

亞洲大學
資訊工程學系
108學年度畢業專題

題目：開發應用於品管檢查線上
工業端子分類檢測系統

105021022林暉峻

105021100陳泳傑

105021047吳承憲

105021053薛嘉華

指導老師：陳興忠教授

中華民國 108 年 11 月 20 日

專題研究背景及動機

由於資訊技術的進步，造成了網路開創一個非常開闊的發展環境。在這樣的環境下，不知不覺得就會對其所具備的功能、系統有了莫大的興趣，許多的疑問也就不斷產生了。

因應工業瑕疵檢測需求，我們發現了工業上目前極少數的工廠有使用程式軟體來快速檢測瑕疵以達到高檢測效率與精確度，在小組成員的討論下我們決定研究有關工業影像辨識。

計劃目的與範圍

開發一套軟體，說起來好像很簡單，實際上是一件非常大的工程，舉凡程式的撰寫、人機的介面、操作的系統、後續的除錯等等，一邊整合一邊除錯，都是非常花時間精力的過程。我們的理想目標是開發出一套功能具體的工業軟體，其實對我們來說，最重要的目的就是如何在過程中不斷的增加自身的技術和實務經驗，希望透過製作專題的經驗，不僅得到豐富的知識和技術，也可以將這些知識傳與後進，使其更上一層樓。

摘要

圖像識別是開發和測試機器學習的一項重要任務，我們採取的方法是讓計算機自己去完成這樣的過程，而不是手把手地一步步教會計算機如何解釋圖像並翻譯成計算機程序。我們給計算機提供總體結構，讓計算機從經驗中學習。

本專題將介紹如何用Tensorflow識別出影像中的物件，讓Object Detection模型可以偵測出影像中的物件，最耗費時間的是影像的準備。

在使用機器學習來實作影像的物件偵測時，都會需要大量的已知資料集，而通常若要準備這類的資料，初期都會使用人工的方式來篩選。

之後就是Tensorflow識別出影像中的物件，從label的影像區域中，透過CNN(捲積中的各式filters)與圖像進行影像處理計算，電腦可從千變萬化的特徵值中學習到彼此間關聯的資訊。

需求環境

1.1 Python

Python是物件導向程式級高階程式語言，也是直譯式程式語言。Python以強調對程式語言的語句易讀、易懂、易學(簡潔和清晰的語法特點)及加快程式開發的時效，方便使用，可以完成各種難度的應用，並可在大多數的系統中運行，以減少開發及維護成本的觀念進行發展。



圖 1- 1 Python

1.2 OpenCV

OpenCV在影像處理方面應用廣泛，可以讀取儲存圖片、視訊、矩陣運算、統計、影像處理等，可用在物體追蹤、人臉辨識、傅立葉轉換、紋理分析、動態視訊的影像處理等。



圖 1- 2 OpenCV

1.3 Anaconda

Anaconda是一個免費開源的Python和R語言的發行版本，用於計算科學（數據科學、機器學習、大數據處理和預測分析），

Anaconda致力於簡化包管理和部署。Anaconda的包使用軟體包管理系統Conda進行管理。超過1200萬人使用Anaconda發行版

本，並且Anaconda擁有超過1400個適用於Windows、Linux和MacOS的數據科學軟體包。



圖1- 3 Anaconda

1.4 Spyder

Spyder（前身為Pydee）是一個使用Python語言的開放原始碼跨平台科學運算整合開發環境(IDE)。Spyder整合了NumPy，SciPy，

Matplotlib與IPython，以及其他開源軟體。與其他科學數值分析專用IDE（如Matlab或RStudio）相比，Spyder有下列特色：開放

原始碼，以Python編寫並且可以相容於非自由軟體授權。Spyder可以使用附加元件擴充，內建互動式工具以處理資料。跨平台

的特性使得它可以通過Anaconda，Winpython和Python (x,y)（Windows平台）。此外在主流的Linux發行版本例如Ubuntu、

Debian、Fedora、OpenSUSE等等中都有它。

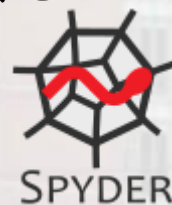


圖1- 4 Spyder

1.5 Keras

Keras是一個用[Python](#)編寫的[開源神經網路](#)庫，能夠在[TensorFlow](#)、[Microsoft Cognitive Toolkit](#)、[Theano](#)或[PlaidML](#)之上執行。

