案

新增資料夾

複製路徑

重新整理



移動 Colab 路徑

```
### 移動Colab的方法有兩種
# 1.Command: %cd
%cd /content/drive/My Drive

# 2.Ipython: os.chdir()
import os
path = '/content/drive/My Drive/人生KPI/工作坊
/1120_NQU_AI'
os.chdir(path)

# 在ipython環境中建議使用 ! 會比較好 % 用來移動路徑即可
!pwd
!ls
```

```
/content/drive/My Drive
/content/drive/My Drive/人生KPI/工作坊/1120_NQU_AI
1_TM2.ipynb stream_video.ipynb
```



解壓縮 converted_keras.zip

- !unzip converted_keras.zip

```
Archive: converted_keras.zip
extracting: keras_model.h5
extracting: labels.txt
```

A screenshot of a file explorer interface. At the top, there's a dark header bar with a back arrow, a file icon labeled "labels.txt", and a search bar containing the placeholder text "在雲端硬碟中...". Below the header, on the left, is a circular "新增" (Add) button with a plus sign. The main area shows the contents of the "converted_keras.zip" archive. A white card displays the extracted files:

- 0 BlackTea
- 1 WaterBottle
- 2 Other



導入Colab拍照模塊

程式碼片段

Filter code snippets

- Adding form fields →
- Camera Capture →**
- Cross-output communication →
- display.Javascript to execute Java... →
- Downloading files or importing dat... →

Camera Capture 插入

Using a webcam to capture images for processing on the runtime.

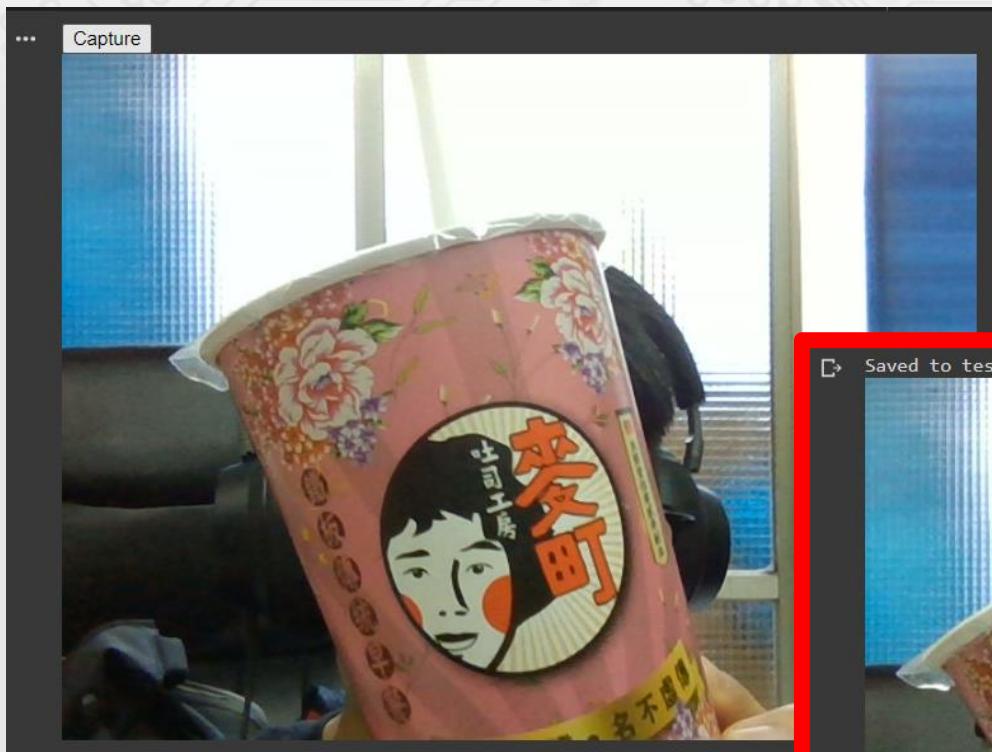
Colab 拍照模塊

```
from IPython.display import display, Javascript
from google.colab.output import eval_js
from base64 import b64decode

def take_photo(filename='test_photo.jpg', quality=0.8):
    js = Javascript('''
        async function takePhoto(quality) {
            const div = document.createElement('div');
            const capture = document.createElement('button');
            capture.textContent = 'Capture';
            div.appendChild(capture);
    ''')
    display(display.Javascript(js))
    eval_js('takePhoto({})'.format(quality))
```



執行拍照模塊 – Capture 拍照





回到 TM 複製貼上

Export your model to use it in projects.

X

Tensorflow.js ⓘ

Tensorflow ⓘ

Tensorflow Lite ⓘ

Model conversion type:

Keras

Savedmodel

[Download my model](#)

Converts your model to a keras .h5 model. Note the conversion happens in the cloud, but your training data is not being uploaded, only your trained model.

Code snippets to use your model:

Keras

Contribute on Github ↗

```
import tensorflow.keras
from PIL import Image, ImageOps
import numpy as np

# Disable scientific notation for clarity
np.set_printoptions(suppress=True)

# Load the model
model = tensorflow.keras.models.load_model('keras_model.h5')

# Create the array of the right shape to feed into the keras model
# The 'length' or number of images you can put into the array is
# determined by the first position in the shape tuple, in this case 1.
data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)

# Replace this with the path to your image
```

Copy ↗

CAVEDU

詳解程式碼



流程 (TF)

- 導入AI模型
- 開啟圖片
- 前處理：將數據資料變形、轉換型態、裁切等等
- 將圖片丟進模型中獲得預測結果
- 取得最大值對應到標籤中



程式碼 - 1

```
import tensorflow.keras
from PIL import Image, ImageOps
import numpy as np

# 設定 numpy 顯示的內容
np.set_printoptions(suppress=True)

# 導入模型
model = tensorflow.keras.models.load_model('keras_model.h5')

# 設定輸入的資料型態
data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)

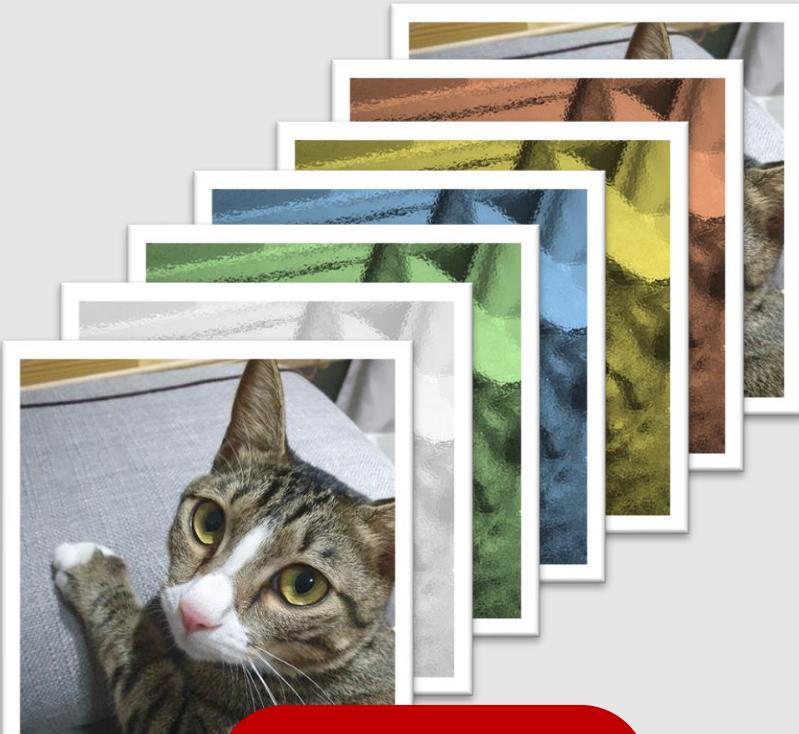
# 輸入圖片
image = Image.open('test_photo.jpg')

# 設定模型大小
size = (224, 224)
image = ImageOps.fit(image, size, Image.ANTIALIAS)
```

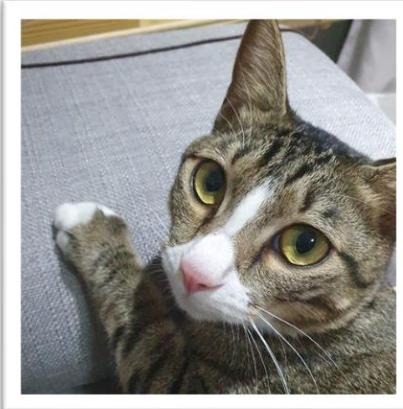


輸入的資料型態

批次大小, 通道數, 寬, 高



7, 3, 224, 224



3, 224, 224



1, 3, 224, 224



程式碼 - 2

```
# 轉換成 numpy 格式
image_array = np.asarray(image)

# 顯示
image.show()

# 正規化
normalized_image_array = (image_array.astype(np.float32) / 127.0) - 1

# 將圖片資訊填入資料型態
data[0] = normalized_image_array

# 進行預測 (推論)
prediction = model.predict(data)

# 打印結果
print(prediction)
```



直接執行 TM 程式

```
WARNING:tensorflow:No training configuration found (at least one of   
[[0.99929583 0.00010437 0.00059982]])
```

1

載下來的模型沒有包含
訓練用的超參數
(可以無視它)

2

0 BlackTea
1 WaterBottle
2 Other

0.9992
0.0001
0.0005

練習、修改 TM 程式

01_TM2.ipynb
新增一個 Cell



修改 TM 程式

```
# 導入模型  
model = tensorflow.keras.models.load_model('keras_model.h5', compile=False)
```

```
# 顯示 image.show()  
from IPython.display import display  
display(image)
```

```
# 打印結果  
f = open('labels.txt')  
label = f.readlines()  
print('res: ', label[np.argmax(prediction[0])])
```





Pre-Trained Model

- 你現在學會用別人的平台快速建置一個神經網路了，如果你連蒐集資料都懶得蒐集，可以嘗試運用看看別人模型，缺點就是你只能被限制在該模型訓練好的類別。

Hey，你好像很厲害，我可以直接拿你來做專題嗎？



可以啊，但是我只能分辨貓跟狗哦！



CAVEDU

運用別人訓練好的模型

Pre Trained Model



什麼是 pre-trained model ?

Do you have any impression that when you try other people's Github, there will be parts like downloading Pre-Trained Model

Pre-Trained Models

The project comes with a number of pre-trained models that are available through the [Model Downloader](#) tool:

Image Recognition

Network	CLI argument	NetworkType enum
AlexNet	alexnet	ALEXNET
GoogleNet	googlenet	GOOGLENET
GoogleNet-12	googlenet-12	GOOGLENET_12
ResNet-18	resnet-18	RESNET_18
ResNet-50	resnet-50	RESNET_50
ResNet-101	resnet-101	RESNET_101
ResNet-152	resnet-152	RESNET_152
VGG-16	vgg-16	VGG-16
VGG-19	vgg-19	VGG-19
Inception-v4	inception-v4	INCEPTION_V4

(Screenshot from "Jetson Inference")



什麼是 pre-trained model ?

- Train from scratch 、 Use pre-trained model





什麼是 pre-trained model ?

- Train from scratch 、 Use pre-trained model





為什麼需要 pre-trained model ?

- You are a newbie





為什麼需要 pre-trained model ?

- Time is money





為什麼需要 pre-trained model ?

- Data is expensived

The word cloud is composed of several large, bold words: 'BIG', 'DATA', and 'VOLUME'. The word 'BIG' is at the top center, 'DATA' is below it, and 'VOLUME' is to the right. Surrounding these are numerous smaller words related to data management, storage, and processing. Some key terms include:

- DATA**: GARTNER, 3V, COMPLEX, FLOW, ASSETS FOR, FORMS.
- BIG**: HIGH DEPENDENCIES, STREAM, TIME, TRIGGER, ENHANCED CAPABILITIES, BUSINESS, SOFTWARE, DISCOVERY, WIRELESS.
- VOLUME**: INSIGHT, LOGS TYPES, HANDLE THINGS, AREAS, OUT DENSITY, PHYSICS, HUNDREDS, RESEARCH, SIMULATIONS, MICROPHONES, MAKING, NETWORKS, PROCESS, REPORT, CAPTURE, DATA PACKAGES, FINANCE, INTERNET, SCIENTISTS, SPEED, PACKAGES, GENOMICS, OPTIMIZATION, IN DATA, VELOCITY, PETABYTES, AMOUNT, RELATIONAL, METEOROLOGY, RANGE, THAT, VARIETY, STATISTICS, DOMAIN, INFORMATICS, GROWNS.
- STORAGE**: ANY BECOMES BLANKET, COLLECTION, TERM, TRANSFER, ANALYSIS, INCLUDE, SHARING, CAPTURE, CHALLENGES, MANAGEMENT, PROCESSING, USING, TUTORIAL, DIFFICULT, EXPLORE, THAT, APPLICATIONS, PROCESS, IS A, DATABASE, SIMULATION, BYTES, PROCESS, PROBLEMS, SEARCH.

Other visible words include: VELOCITY, VELOCITY, AND, INFORMATION, CURATE, TAKE, CURATE, SENSOR, DOMAIN, INFORMATICS, CONNECTOMICS, DOUG, LANEY, GOVERNMENT, MANAGE, SYSTEMS, SIZE, NEW SOURCES, ENVIRONMENTAL, TOLERABLE, CAMERAS, RANGE, THAT, VARIETY, STATISTICS, DOMAIN, INFORMATICS, GROWNS.



為什麼需要 pre-trained model ?

- 你是個新手，正要入門
- 你沒有時間從頭開始建構神經網路
- 你沒有足夠大量的數據集
 - ImageNet 有1000類別、每類都不會少於1000
 - 數據量過低容易導致過擬合 (overfitting)
- 少去優化模型的麻煩事



PyTorch提供的Models

<https://pytorch.org/docs/stable/torchvision/models.html>

The screenshot shows the PyTorch documentation website. The top navigation bar includes links for Get Started, Ecosystem, Mobile, Blog, Tutorials, Docs (which is highlighted with a red dot), Resources, and Github. A sidebar on the left lists various PyTorch components: 1.7.0 (with a dropdown arrow), Search Docs, Notes [Expand], Language Bindings, C++, Javadoc, Python API, torch, torch.nn, torch.nn.functional, torch.Tensor, Tensor Attributes, Tensor Views, torch.autograd, torch.cuda, torch.cuda.amp, and torch.backends. The main content area displays the 'TORCHVISION.MODELS' page. It starts with a brief description of the models subpackage and its applications. Below this, the 'Classification' section is introduced, stating that the subpackage contains definitions for image classification model architectures. A bulleted list follows, linking to each model: AlexNet, VGG, ResNet, SqueezeNet, DenseNet, Inception v3, GoogLeNet, ShuffleNet v2, MobileNet v2, ResNeXt, Wide ResNet, and MNASNet.

Get Started Ecosystem Mobile Blog Tutorials **Docs** Resources Github

1.7.0 ▾

Search Docs

Notes [Expand]

Language Bindings

C++

Javadoc

Python API

torch

torch.nn

torch.nn.functional

torch.Tensor

Tensor Attributes

Tensor Views

torch.autograd

torch.cuda

torch.cuda.amp

torch.backends

Docs > torchvision.models

TORCHVISION.MODELS

The `models` subpackage contains definitions of models for addressing different tasks, including: image classification, pixelwise semantic segmentation, object detection, instance segmentation, person keypoint detection and video classification.

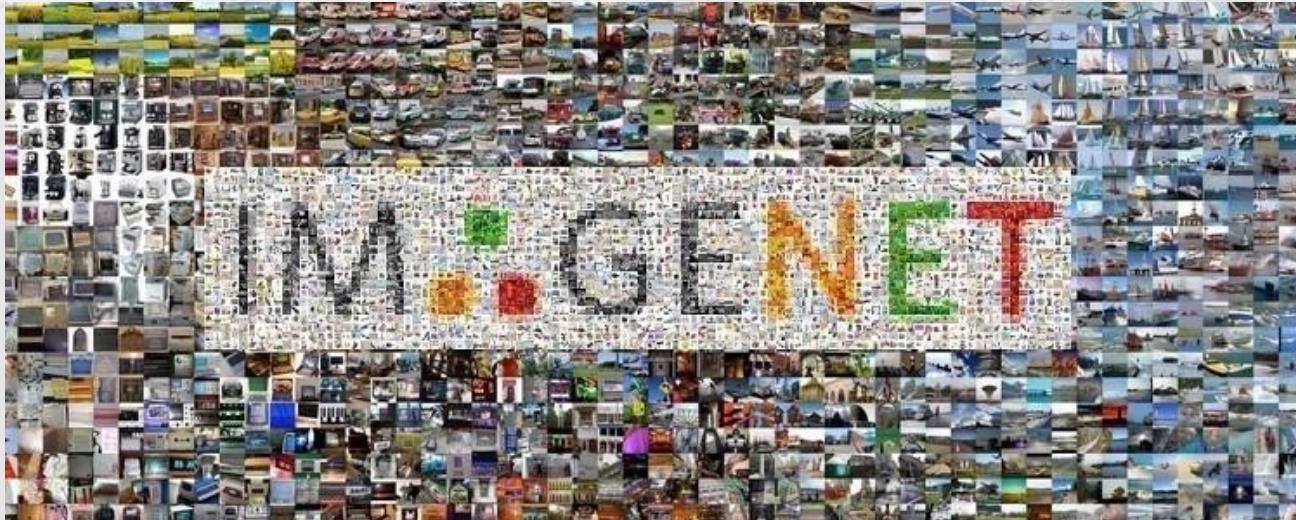
Classification

The `models` subpackage contains definitions for the following model architectures for image classification:

- [AlexNet](#)
- [VGG](#)
- [ResNet](#)
- [SqueezeNet](#)
- [DenseNet](#)
- [Inception v3](#)
- [GoogLeNet](#)
- [ShuffleNet v2](#)
- [MobileNet v2](#)
- [ResNeXt](#)
- [Wide ResNet](#)
- [MNASNet](#)