Keras旨在快速實現[深度神經網路](#)，專注於用戶友好、模組化和可延伸性，是ONEIROS（開放式神經電子智慧機器人作業系統）

專案研究工作的部分產物，主要作者和維護者是Google工程師弗朗索瓦·肖萊。肖萊也是Xception深度神經網路模型的作者。

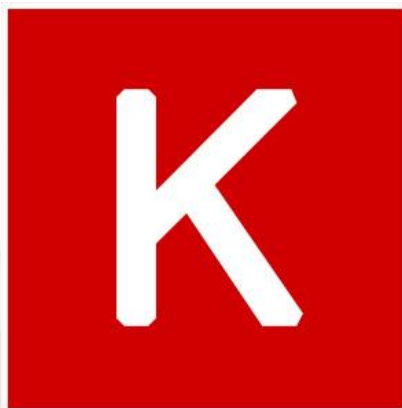


圖1- 5 Keras

人工智慧 (Artificial Intelligence)

Strong AI

Weak AI

機器學習 (Machine Learning)

監督式學習

非監督式學習

增強式學習

深度學習 (Deep Learning)

DNN

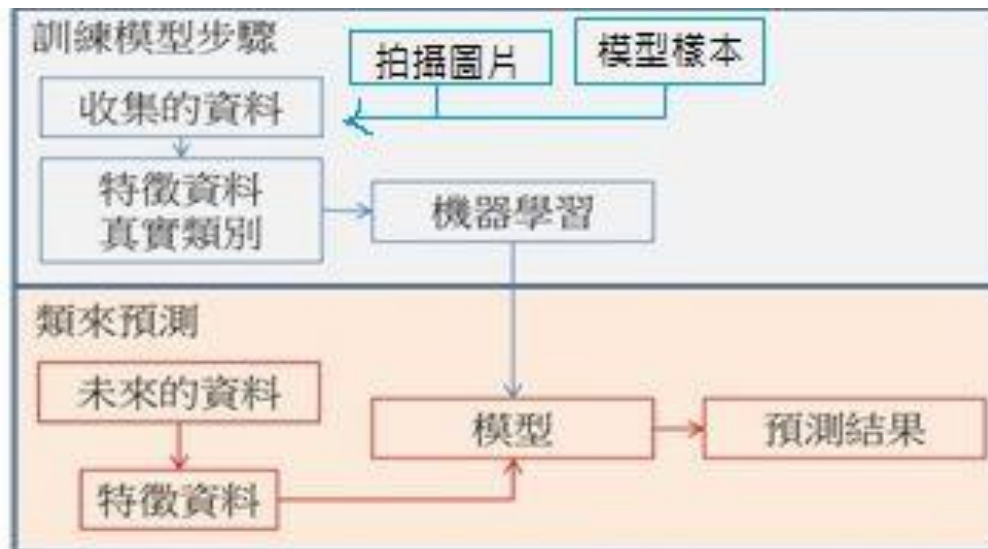
CNN

RNN

大數據 Big Data 分散式儲存

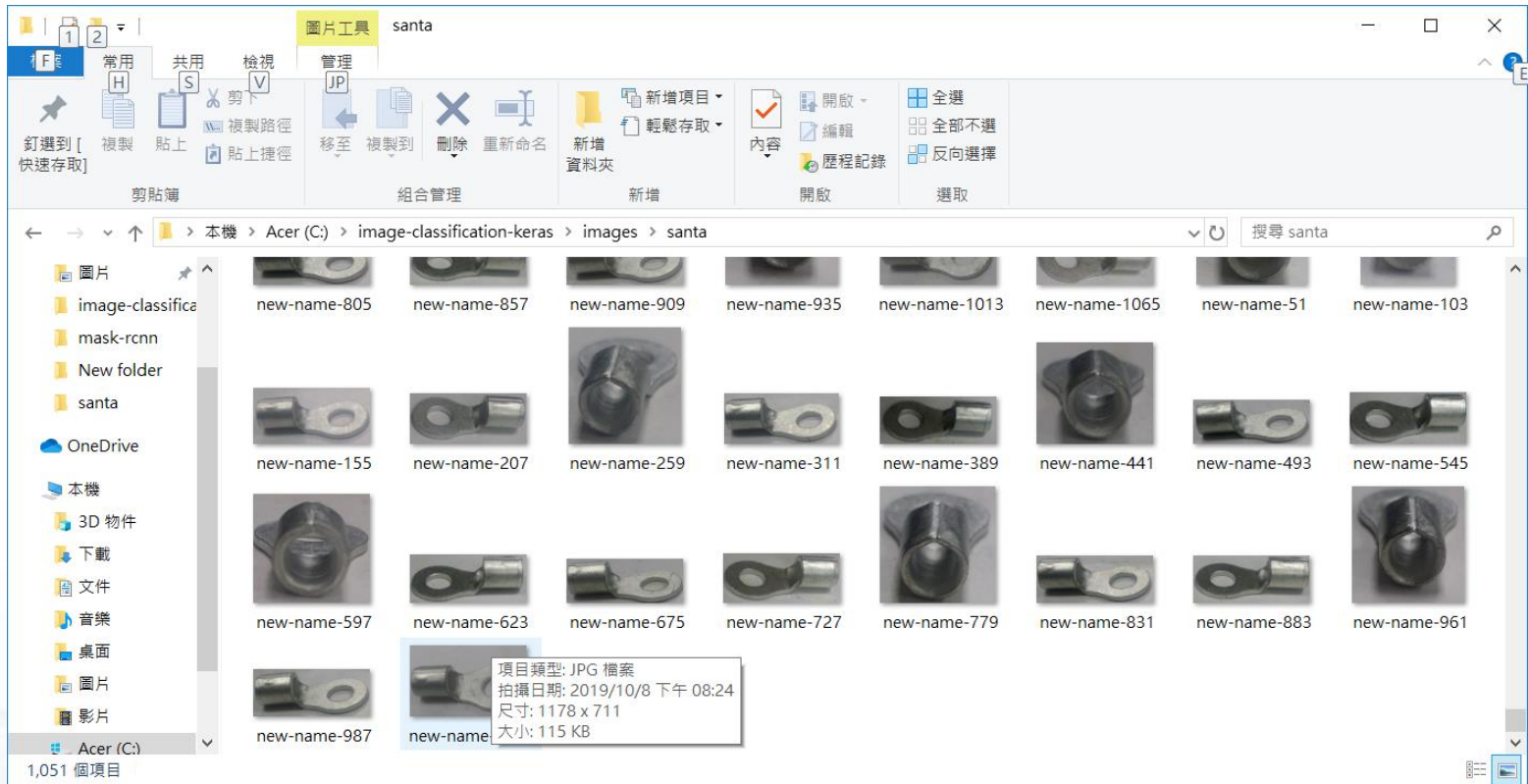
GPU、TPU 平行運算

透過從過往的資料和經驗中學習並找到其運行規則，最後達到人工智慧的方法。機器學習包含透過樣本訓練機器辨識出運作模式，而不是用特定的規則來編程。這些樣本可以在資料中找到。換句話說，機器學習是一種弱人工智慧 (narrow AI)，它從資料中得到複雜的函數 (或樣本) 來學習以創造演算法 (或一組規則)，並利用它來做預測。



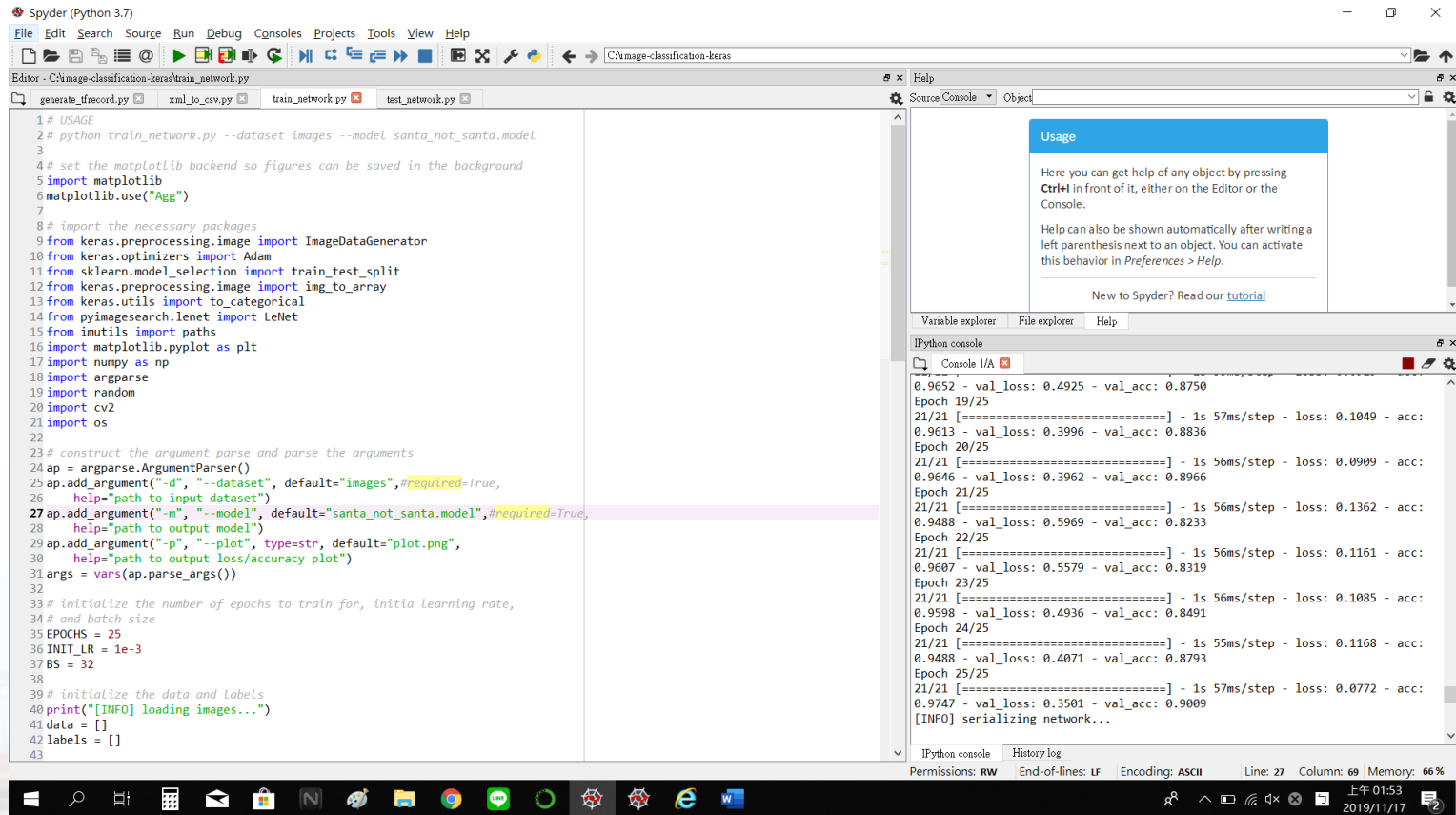
預訓練模型(pre-trained model)是前人為了解決類似問題所創造出來的模型。你在解決問題的時候，不用從零開始訓練一個新模型，可以從在類似問題中訓練過的模型入手。