Dataset : Imagenet



ImageNet專案是一個大型視覺資料庫，用於視覺目標辨識軟體研究。該專案已手動注釋了1400多萬張圖像[1][2]，以指出圖片中的物件，並在至少100萬張圖像中提供了邊框[3]。ImageNet包含2萬多個典型類別[2]，例如「氣球」或「草莓」，每一類包含數百張圖像[4]。儘管實際圖像不歸ImageNet所有，但可以直接從ImageNet免費獲得標註的第三方圖像URL[5]。2010年以來，ImageNet專案每年舉辦一次軟體競賽，即ImageNet大規模視覺辨識挑戰賽(ILSVRC)。挑戰賽使用1000個「整理」後的非重疊類[6]，軟體程式比賽正確分類和檢測目標及場景。



將標籤檔上傳至 Google Drive

設定

完成

一般

通知

管理應用程式

儲存空間

千萬別勾

用的儲存空間

將已上傳的檔案轉換成 Google 文件編輯器格式

語言

[變更語言設定](#)

離線設定

另一位使用者 (p513817@gmail.com) 已經在這台電腦上啟用離線存取設定。
您可以在其他電腦上啟用離線存取功能，也可以在這台電腦上使用 Chrome 設定檔為您的帳戶設定離線存取功能。[瞭解詳情](#)

密度

標準

建議

在你需要時，從「快速存取」中輕鬆取得所需的檔案。

實作二、Classification

02_Classification.ipynb



起手式：找張貓咪的照片吧！

- <https://chrome.google.com/webstore/detail/save-to-google-drive/gmbmikajjgmnabiglmofigpeabaddhgne?hl=zh-TW>

[首頁](#) > [擴充功能](#) > 儲存至 Google 雲端硬碟



儲存至 Google 雲端硬碟

[加到 Chrome](#)

來源網站: drive.google.com

★★★★★ 4,641 | 實用工具 | 5,000,000+ 位使用者

適用於 Android [立即取得。](#) | G 由 Google 開發



起手式：找張貓咪的照片吧！

- <https://chrome.google.com/webstore/detail/save-to-google-drive/gmbmikajjgmnabiglmoipeabaddhgne?hl=zh-TW>





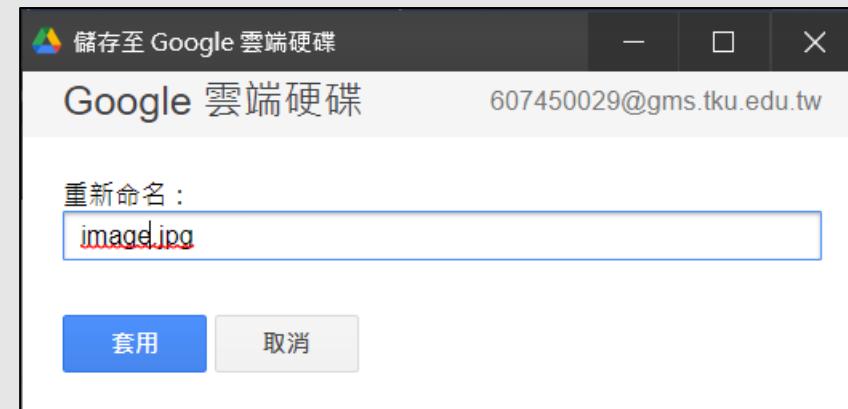
起手式：找張貓咪的照片吧！

- <https://chrome.google.com/webstore/detail/save-to-google-drive/gmbmikajjgmnabiglmoipeabaddhgne?hl=zh-TW>

The screenshot shows the 'Sign in with Google' interface. At the top, it says 'Choose an account to continue to Google Drive Chrome Extension'. Below are four account options:

- 張嘉鈞 s110310518@student.nqu.edu.tw
- 嘉鈞張 p513817@gmail.com
- 張嘉鈞 p513817@cavedu.com
- ChiaChun Chang 607450029@gms.tku.edu.tw (highlighted in blue)

At the bottom, a note states: 'To continue, Google will share your name, email address, language preference, and profile picture with Google Drive Chrome Extension.' Navigation links at the very bottom include English (United States), Help, Privacy, and Terms.





Flow chart

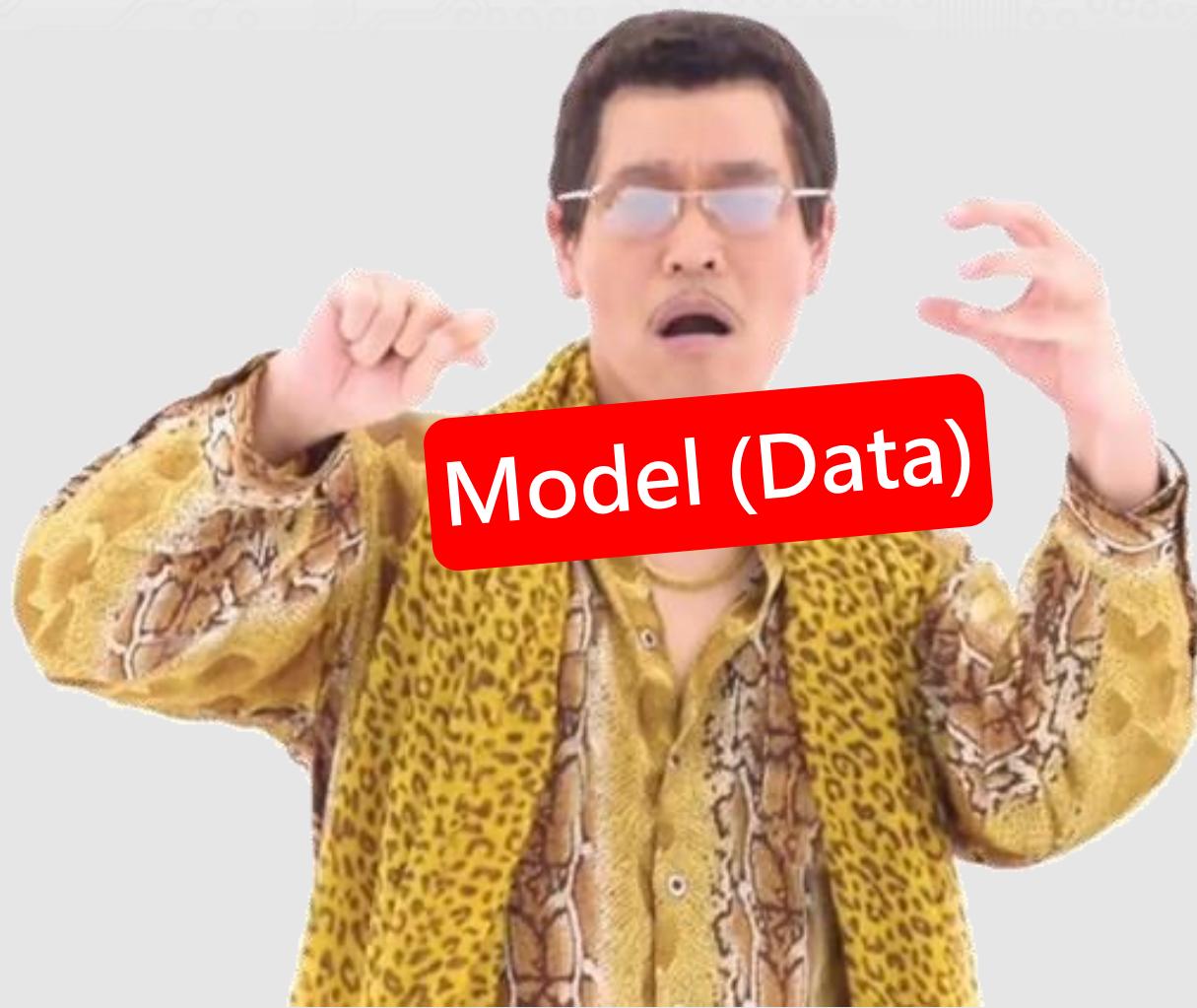
Model

Data





Flow chart





Flow chart

- Model
 - 導入 & 實例化 模型
- Data
 - 輸入資料 (PIL type)
 - 資料前處理 (scale、flip、crop、normalization、ToTensor)
 - 載入標籤檔
- Predict
 - Model(Data)
- Target Label
 - 找到最大值並對應到標籤。



先確認 Colab 位置

```
import os  
path = '/content/drive/My Drive/人生KPI/工作坊/  
1120_NQU_AI'  
os.chdir(path)  
print('Move to WorkSpace : /content/drive/My Drive/  
人生KPI/工作坊/1120_NQU_AI')
```

Move to WorkSpace : /content/drive/My Drive/人生KPI/工作坊/1120_NQU_AI



Import Model & Open Image

- You can selected pre-trained or not pre-trained

```
import torchvision.models as models
#trg_model = models.mobilenet_v2()
trg_model = models.mobilenet_v2(pretrained=True)
```

- Use PIL.Image open file

```
from PIL import Image
img_path = 'image.jpg'
img_pil = Image.open(img_path)
print(f'Org Size: {img_pil.size}')
```

Org Size: (284, 177)



Transform data

- Setup transform's format

```
from torchvision import datasets, transforms as T  
trans = T.Compose([  
    T.Resize(256),  
    T.CenterCrop(224),  
    T.ToTensor(),  
    T.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])  
])
```

- Create batchsize

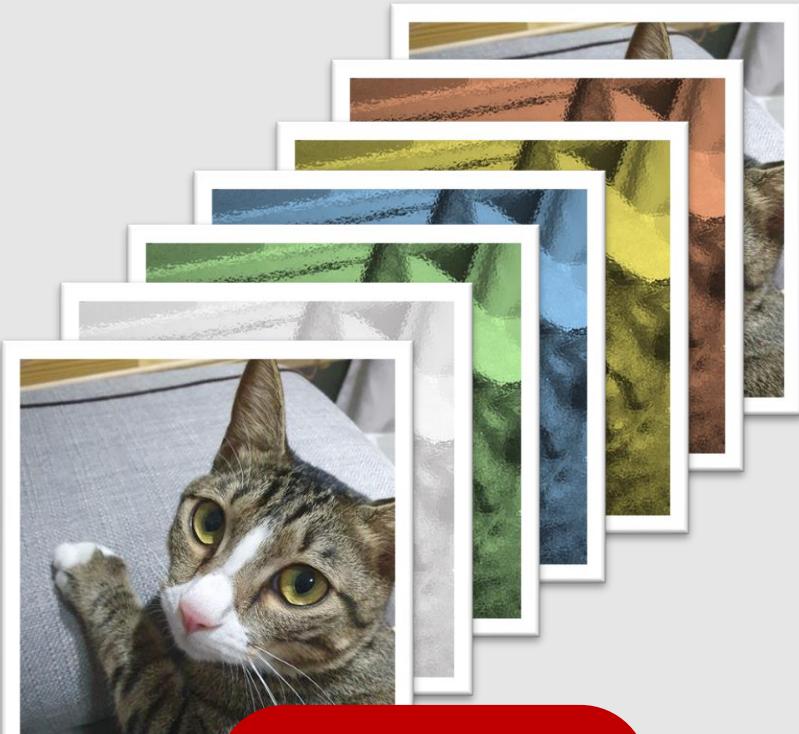
```
Tensor Size: torch.Size([3, 224, 224])  
Target Size: torch.Size([1, 3, 224, 224])
```

```
img_tensor = trans(img_pil)  
print(f"Tensor Size: {img_tensor.shape}")  
img_tensor = img_tensor.unsqueeze(0)  
print(f"Target Size: {img_tensor.shape}")
```

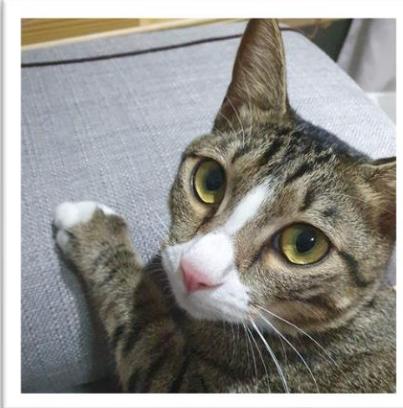


當初神經網路訓練的格式

批次大小, 通道數, 寬, 高



7, 3, 224, 224



3, 224, 224

`unsqueeze(0)`

1, 3, 224, 224



Predicted & Format

- Change Mode and Put data into model

```
import torch
from torch import nn
with torch.no_grad():

    trg_model.eval()
    predict = trg_model(img_tensor)
    print('Predict Shape: ', predict.shape)
```

- format

```
softmax = nn.Softmax()
predict_softmax = softmax(predict[0])
```

[softmax]	
Before	After
2.09276	0.00000
14.48694	0.59407
12.07950	0.05349
11.67691	0.03576
4.87094	0.00004



Get target label

- Load label file

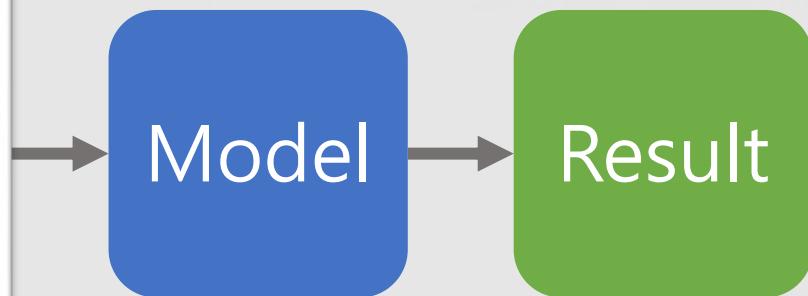
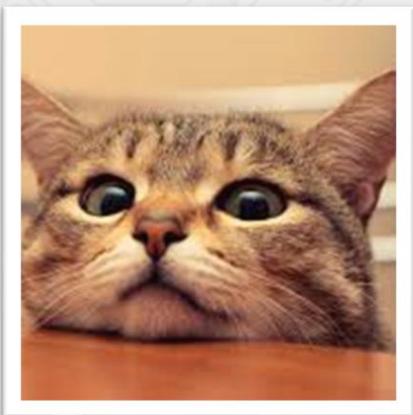
```
import numpy as np
f = open('imagenet_classes.txt', 'r')
content = f.readlines()
res = content[torch.argmax(predict_softmax)].replace('\n', ' ')
value = torch.max(predict_softmax)
print("Result | {} | {:.5f}".format(res , value))
from IPython.display import display, Image
display(Image(img_path))
```

Result | tabby, tabby cat | 0.59407
/usr/local/lib/python3.6/dist-packages





Model predicted

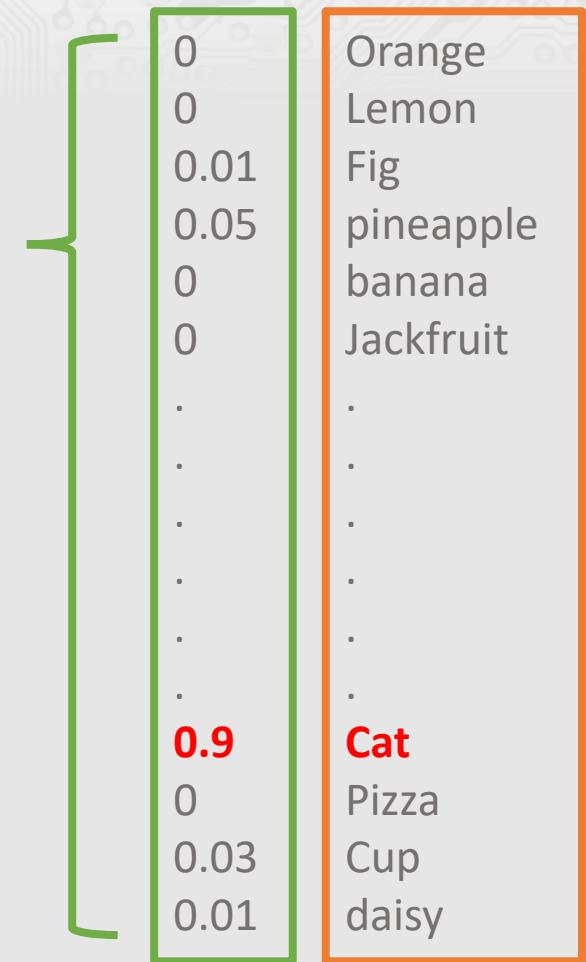


imagenet_classes.txt - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明

```
stinkhorn, carrion fungus
earthstar
hen-of-the-woods, hen of the woods,
Polyporus frondosus, Grifola frondosa
bolete
ear, spike, capitulum
toilet tissue, toilet paper, bathroom tissue
```

第 1000 列，第 45 行 | 100% | Unix (LF) | UTF-8



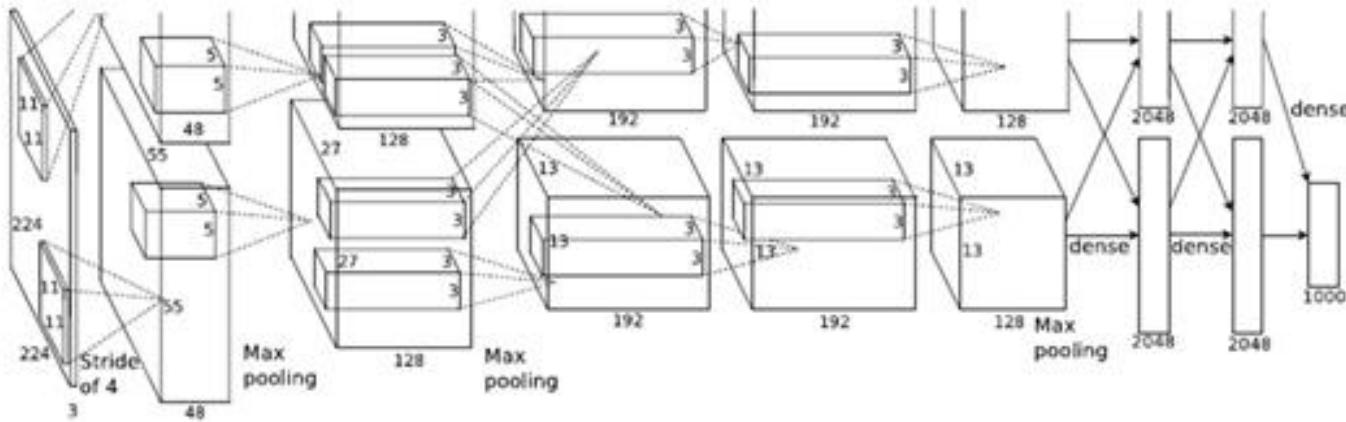
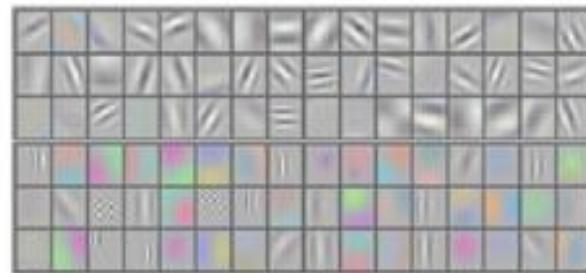
CAVEDU