Select data



訓練圖片 example

Generating a pre-training model



The screenshot displays the Spyder Python IDE interface. The main editor window shows a Python script named `train_network.py` with the following code:

```
1 # USAGE
2 # python train_network.py --dataset images --model santa_not_santa.model
3
4 # set the matplotlib backend so figures can be saved in the background
5 import matplotlib
6 matplotlib.use("Agg")
7
8 # import the necessary packages
9 from keras.preprocessing.image import ImageDataGenerator
10 from keras.optimizers import Adam
11 from sklearn.model_selection import train_test_split
12 from keras.preprocessing.image import img_to_array
13 from keras.utils import to_categorical
14 from pyimagesearch.lemnet import LemNet
15 from imutils import paths
16 import matplotlib.pyplot as plt
17 import numpy as np
18 import argparse
19 import random
20 import cv2
21 import os
22
23 # construct the argument parse and parse the arguments
24 ap = argparse.ArgumentParser()
25 ap.add_argument("-d", "--dataset", default="images", #required=True,
26                 help="path to input dataset")
27 ap.add_argument("-m", "--model", default="santa_not_santa.model", #required=True,
28                 help="path to output model")
29 ap.add_argument("-p", "--plot", type=str, default="plot.png",
30                 help="path to output loss/accuracy plot")
31 args = vars(ap.parse_args())
32
33 # initialize the number of epochs to train for, initial learning rate,
34 # and batch size
35 EPOCHS = 25
36 INIT_LR = 1e-3
37 BS = 32
38
39 # initialize the data and labels
40 print("[INFO] loading images...")
41 data = []
42 labels = []
43
```

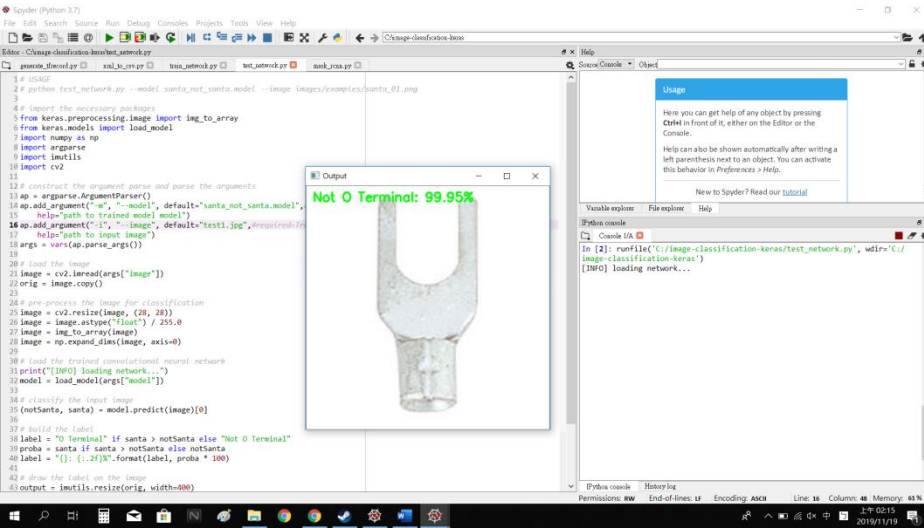
The IPython console on the right shows the execution output, including training progress and accuracy metrics:

```
0.9652 - val_loss: 0.4925 - val_acc: 0.8750
Epoch 19/25
21/21 [=====] - 1s 57ms/step - loss: 0.1049 - acc:
0.9613 - val_loss: 0.3996 - val_acc: 0.8836
Epoch 20/25
21/21 [=====] - 1s 56ms/step - loss: 0.0909 - acc:
0.9646 - val_loss: 0.3962 - val_acc: 0.8966
Epoch 21/25
21/21 [=====] - 1s 56ms/step - loss: 0.1362 - acc:
0.9488 - val_loss: 0.5969 - val_acc: 0.8233
Epoch 22/25
21/21 [=====] - 1s 56ms/step - loss: 0.1161 - acc:
0.9607 - val_loss: 0.5579 - val_acc: 0.8319
Epoch 23/25
21/21 [=====] - 1s 56ms/step - loss: 0.1085 - acc:
0.9598 - val_loss: 0.4936 - val_acc: 0.8491
Epoch 24/25
21/21 [=====] - 1s 55ms/step - loss: 0.1168 - acc:
0.9488 - val_loss: 0.4071 - val_acc: 0.8793
Epoch 25/25
21/21 [=====] - 1s 57ms/step - loss: 0.0772 - acc:
0.9747 - val_loss: 0.3501 - val_acc: 0.9009
[INFO] serializing network...
```