Other Models



2012 大AI時代 - AlexNet

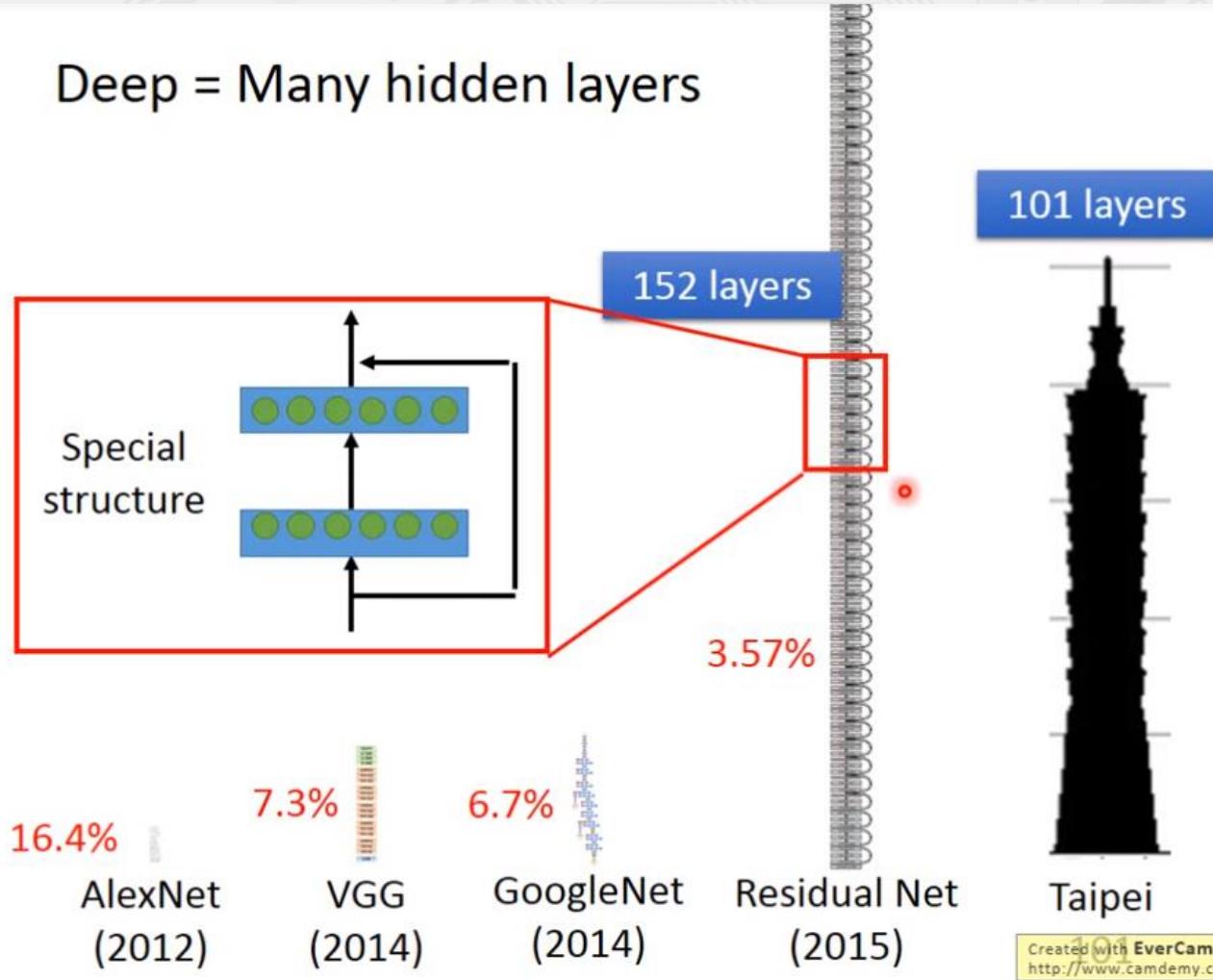
CNN经典模型 AlexNet





深，還要更深

Deep = Many hidden layers





太深的網路會有梯度消失的問題

- 權重會不見 -> 梯度消失
- 所以 Residual Net 嘗試改善這個問題

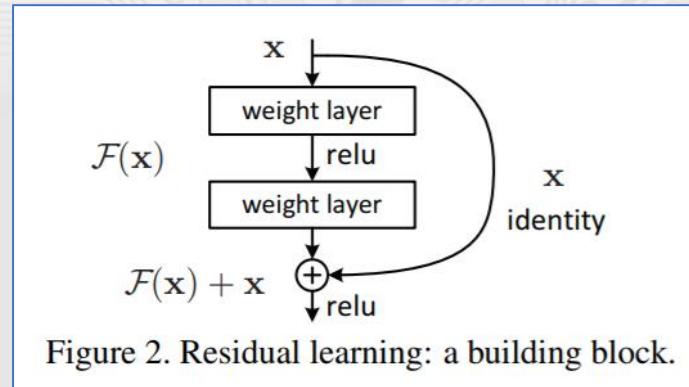
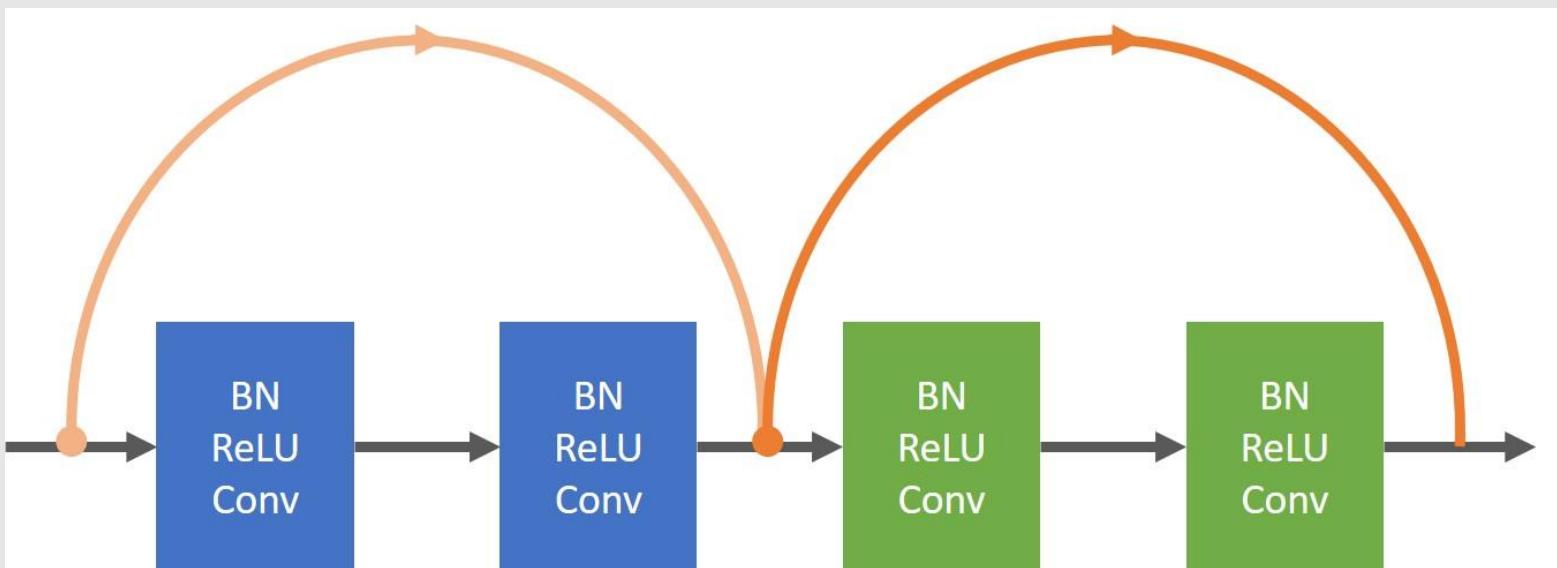


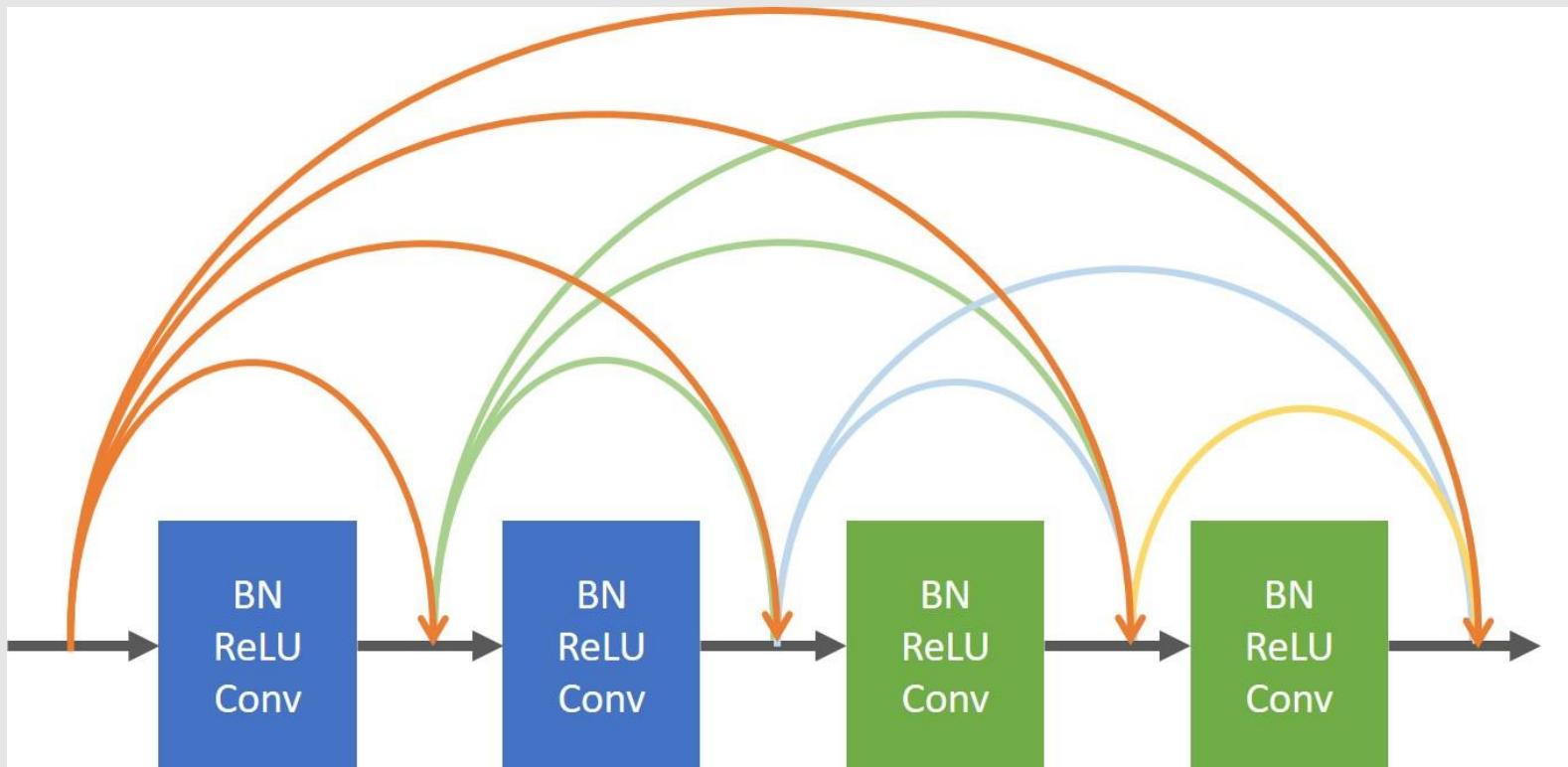
Figure 2. Residual learning: a building block.





Dense Net 加強 Residual Block

- 連結變多，但是運算沒變多





Inception v3、Mobile Net v2

- Inception的第一代就是GoogleNet，一直到今年已經有了第四代了，中間變化的過程有興趣再自行研究囉～
- Mobile Net是著名的輕量化網路，一直到現在各種邊緣裝置上都能見到他的身影，那第二代跟第一代的差異在於導入了Linear Bottleneck 和 Inverted Residual Blocks，我們今天的Pre-Trained Model也會利用Mobile Net來做實作。

練習、選兩種模型比較

03_CompareModel.ipynb
信心指數、分類是否正確



給你們10分鐘嘗試看看吧！

```
print('{:15} | {:20} | {:.5f}'.format(name, label, val))
```

mobilenet_v2	tabby, tabby cat	0.60470
vgg16	tabby, tabby cat	0.65443

```
trg_model = models.mobilenet_v2(pretrained=True).eval()
trg_model_2 = models.vgg16(pretrained=True).eval()
```

```
import torchvision.models as models
resnet18 = models.resnet18(pretrained=True)
alexnet = models.alexnet(pretrained=True)
squeezenet = models.squeeze1_0(pretrained=True)
vgg16 = models.vgg16(pretrained=True)
densenet = models.densenet161(pretrained=True)
inception = models.inception_v3(pretrained=True)
googlenet = models.googlenet(pretrained=True)
shufflenet = models.shufflenet_v2_x1_0(pretrained=True)
mobilenet = models.mobilenet_v2(pretrained=True)
resnext50_32x4d = models.resnext50_32x4d(pretrained=True)
wide_resnet50_2 = models.wide_resnet50_2(pretrained=True)
mnasnet = models.mnasnet1_0(pretrained=True)
```



探討：比較五個的結果

Google Net 信心指數只有0.349 那第二類是什麼？



A photograph of a tabby cat with brown and black stripes, sitting on a light-colored couch. The cat is looking slightly to the left of the camera.

Model	Category	Confidence Score
mobilenet_v2	tabby, tabby cat	0.60470
vgg16	tabby, tabby cat	0.65443
googlenet	tabby, tabby cat	0.34975
resnet101	tabby, tabby cat	0.61371
alexnet	tabby, tabby cat	0.71916



探討 : GoogleNet Results

```
def test(trg_model, trg_label, trg_tensor):

    with torch.no_grad():

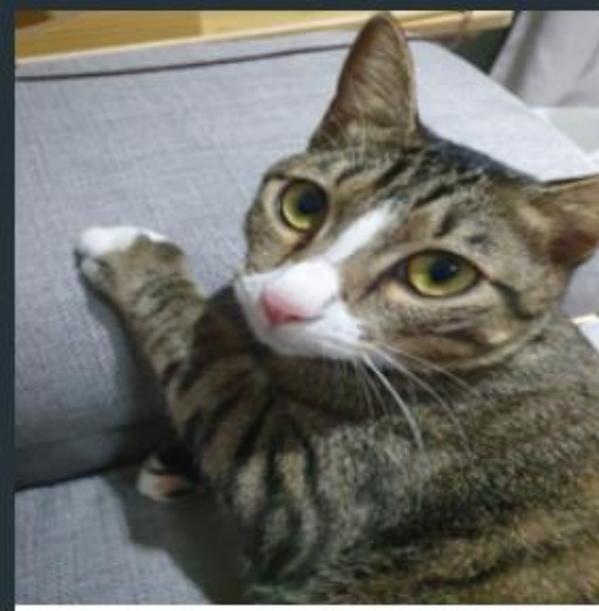
        predict = trg_model(trg_tensor)[0]

        softmax = nn.Softmax()
        predict_softmax = softmax(predict)

        for i in range(5):
            label, val = content[torch.argmax(predict_softmax)]
            print("[{}]\t{:20}  {:.10f}\n".format(i+1, label, val))

        predict_softmax[torch.argmax(predict_softmax)] = 0.0
        predict_softmax[trg_label] = 1.0

        display(img_pil.resize((224,224)))
        print('{}\n{}'.format('GoogleNet Result', '-'*50))
        test(trg_model_3, content, img_tensor)
```

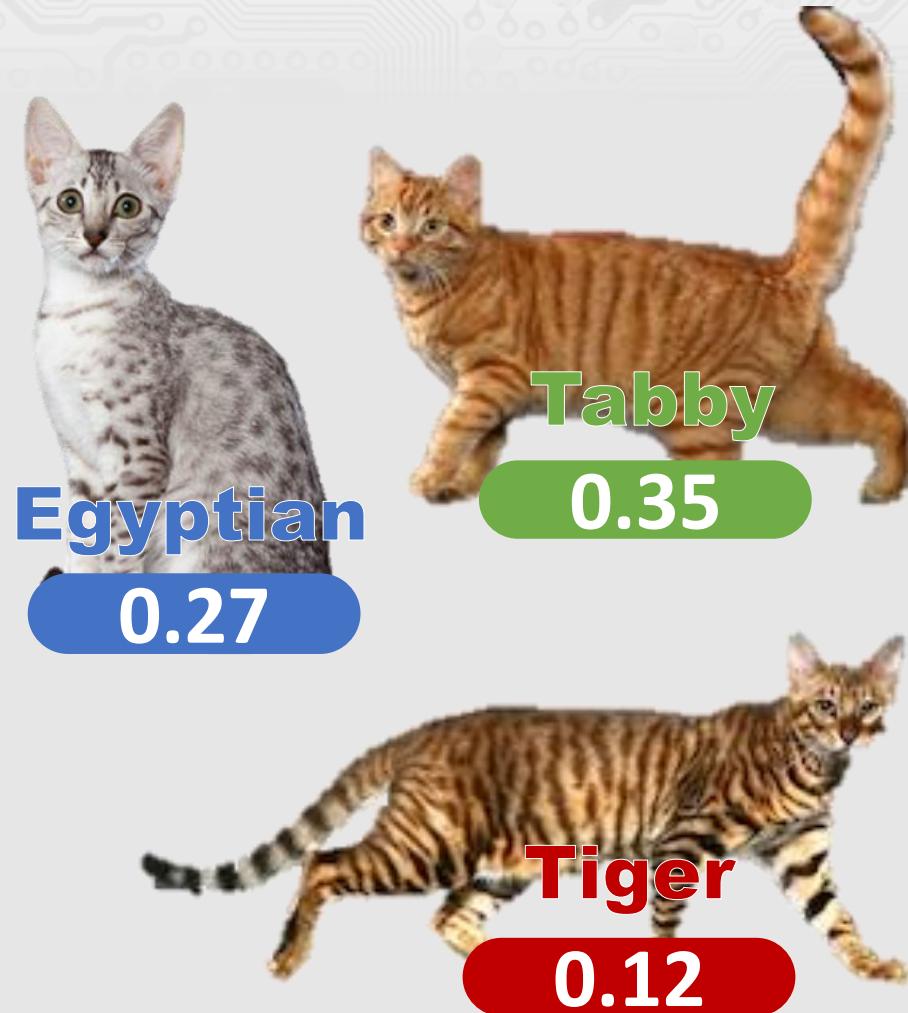
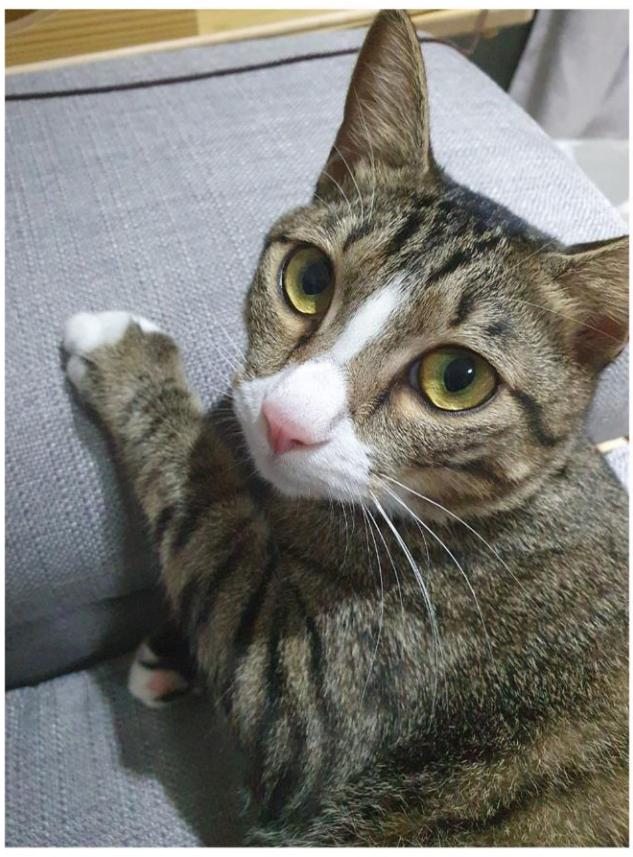


GoogleNet Result

[1]	tabby, tabby cat	0.34975
[2]	Egyptian cat	0.27327
[3]	tiger cat	0.12016
[4]	carton	0.00653
[5]	window screen	0.00460



Tabby, Egyptian , Tiger





Review

- 模型匯入 import ...
- 數據格式要正確 transform ...
- 進行預測 predict ...
- 取得標籤對應到結果最大值 blablabla ...

Objected Detection

04_ObjDetection.ipynb



Classification (類別、信心指數)



卡比獸



皮卡丘



妙花種子



Other



Object Detection (座標、類別、信心)

Predict → (Boxes, Labels, Scores)





04_ObjDetection.ipynb - WorkFlow

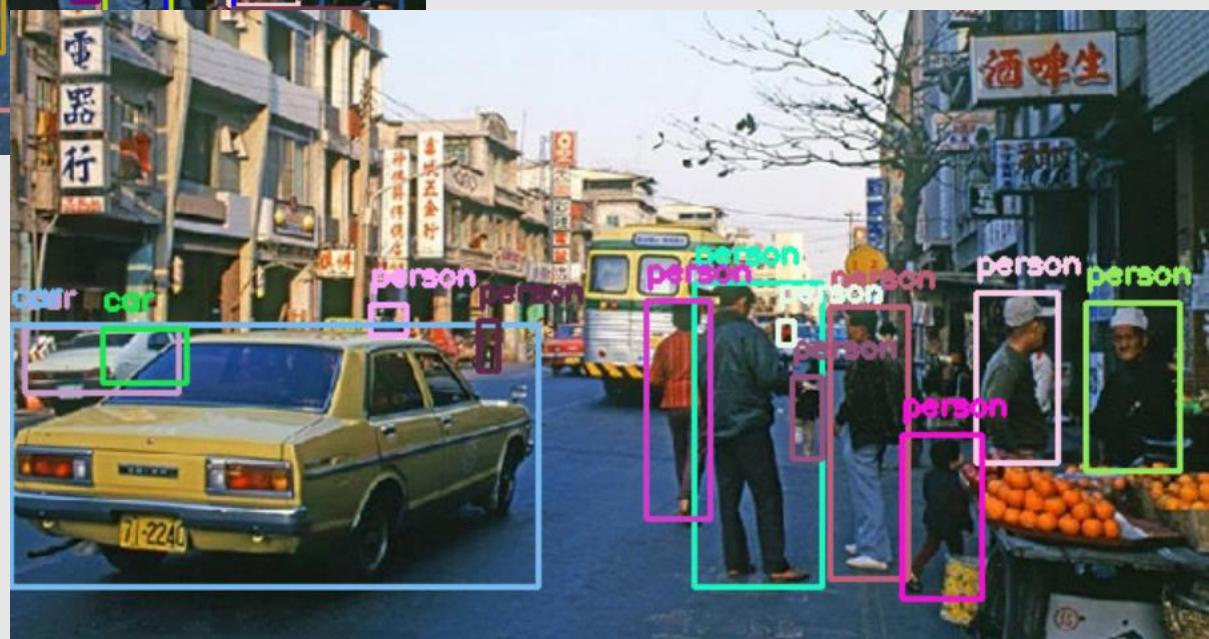
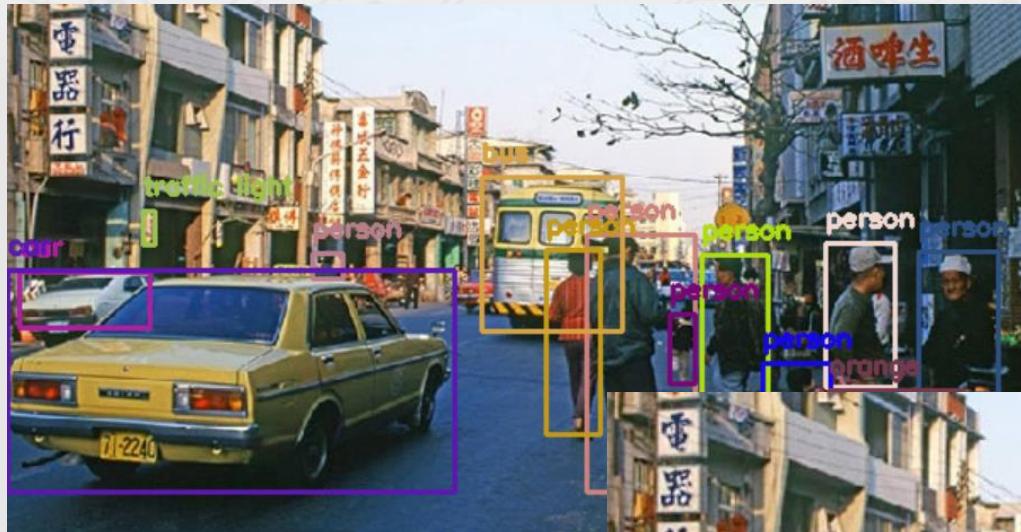
1. 移動 Colab 所在位置
2. 輸入圖片以及前處理
3. 副函式： 取得標籤、取得隨機顏色
4. 主要運作的部分
 1. 導入模型進行預測
 2. 繪製 Bounding Box & Label
 3. 計算物件的種類跟數量

練習、如果只要辨識人跟車呢？

05_Car_Person.ipynb



只要一行 Code 就能滿足你的需求



Object Detection 最經典的 YOLO

YOLOv5

<https://github.com/ultralytics/yolov5>



檔案位置 : Transfer 資料夾

我的雲端硬碟 > ... > 1091120_NQU_AI > transfer

檔案

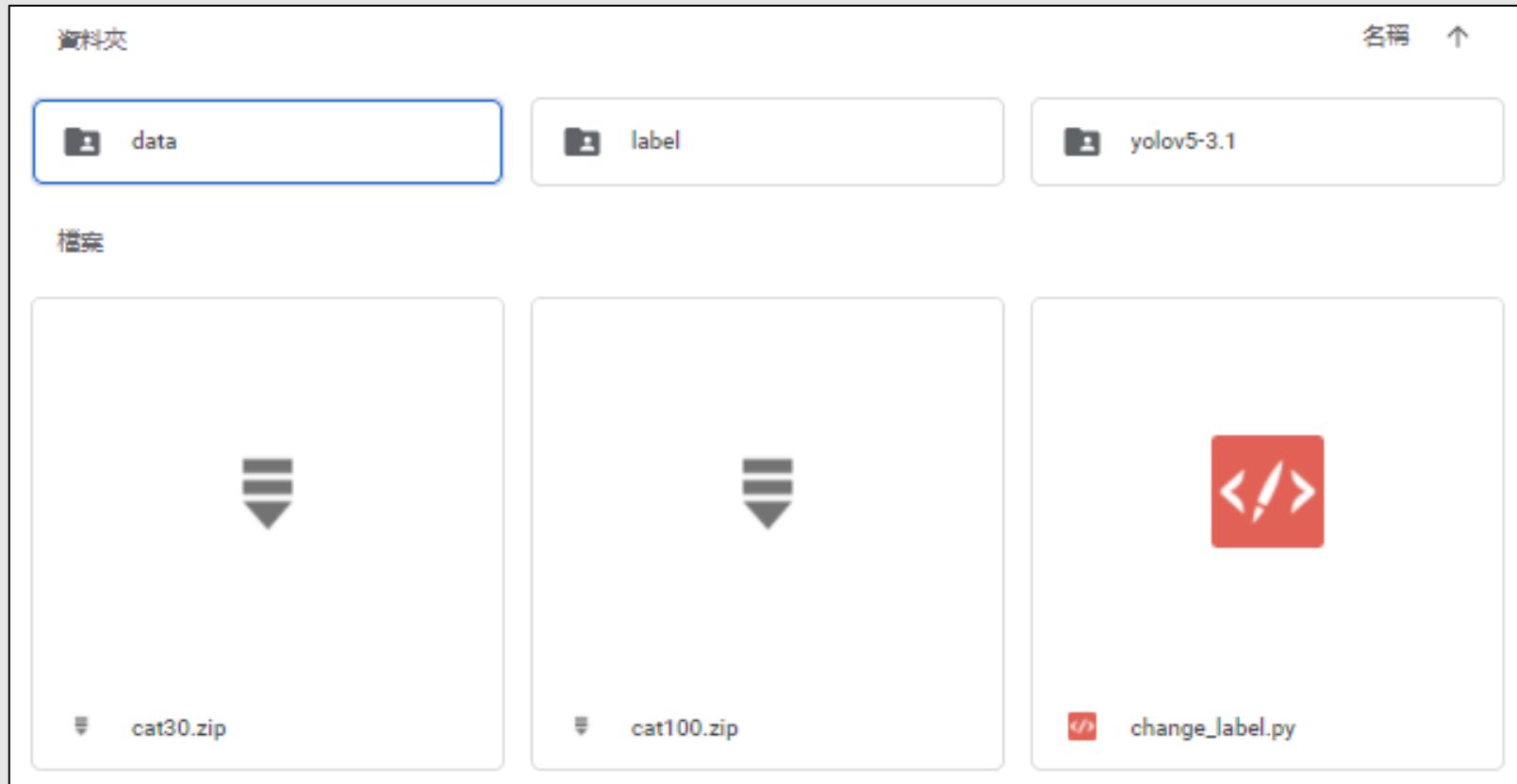
名稱 ↑

cat30.zip	cat100.zip	change_label.py
make_datasets.py	take_pic.py	yolov5-3.1.zip



06_YOLOv5 進行解壓縮

```
!unzip yolov5-3.1.zip  
!unzip cat100.zip
```

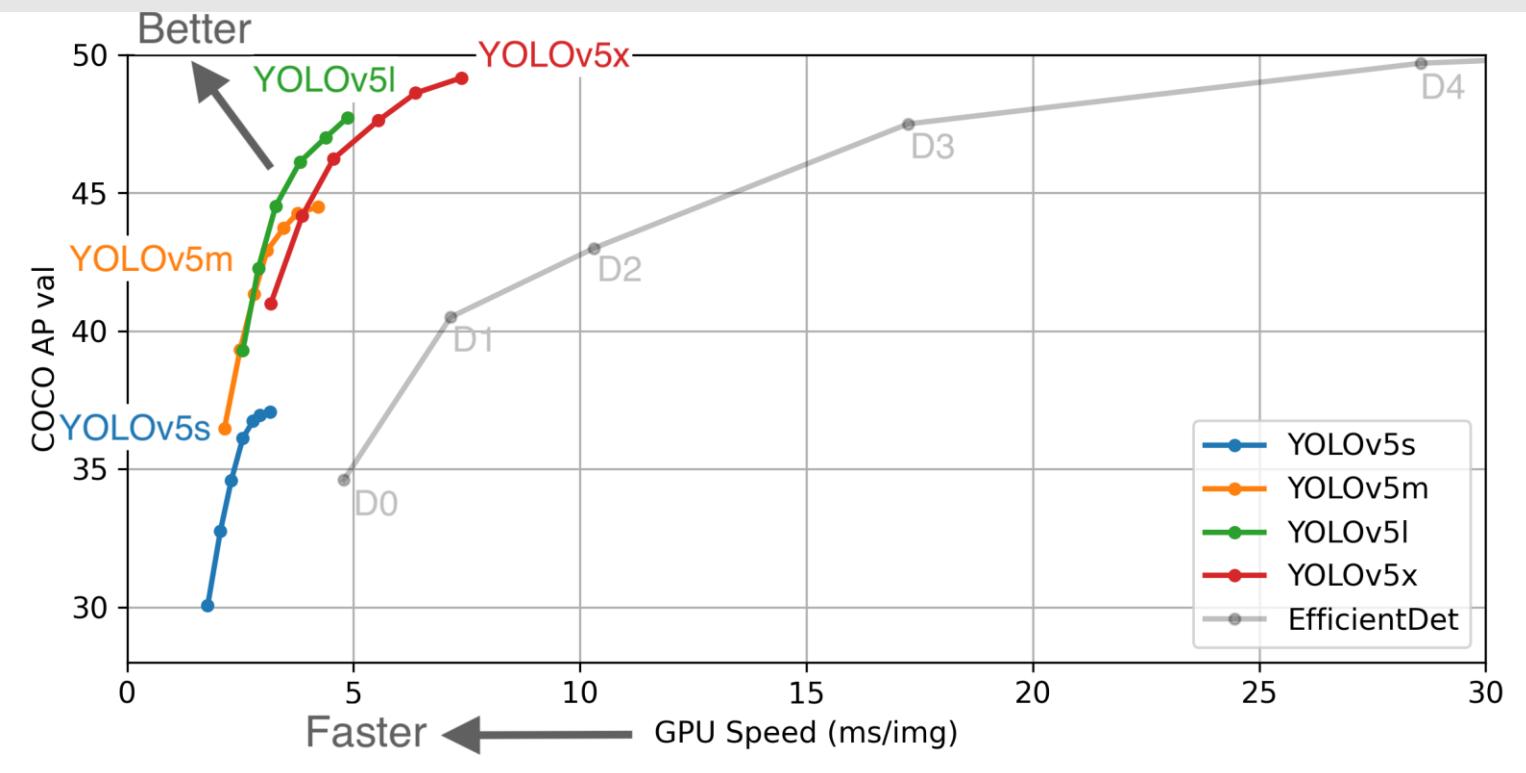




下載 yolov5 pre-trained model

下載權重檔 <https://github.com/ultralytics/yolov5/releases>

Yolov5s.pt 權重放在 yolov5 資料夾中





06_YOLOv5 移動權重檔

```
#%mv yolov5s.pt yolov5-3.1
import os
import shutil

weights = ['yolov5s.pt', 'yolov5m.pt', 'yolov5l.pt']
trg_dir = 'yolov5-3.1/'
trg_weights = [os.path.join(trg_dir, i) for i in weights]

for i in range(len(trg_weights)):
    if os.path.isfile(trg_weights[i]):
        print(f"{trg_weights[i]} existed !")
    else:
        shutil.copy(weights[i], trg_weights[i])
        print(f"{weights[i]} copy to ... {trg_weights[i]}")
```

yolov5s.pt copy to ... yolov5-3.1/yolov5s.pt
yolov5m.pt copy to ... yolov5-3.1/yolov5m.pt
yolov5l.pt copy to ... yolov5-3.1/yolov5l.pt



How to use ? Inference

- Load Image / Video , Open Camera...
- \$ python detect.py --source _____ --weights _____

Inference

detect.py runs inference on a variety of sources, downloading models automatically from the [latest YOLOv5 release](#) and saving results to `inference/output` .

```
$ python detect.py --source 0  # webcam
                      file.jpg  # image
                      file.mp4  # video
                      path/  # directory
                      path/*.*jpg  # glob
                      rtsp://170.93.143.139/rtplive/470011e600ef003a004ee33696235daa  # rtsp stream
                      rtmp://192.168.1.105/live/test  # rtmp stream
                      http://112.50.243.8/PLTV/88888888/224/3221225900/1.m3u8  # http stream
```

To run inference on example images in `inference/images` :



How to use ? File

```
python detect.py --source inference/images --weights yolov5s.pt
```

```
%cd yolov5-3.1/  
!python detect.py --source inference/images --  
weights yolov5s.pt
```

```
/content/drive/My Drive/人生KPI/工作坊/1091120_NQU_AI/transfer/yolov5-3.1  
Namespace(agnostic_nms=False, augment=False, classes=None, conf_thres=0.25, device='', img_size=640, iou_thres=0.45,  
Using CPU  
  
Fusing layers...  
Model Summary: 140 layers, 7.45958e+06 parameters, 0 gradients  
image 1/2 /content/drive/My Drive/人生KPI/工作坊/1091120_NQU_AI/transfer/yolov5-3.1/inference/images/bus.jpg: 640x48  
image 2/2 /content/drive/My Drive/人生KPI/工作坊/1091120_NQU_AI/transfer/yolov5-3.1/inference/images/zidane.jpg: 384  
Results saved to inference/output  
Done. (0.931s)
```



顯示結果

```
src_dir, out_dir = 'inference/images/', 'inference/output/'

for i in os.listdir(out_dir):
    res = cv2.imread( os.path.join( out_dir, i) )
    src = cv2.imread( os.path.join( src_dir, i) )
    cv2_imshow( cv2.hconcat( [src, res] ) )
```





How to use ? camera

```
python detect.py --source 0 --weights yolov5s.pt
```

The image shows a Linux desktop interface with a terminal window open in the foreground. The terminal displays a series of detections for a video frame, likely from a YOLOv3 model. Each detection is a tuple: width x height, number of persons, number of ties, number of cell phones, and a timestamp. The detections are as follows:

- 0: 480x640 2 persons, 1 ties, 1 cell phones, Done. (0.192s)
- 0: 480x640 2 persons, 1 ties, 1 cell phones, Done. (0.191s)
- 0: 480x640 2 persons, 1 cell phones, Done. (0.191s)
- 0: 480x640 2 persons, 1 cell phones, Done. (0.191s)
- 0: 480x640 2 persons, 1 cell phones, Done. (0.191s)
- 0: 480x640 2 persons, 1 cell phones, Done. (0.191s)
- 0: 480x640 2 persons, 1 cell phones, Done. (0.191s)
- 0: 480x640 2 persons, 1 ties, 1 cell phones, Done. (0.191s)
- 0: 480x640 2 persons, 1 cell phones, Done. (0.191s)
- 0: 480x640 2 persons, Done. (0.191s)
- 0: 480x640 2 persons, Done. (0.197s)
- 0: 480x640 2 persons, Done. (0.191s)
- 0: 480x640 2 persons, Done. (0.191s)
- 0: 480x640 2 persons, 1 ties, Done. (0.191s)
- 0: 480x640 2 persons, Done. (0.191s)
- 0: 480x640 2 persons, Done. (0.196s)
- 0: 480x640 1 persons, 1 cell phones, Done. (0.191s)
- 0: 480x640 2 persons, 1 cell phones, Done. (0.194s)
- 0: 480x640 2 persons, 1 cell phones, Done. (0.188s)
- 0: 480x640 2 persons, 1 cell phones, Done. (0.192s)

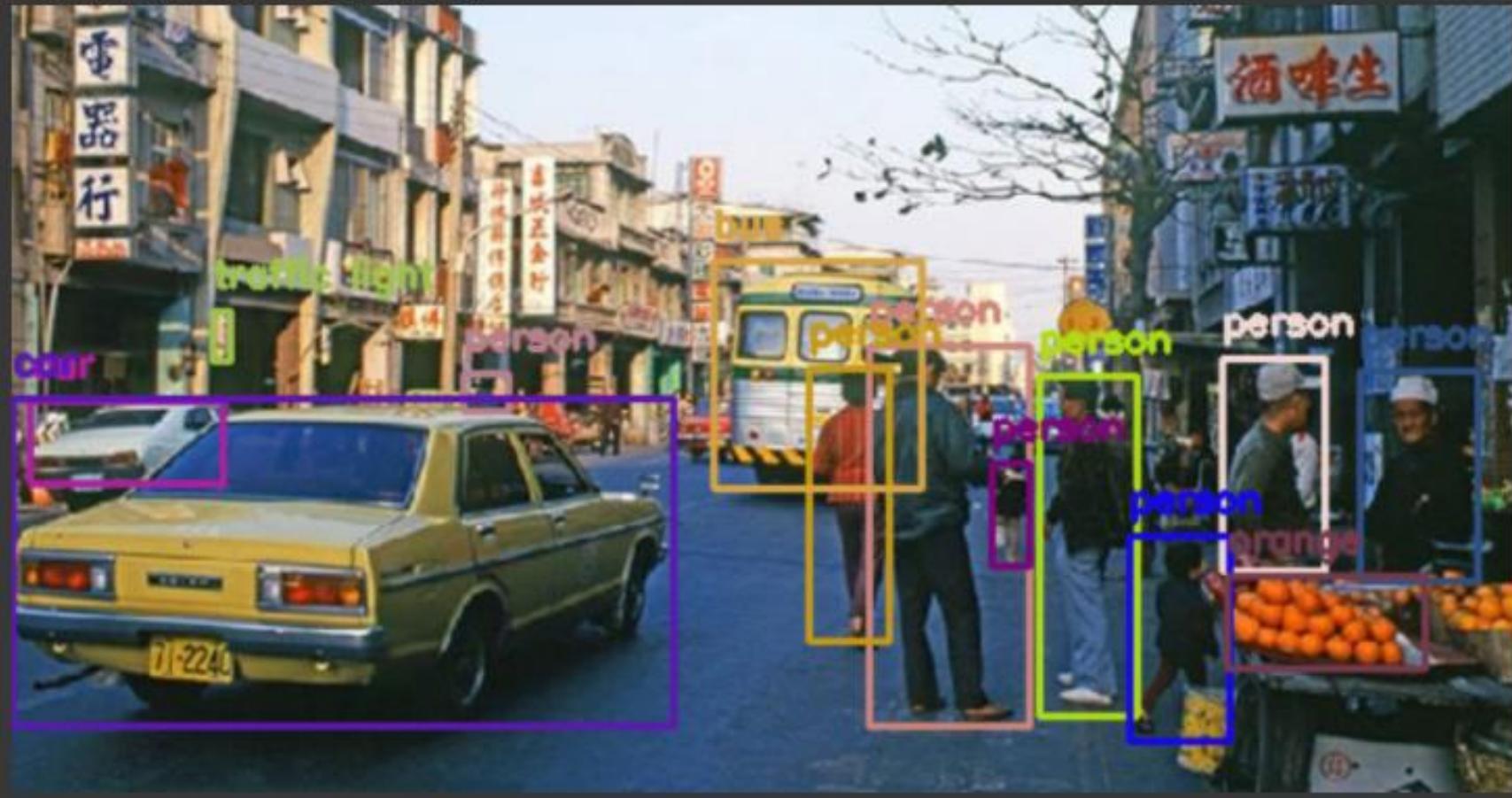
In the background, a camera feed is visible, showing a person wearing glasses and a black shirt, and another person holding a white cell phone. Bounding boxes and confidence scores are overlaid on the image: "person 0.58" and "person 0.40" for the two people, and "cell phone 0.49" for the phone.



fasterrcnn

person : 8, orange : 1, car : 2, traffic light : 1, bus : 1

Cost time: 8.154329776763916s (CPU)





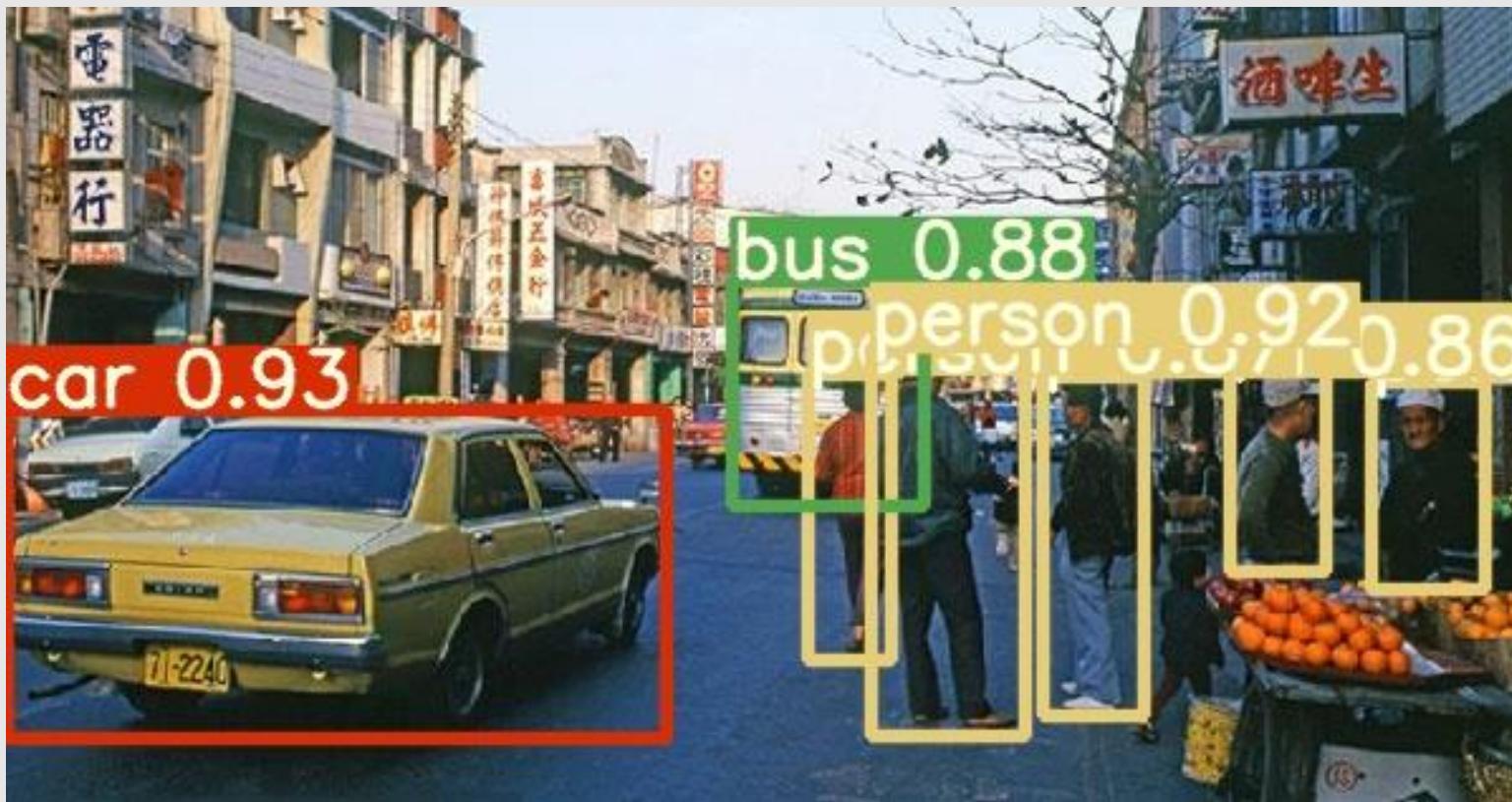
Yolov5l.pt

Using CPU

Fusing layers...

Model Summary: 236 layers, 4.77901e+07 parameters, 0 gradients

image 1/1 /content/drive/MyDrive/人生KPI/工作坊/1091120_NQU_AI/images/obj_image.jpg: 352x640 5 persons, 1 cars, 1 buss, Done. (1.066s)



討論一下

同樣都是CPU的狀況下
Fasterrcnn 框出更多但耗時更久
YOLOv5 (yolov5l.pt) 少了不少但耗時僅1秒



Another way - Hub

PyTorch Hub

To run batched inference with YOLOv5 and PyTorch Hub:

```
import torch
from PIL import Image

# Model
model = torch.hub.load('ultralytics/yolov5', 'yolov5s', pretrained=True).
model = model.autoshape() # for autoshaping of PIL/cv2/np inputs and NMS

# Images
img1 = Image.open('zidane.jpg')
img2 = Image.open('bus.jpg')
imgs = [img1, img2] # batched list of images

# Inference
prediction = model(imgs, size=640) # includes NMS
```

PyTorch
Hub



PyTorch Hub - How to use

```
# Model
import torch
# yolov5s.pt
model = torch.hub.load('ultralytics/yolov5', 'yolov5s',
pretrained=True).fuse().eval()
# for autoshaping of PIL/cv2/np inputs and NMS
model = model.autoshape()
# info
print(model)
print(model.names)
```



PyTorch Hub - model · model.names

```
Model Summary: 140 layers, 7.45958e+06 parameters, 7.45958e+06 gradients
Fusing layers...
Adding autoShape...
autoShape(
    (model): Model(
        (model): Sequential(
            (0): Focus(
                (conv): Conv(
                    (conv): Conv2d(12, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
                    (act): Hardswish()
                )
            )
            (1): Conv(
                (conv): Conv2d(32, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
                (act): Hardswish()
            )
            (2): BottleneckCSP(
                (cv1): Conv(
                    (conv): Conv2d(64, 32, kernel_size=(1, 1), stride=(1, 1))
                    (act): Hardswish()
                )
                (cv2): Conv2d(64, 32, kernel_size=(1, 1), stride=(1, 1), bias=False)
            )
        )
    )
)
```

```
['person', 'bicycle', 'car', 'motorcycle', 'airplane', 'bus', 'train', 'truck', 'boat', 'traffic light', 'fire hydrant',
'stop sign', 'parking meter', 'bench', 'bird', 'cat', 'dog', 'horse', 'sheep', 'cow', 'elephant', 'bear', 'zebra', 'giraffe',
'backpack', 'umbrella', 'handbag', 'tie', 'suitcase', 'frisbee', 'skis', 'snowboard', 'sports ball', 'kite', 'baseball bat',
'baseball glove', 'skateboard', 'surfboard', 'tennis racket', 'bottle', 'wine glass', 'cup', 'fork', 'knife', 'spoon',
'bowl', 'banana', 'apple', 'sandwich', 'orange', 'broccoli', 'carrot', 'hot dog', 'pizza', 'donut', 'cake', 'chair',
'couch', 'potted plant', 'bed', 'dining table', 'toilet', 'tv', 'laptop', 'mouse', 'remote', 'keyboard', 'cell phone',
'microwave', 'oven', 'toaster', 'sink', 'refrigerator', 'book', 'clock', 'vase', 'scissors', 'teddy bear', 'hair drier', 'toothbrush']
cat
```



PyTorch Hub - Load Image & Inference

```
# Images
img = Image.open('image.jpg')

# Inference
with torch.no_grad():
    prediction = model(img, size=640) # includes NMS

for i in prediction[0]:
    idx = (int(i[-1]))
    print(model.names[idx])
```



Transfer Learning

- 現在你為了節省力氣用了別人訓練好的模型，但是你發現你要辨識的東西不在他當初訓練的數據集當中，也就是這個AI模型看不懂你要辨識的東西。

我的專題要做馬類辨識，你很厲害我想繼續使用你可以嗎？



摠...我只會分辨貓跟狗



CAVEDU

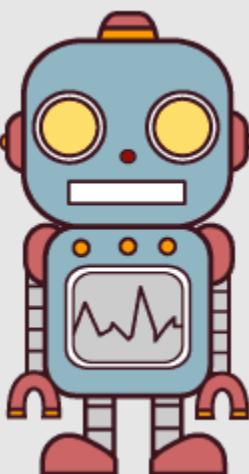
站在巨人的肩膀上

Transfer Learning



Transfer learning – pre-trained model

Hi, I'm an AI robot
I can recognize the data
I had seen before, like
cat and dog



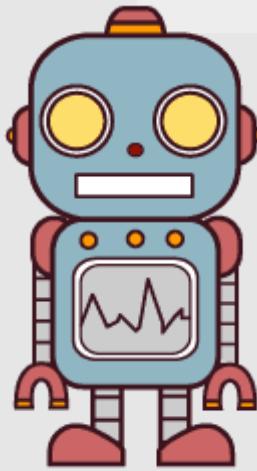


Transfer learning – pre-trained model

a cat



a dog





Transfer learning – custom data

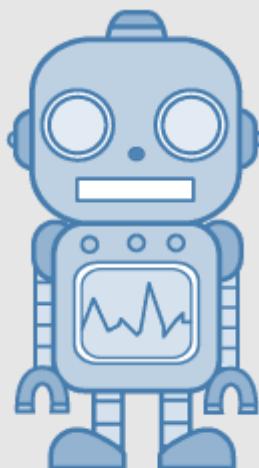
uhhh...dog



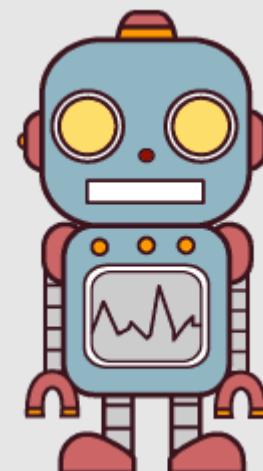


How to Implement AI on your data ?

Hey ! How do
you recognize
cat and dog ?



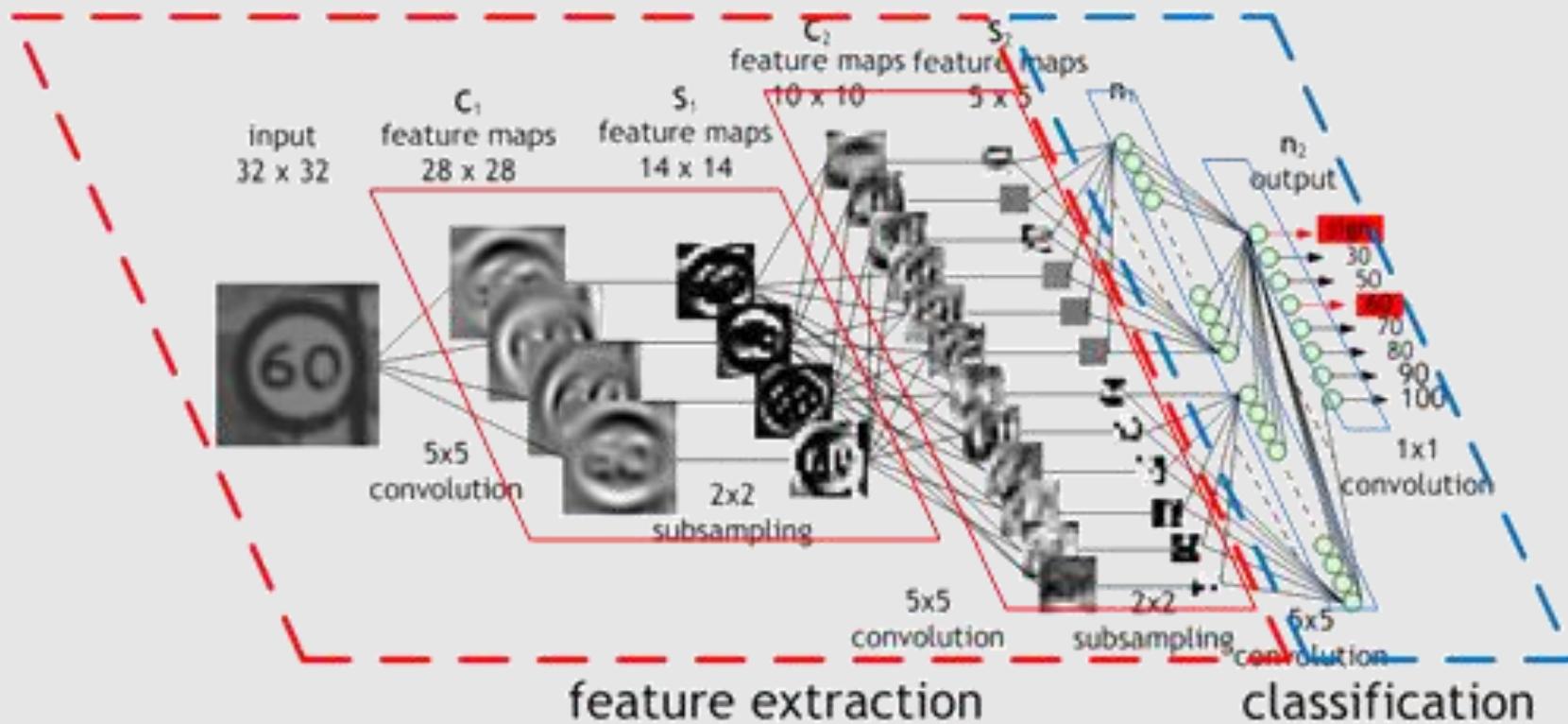
You can use the
same way to
recognize horse



I got it

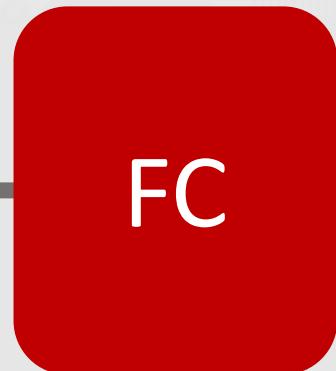


起手式 : CNN





Transfer Learning -1



Dog

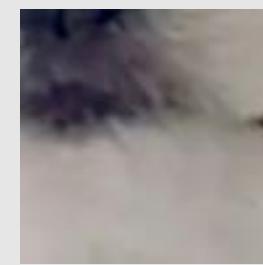
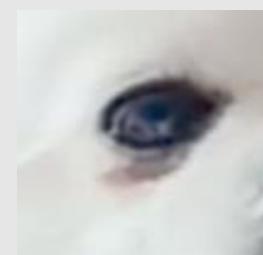
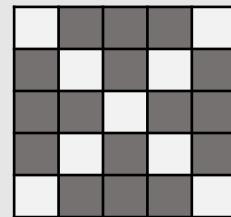
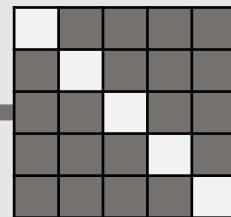
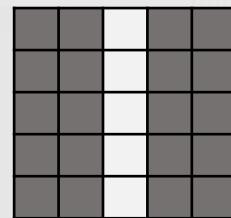
Only train a part of network, freeze another layer



Horse

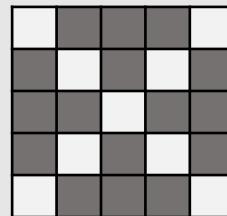
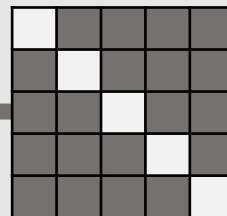
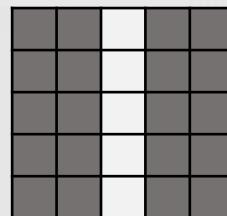


How???? Weights like filters



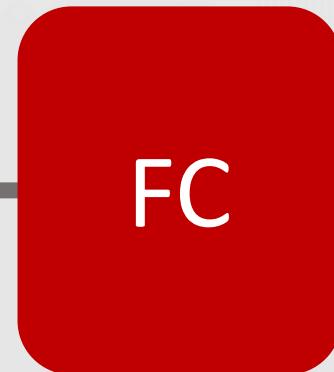


Use same weights to extract feature



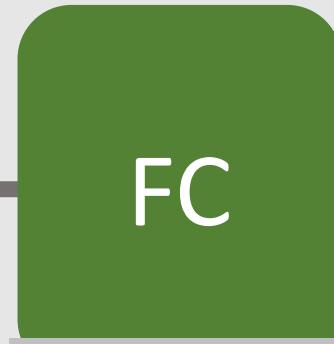


This way we called “Feature Extraction”



Dog

Only train a part of network, freeze another layer



Horse

特徵擷取

組合特徵



PyTorch 程式範例

Load a pretrained model and reset final fully connected layer.

```
model_ft = models.resnet18(pretrained=True)
num_ftrs = model_ft.fc.in_features
# Here the size of each output sample is set to 2.
# Alternatively, it can be generalized to nn.Linear(num_ftrs, len(class_names)).
model_ft.fc = nn.Linear(num_ftrs, 2)

model_ft = model_ft.to(device)

criterion = nn.CrossEntropyLoss()

# Observe that all parameters are being optimized
optimizer_ft = optim.SGD(model_ft.parameters(), lr=0.001, momentum=0.9)

# Decay LR by a factor of 0.1 every 7 epochs
exp_lr_scheduler = lr_scheduler.StepLR(optimizer_ft, step_size=7, gamma=0.1)
```



Transfer Learning -2



CNN

FC

Dog

Inherited weights from pre-trained model



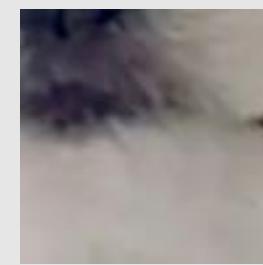
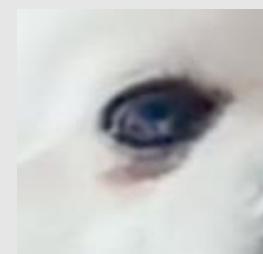
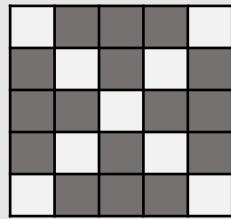
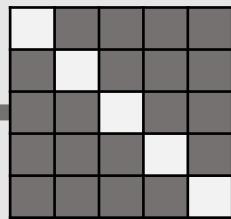
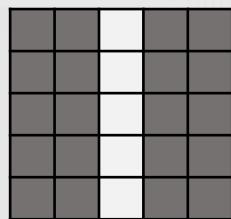
CNN

FC

Horse

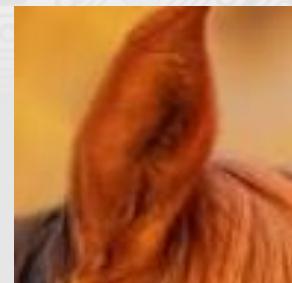
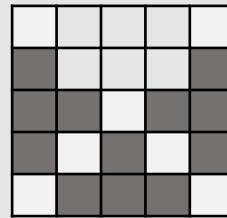
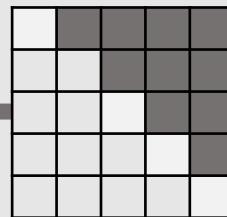
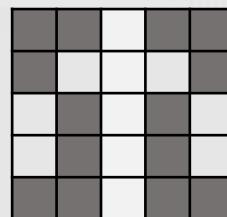


How ???





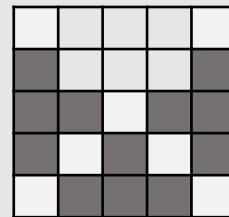
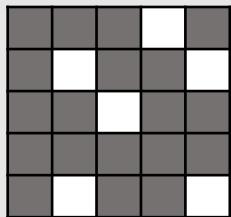
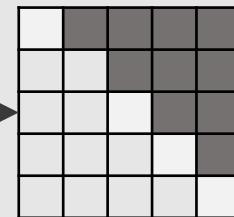
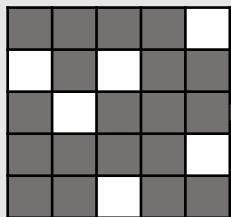
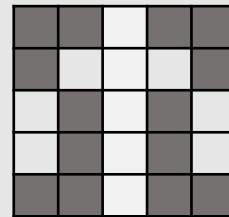
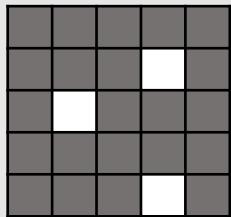
Improve filter , but no need more time



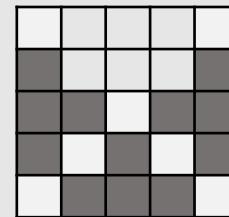
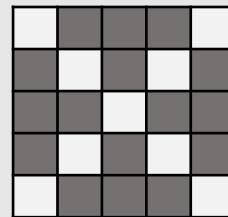
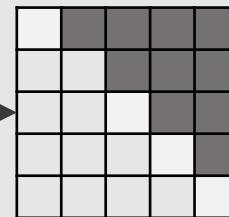
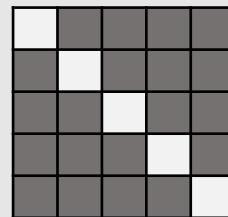
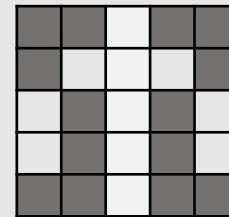
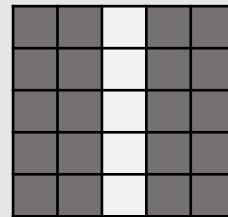


Which way is faster ?

From Noise ?



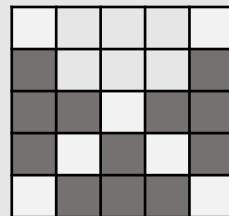
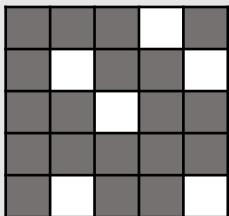
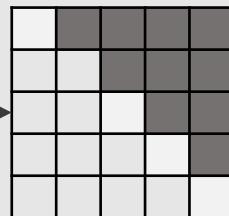
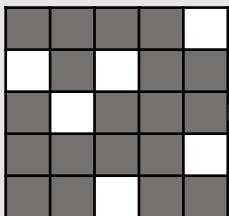
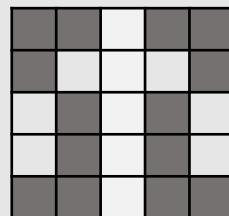
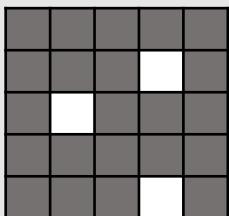
From Pre-Trained ?



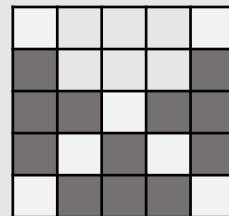
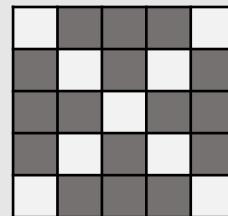
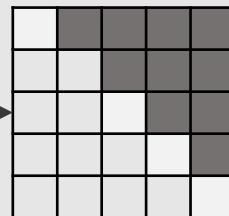
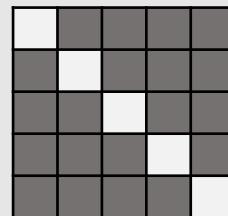
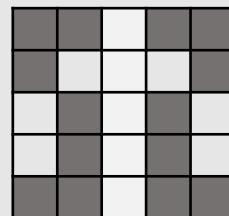
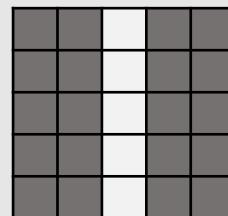


Which way is faster ?

Trained from scratch

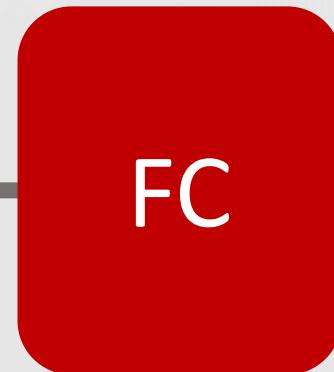
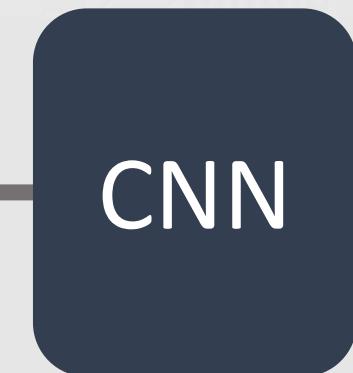


Transfer Learning





This way call “Fine-Tuning”



Dog

Inherited weights from pre-trained model



Horse

學習抓取特徵 組合特徵



Timing of using the method

數據集數量	數據集相似	使用方法	備註
小	相似	特徵擷取	由於內容相似，特徵擷取的結果也會符合需求，所以直接訓練線性分類的部分即可。
小	不相似	微調	這邊就比較特殊了，因為數據集不相似所以CNN的特徵擷取不一定適用，所以建議會從CNN的某一段就開始進行訓練。
大	相似	微調	數據量大又相似，建議還是透過對整個網路微調會較佳
大	不相似	微調	通常遇到不相似又數據量大的，會直接建議從頭訓練 (learn from scratch)，當然也是可以嘗試進行全網路的微調，在大部分的狀況下微調是能幫助神經網路快速收斂的



Yolov5 use Transfer Learning?

- 準備數據 100張貓咪
- 標註圖片 (標 label) [labelImg](#)
- 建立神經網路 (MLP、CNN、RNN)
- 處理數據 (正規化、資料擴增、轉換成張量 ...)
- 訓練 (疊代次數、超參數) [YOLO via Colab](#)
- 測試 (其他資料) [YOLO on Nano](#)

Try to make your own dataset



我準備了100張照片貓咪的照片

- 來源：<https://www.kaggle.com/crawford/cat-dataset>

The screenshot displays the Kaggle dataset page for 'Meow Meow! Cat Dataset'. At the top left is the dataset icon and title 'Meow Meow! Cat Dataset'. Below the title is a subtitle 'Over 9,000 images of cats with annotated facial features'. A thumbnail image shows a cat from below. To the right is a close-up of a cat's face with numbered points (1 through 5) indicating detected features. Below the image are navigation links: 'Data' (underlined), 'Tasks', 'Notebooks (10)', 'Discussion (4)', 'Activity', and 'Metadata'. To the right are buttons for 'Download (2 GB)' and 'New Notebook'. A sidebar on the left says 'Your Dataset download has started. Show your appreciation with an upvote' and shows a count of 399 upvotes with a button labeled '399'. Below the sidebar are icons for 'Usability 7.5', 'License CC0: Public Domain', and 'Tags image data, animals, computer vision'.



有沒有感覺傻眼貓咪？



傻眼貓咪

CAVEDU

好啦，你們只要標30張就好



將 cat30 下載到你的電腦中

開啟(O)
在新視窗開啟(E)
使用 Skype 分享
以 Code 開啟
解壓縮全部(T)...
7-Zip
CRC SHA
釘選到開始畫面(P)

cat30.zip

機 機 Acer (C:) > 使用者 > p5138 > 下載 > cat30

00000906_004.j pg	00000906_005.j pg	00000906_019.j pg	00000906_021.j pg	00000906_025.j pg	00000907_011.j pg	00000907_013.j pg

CAVEDU

來嘗試標 label 吧！

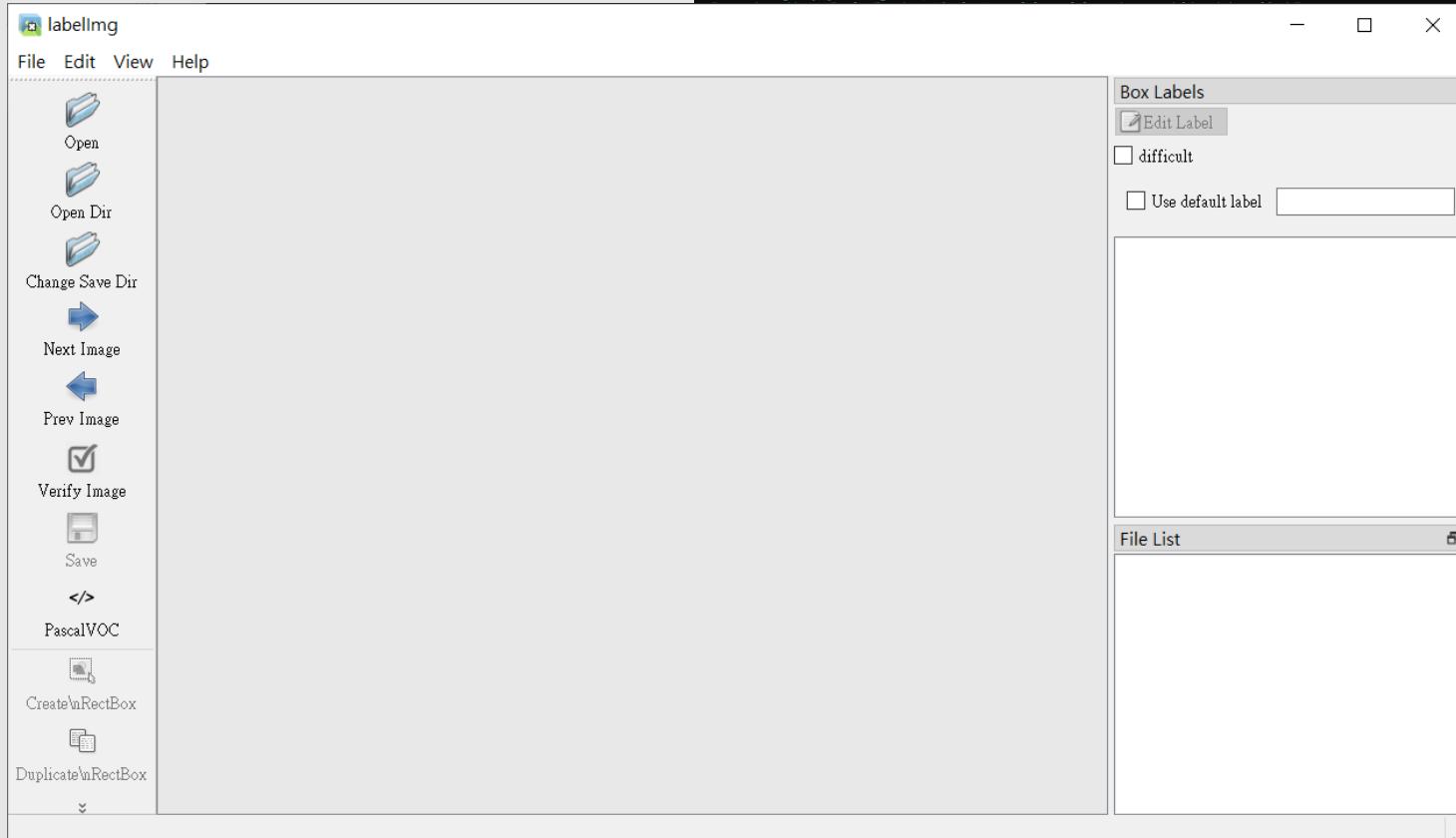
LabelImg



安裝 Label Img (Windows + PyPI)

\$ Pip3 install labelImg
\$ labelImg

```
(tf) D:\WorkSpace\1091013_淡江大學_JetsonNano手把手新手教學\Day_3>pip3 install labelImg
Collecting labelImg
  Downloading labelImg-1.8.3-py2.py3-none-any.whl (248 kB)
|████████████████████████████████████████████████████████████████████████████████| 248 kB 652 kB/s
Collecting lxml
  Downloading lxml-4.6.1-cp36-cp36m-win_amd64.whl (3.5 MB)
|████████████████████████████████████████████████████████████████████████████████| 3.5 MB 1.7 MB/s
Collecting PyQt5
  Downloading PyQt5-5.15.1-5.15.1-cp35.cp36.cp37.cp38.cp39-none-win_amd64.whl (59.4 MB)
|████████████████████████████████████████████████████████████████████████████████| 59.4 MB 6.4 MB/s
Collecting PyQt5-sip<13,>=12.8
```



LabelImg 操作

Windows 跟 Linux 操作基本一樣



Open Dir 選擇 data 資料夾





Open Dir 選擇 data 資料夾





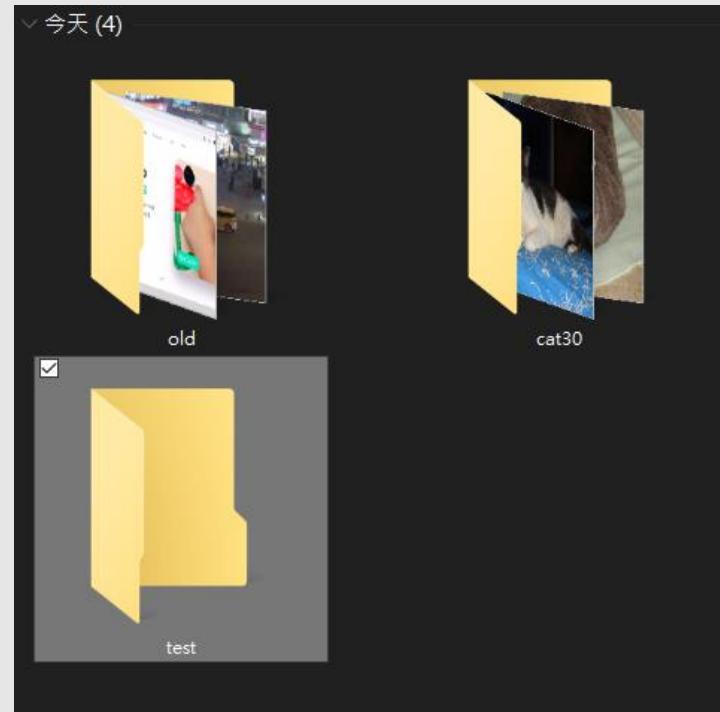
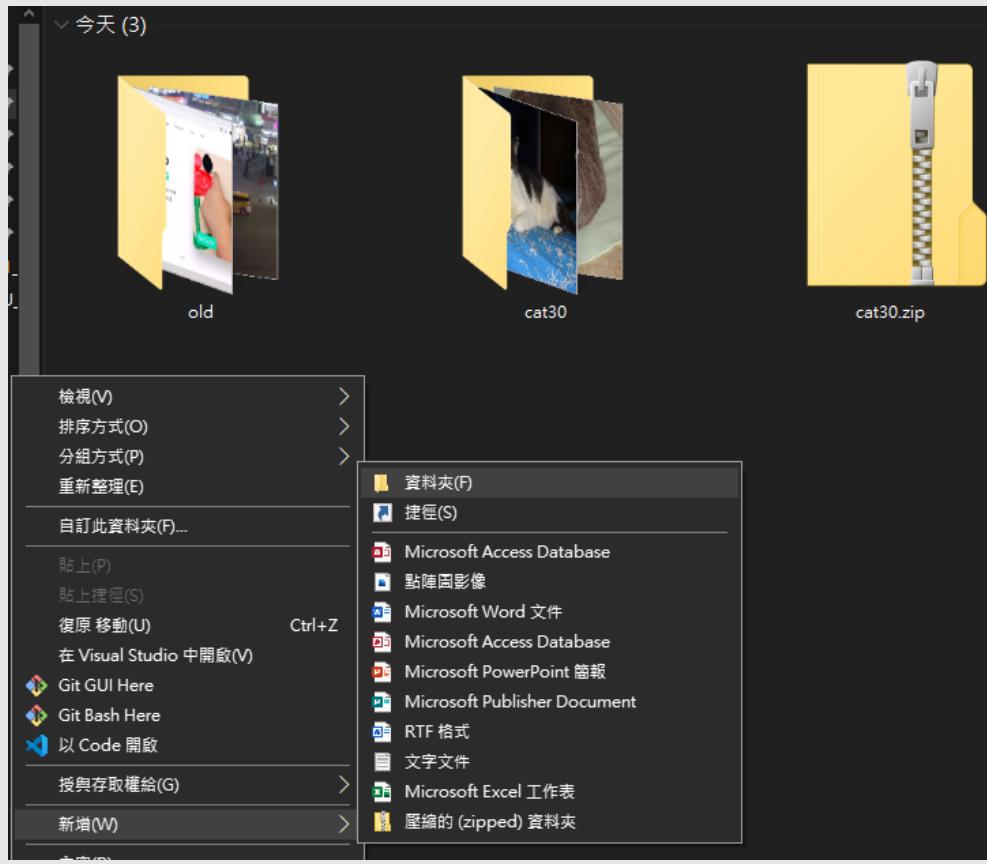
點擊 PascalVOC 改成 YOLO 格式





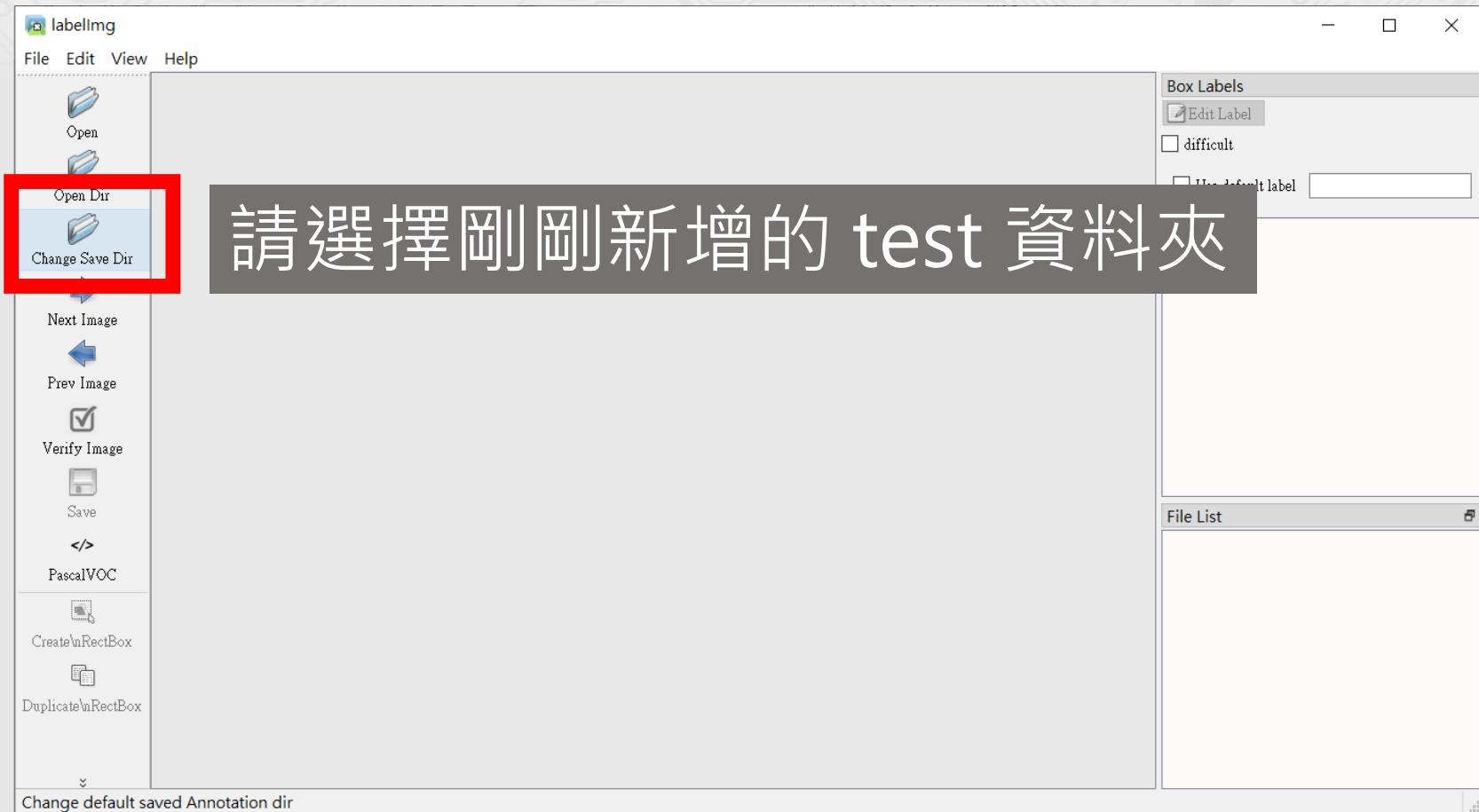
新建 test 資料夾存放標籤檔

- 先創建一個 test 的資料夾





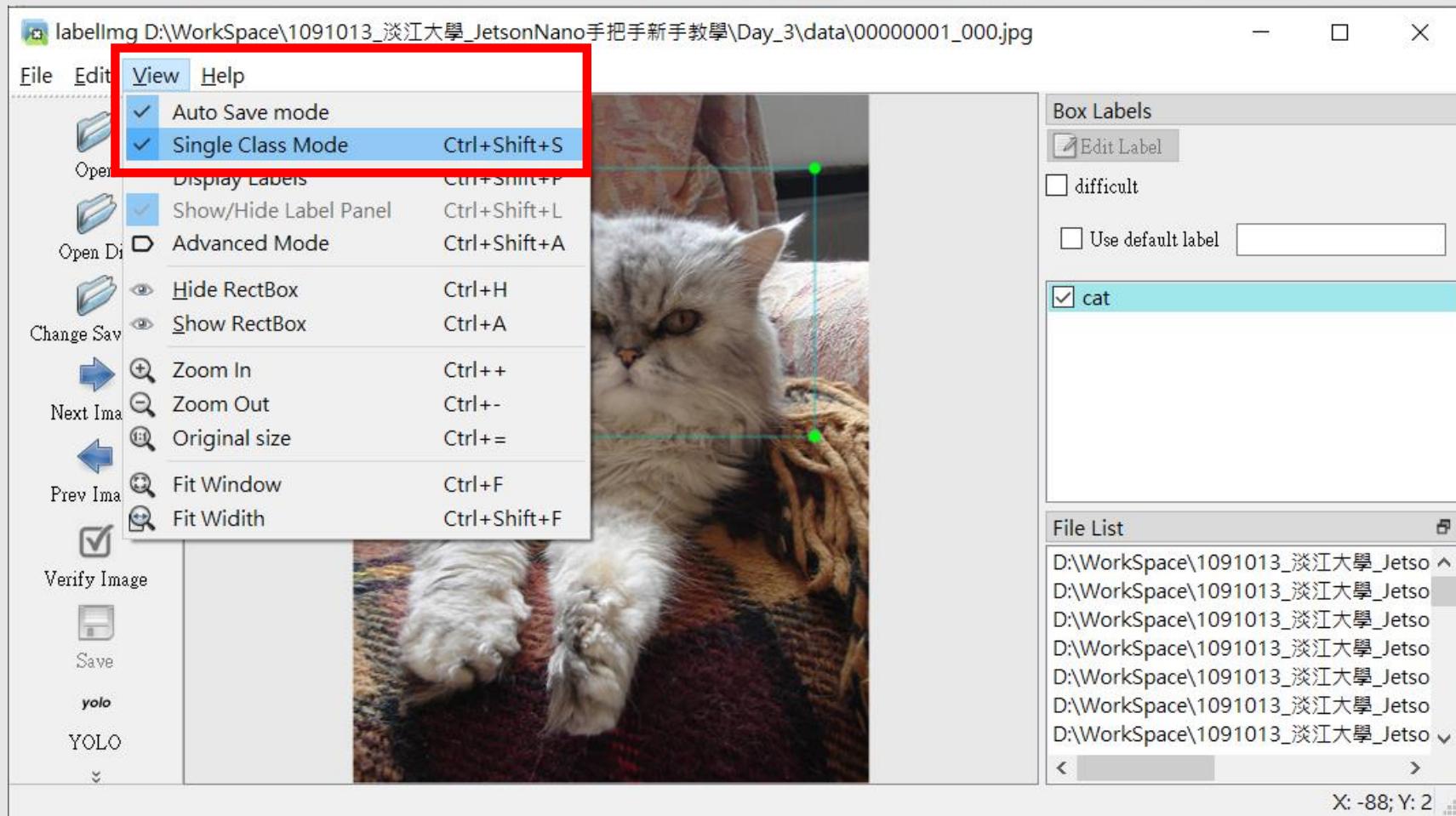
選擇儲存Label的目錄 Change Save Dir





選擇自動儲存模式

- 開啟 Auto Save Mode & Single Class Mode





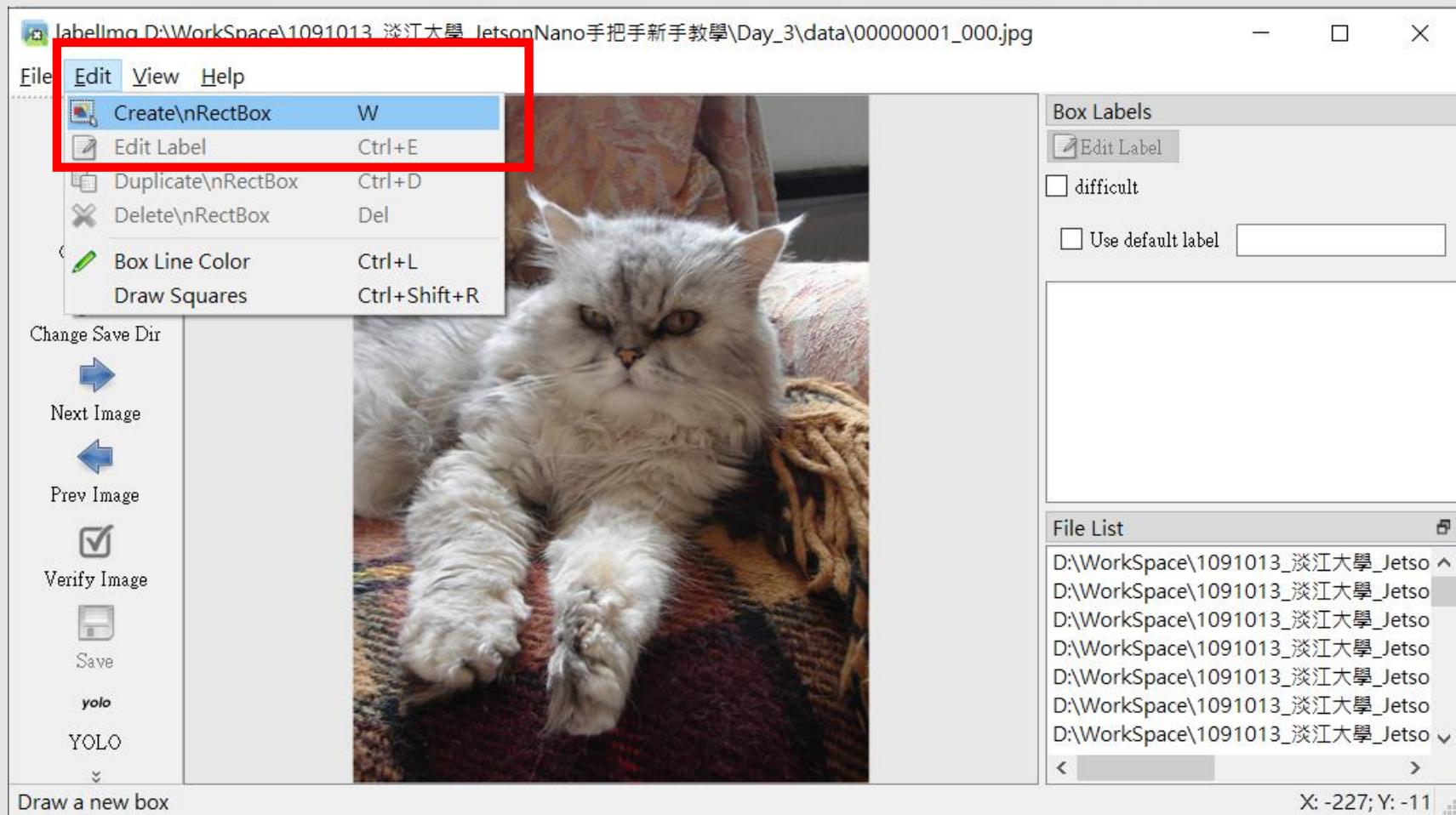
多個 Label 範例





開始標Label：快捷鍵W

- 回到 LabelImg 進行標註：快捷鍵 W





開始標Label：建議標整隻貓

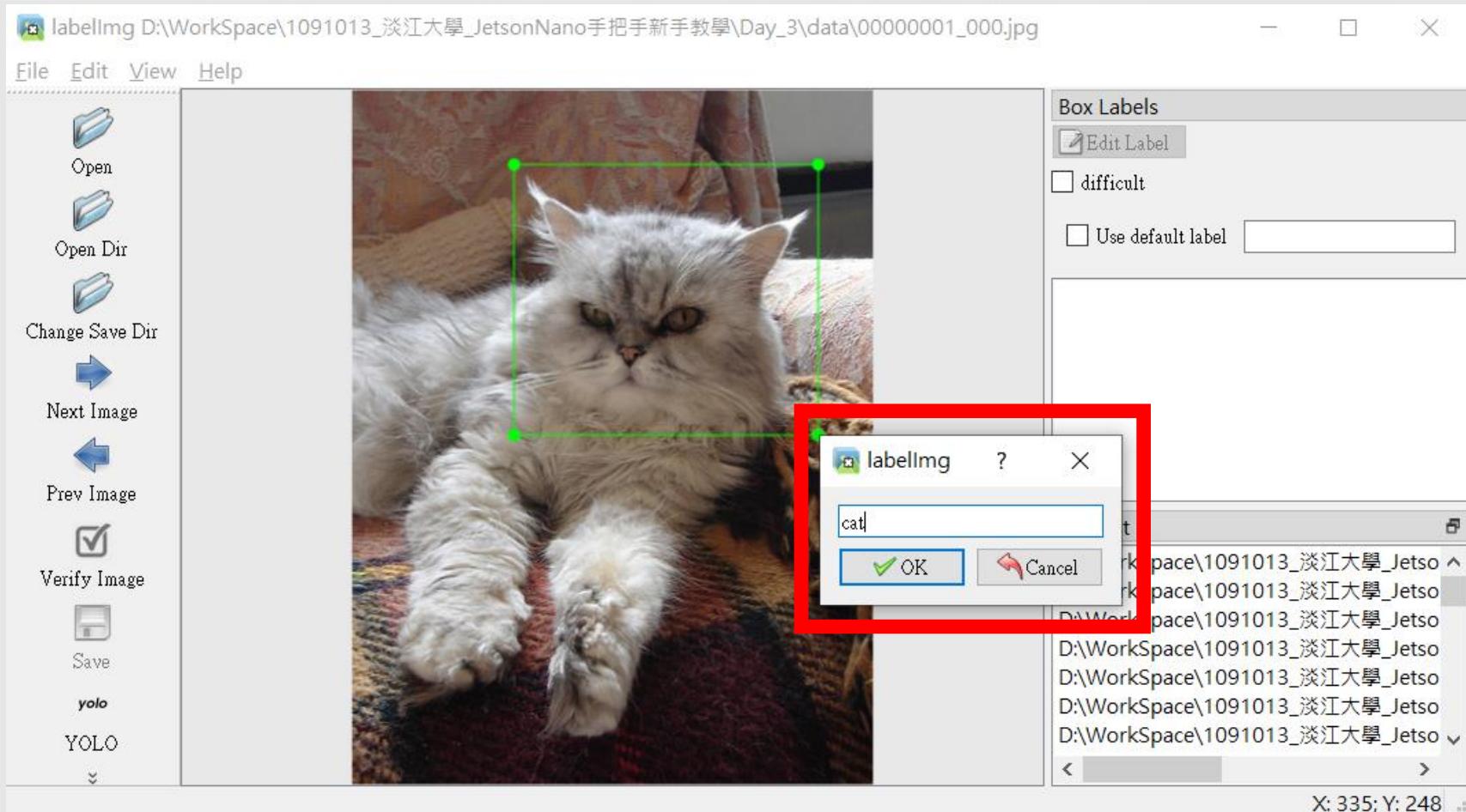
- 回到 LabelImg 進行標註





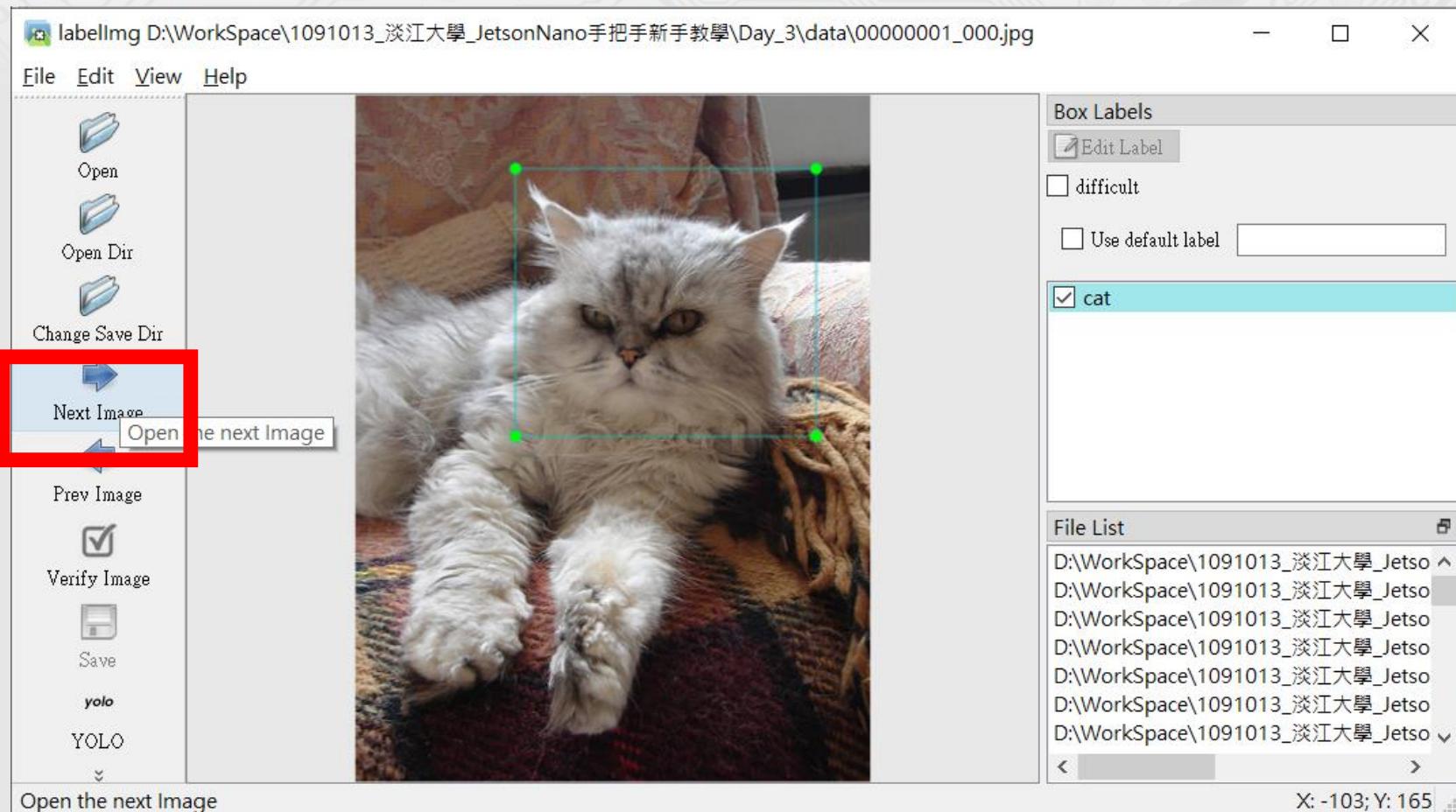
開始標Label：定義標籤名稱

- 宣告標籤





下一張：直到不能再跳下一張



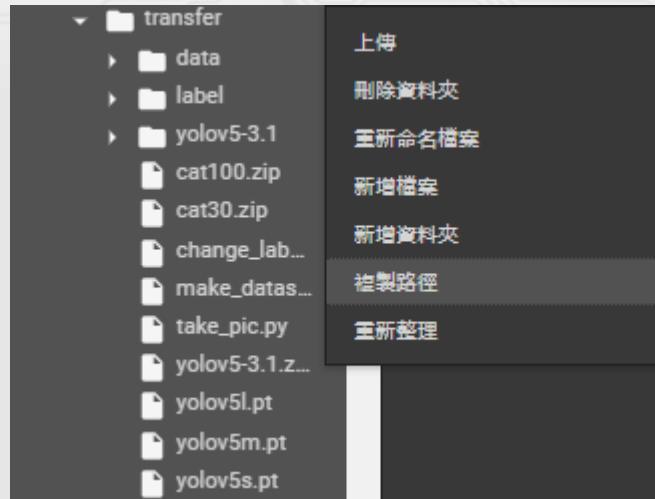


複製檔案到 data & label 資料夾

- 將 cat30 的圖片 上傳到 data
- 將 test 的標籤 上傳到 label



移動到 transfer 資料夾



移動到 transfer 目錄



```
%cd /content/drive/MyDrive/人生KPI/工作坊/1091120_NQU_AI/transfer  
!ls
```

```
/content/drive/MyDrive/人生KPI/工作坊/1091120_NQU_AI/transfer  
cat100.zip      data          take_pic.py    yolov5l.pt  
cat30.zip       label         yolov5-3.1    yolov5m.pt  
change_label.py  make_datasets.py  yolov5-3.1.zip  yolov5s.pt
```



進行資料整理

```
!python make_datasets.py
```

```
!apt-get install tree  
print('='*100)  
!tree custom -d
```

```
Create dir : custom/images/train  
Create dir : custom/images/val  
Create dir : custom/labels/train  
Create dir : custom/labels/val  
Total Data length: 100  
Validation Data: ['00000906_011', '00000001_011', '00000901_023', '00000901_027', '00000001_000']  
-----  
New Data length: 95  
Finish!
```



6 directories



確認資料結構

```
└── custom
    ├── images
    │   ├── train
    │   └── val
    └── labels
        ├── train
        └── val
├── data
└── label
└── yolov5-3.1
    ├── data
    │   └── scripts
    ├── inference
    │   ├── images
    │   └── output
    ├── models
    │   ├── hub
    │   └── __pycache__
    ├── utils
    │   ├── google_app_engine
    │   └── __pycache__
    └── weights
```

22 directories



修改YAML檔案 : yolov5-3.1/data

1. 複製 coco.yaml
2. 更名成 custom.yaml

```
import os
import shutil
coco_yaml = './yolov5-3.1/data/coco.yaml'
custom_yaml = './yolov5-
3.1/data/custom.yaml'

shutil.copy(coco_yaml , custom_yaml)

print('Finish !' if os.path.isfile(custom_
yaml) else "Failed !" )
```



雲端上編輯 Text Editor

- 點擊兩下 yolov5-3.1/data/custom.yaml
- 連結更多應用程式
- 搜尋「Text Editor」
- 選擇開啟工具 > 使用 Text editor
- 授權使用

The screenshot shows a Mac OS file viewer window. In the center, there is a file named "custom.yaml". A contextual menu is open at the top right of the file, with the option "選擇開啟工具" (Select Open With) highlighted. A sub-menu titled "無法預覽" (Cannot Preview) is displayed, listing available applications. The "Text Editor" application is listed, showing its icon, rating (★★★★★ 1456), source (texteditor.co), and user count (100,000+). Below this, a preview of the "Text Editor" application's interface is shown, which is a simple text editor with buttons for opening files from the computer or Google Drive, and creating new files.

custom.yaml

選擇開啟工具 ▾

新增

重要檔案存放區

無法預覽

下載

+ 連結

Text Editor

★★★★★ (1456) 來源網站: texteditor.co 100,000+ 位使用者

Welcome to Text Editor!

Text Editor is a free app that allows you to create, open, and edit text files on your computer and Google Drive.

To get started, open or create a text file with one of the buttons below.

Open File From Computer

Open File From Google Drive

Create New Text File



修改 Custom.yaml 檔

```
train: ../custom/images/train  
val: ../custom/images/val
```

```
# number of classes
```

```
nc: 1
```

```
# class names
```

```
names: ['cat']
```

```
1 train: ../custom/images/train  
2 val: ../custom/images/val  
3  
4 # number of classes  
5 nc: 1  
6  
7 # class names  
8 names: ['cat']
```



安裝相依套件 & 進行訓練

```
!pip install -r yolov5-3.1/requirements.txt
```

```
Collecting PyYAML>=5.3
  Downloading https://files.pythonhosted.org/packages/64/c2/b80047c7ac2478f9501676c988a5411ed5572f35d1beff9cae07d321512c/PyYAML-5.3.1.tar.gz (269kB)
|██████████| 276kB 5.9MB/s
```

```
%cd yolov5-3.1/
!python train.py --epochs 50 --batch-size 16 --
data data/custom.yaml --cfg models/yolov5s.yaml --
weights yolov5s.pt
```

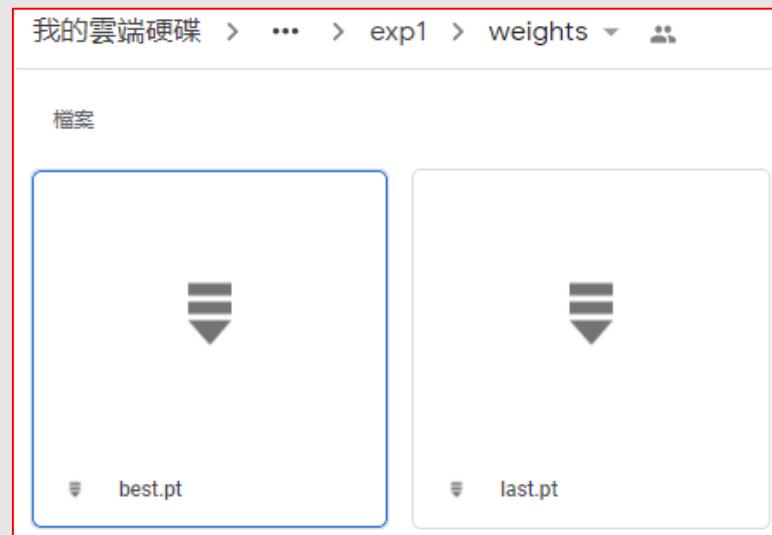
```
Transferred 362/370 items from yolov5s.pt
Optimizer groups: 62 .bias, 70 conv.weight, 59 other
Scanning images: 100% 95/95 [00:00<00:00, 314.53it/s]
Scanning labels .../custom/labels/train.cache (95 found,
Scanning images: 100% 5/5 [00:00<00:00, 164.40it/s]
Scanning labels .../custom/labels/val.cache (5 found, 0
NumExpr defaulting to 2 threads.
```



輸出權重 best.py 、 Last.pt

```
Epoch    gpu_mem      box      obj      cls      total      targets      img_size
98/99    5.16G    0.0315    0.01294        0    0.04444        41          640: 100% 6/6 [00:04<00:00,  1.41it/s]
          Class      Images      Targets      P      R      mAP@.5      mAP@.5:.95: 100% 1/1 [00:00<00:00,  8.21it/s]
          all           5           5      0.847        1      0.995      0.649

Epoch    gpu_mem      box      obj      cls      total      targets      img_size
99/99    5.16G    0.0296    0.01313        0    0.04274        36          640: 100% 6/6 [00:04<00:00,  1.32it/s]
          Class      Images      Targets      P      R      mAP@.5      mAP@.5:.95: 100% 1/1 [00:01<00:00,  1.26s/it]
          all           5           5      0.836        1      0.995      0.571
Optimizer stripped from runs/exp1/weights/last.pt, 14.8MB
Optimizer stripped from runs/exp1/weights/best.pt, 14.8MB
100 epochs completed in 0.150 hours.
```



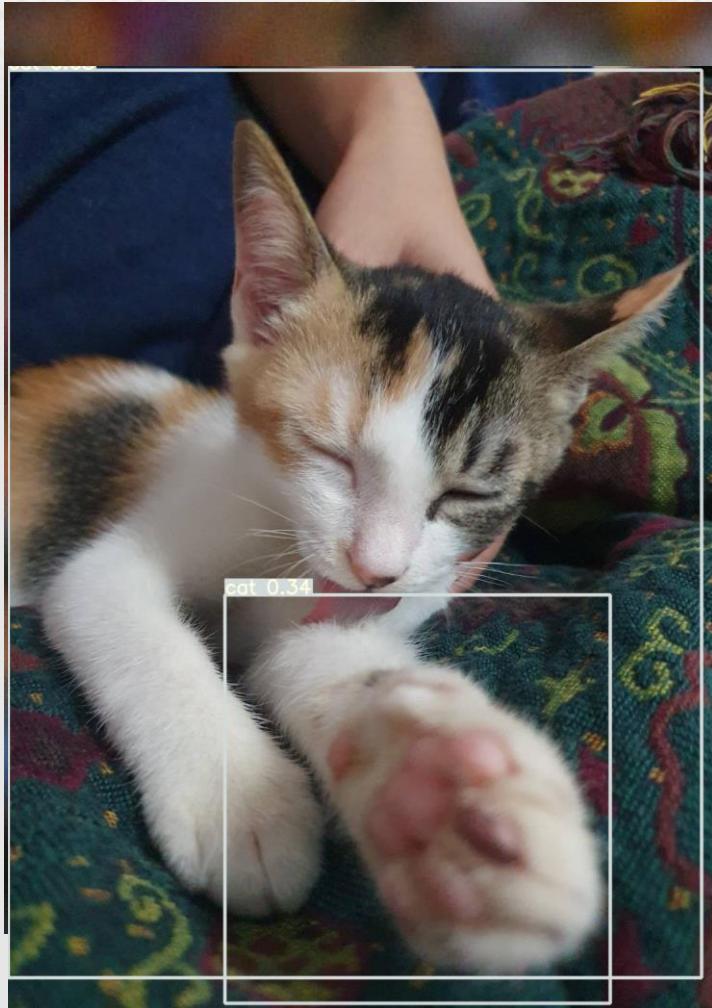


使用自己的圖片進行Inference





探討、問題點在哪裡？



Review

給大家留白一下，思考今天到底學了那些？

1. 用簡單的方式訓練神經網路
2. 使用別人建構好的神經網路
3. 將模型客製化於自己的資料

CAVEDU

完成！

是否感到非常的痛苦 X D