The Windows taskbar at the bottom shows the system time as 01:53 on 2019/11/17.

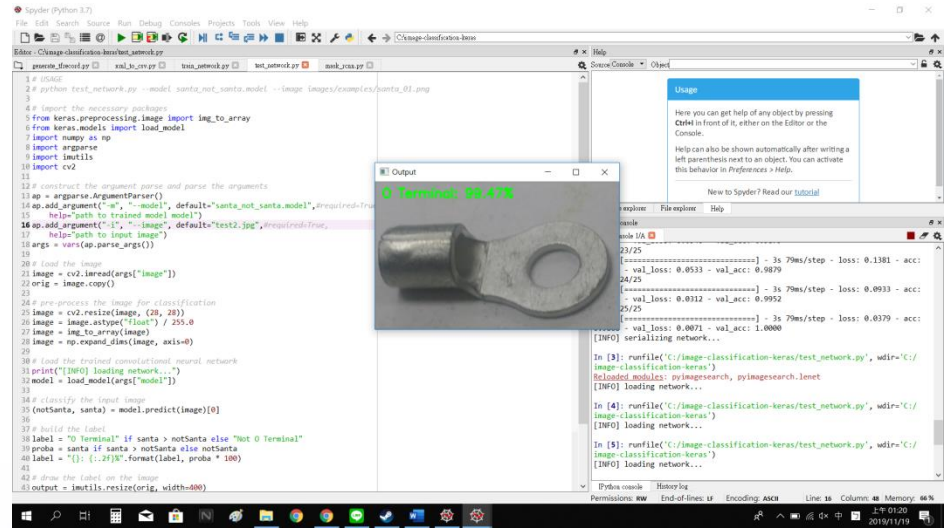
訓練模型主程式

Distinguish results



```
1 # USAGE
2 # python test_network.py --model santa_not_santa_model --image images/examples/santa_01.png
3
4 # Import the necessary packages
5 from keras.preprocessing.image import img_to_array
6 from keras.models import load_model
7 import numpy as np
8 import argparse
9 import imutils
10 import cv2
11
12 # construct the argument parser and parse the arguments
13 ap = argparse.ArgumentParser()
14 ap.add_argument("-m", "--model", default="santa_not_santa_model",
15                 help="path to trained model")
16 ap.add_argument("-i", "--image", default="test1.jpg", required=True,
17                 help="path to input image")
18 args = vars(ap.parse_args())
19
20 # load the image
21 image = cv2.imread(args["image"])
22 orig = image.copy()
23
24 # pre-process the image for classification
25 image = cv2.resize(image, (28, 28))
26 image = image.astype("float") / 255.0
27 image = img_to_array(image)
28 image = np.expand_dims(image, axis=0)
29
30 # load the trained convolutional neural network
31 print("[INFO] loading network...")
32 model = load_model(args["model"])
33
34 # classify the input image
35 (notSanta, santa) = model.predict(image)[0]
36
37 # build the label
38 label = "O Terminal" if santa > notSanta else "Not O Terminal"
39 proba = santa if santa > notSanta else notSanta
40 label = "{}: {:.2f}%".format(label, proba * 100)
41
42 # draw the label on the image
43 output = imutils.resize(orig, width=400)
```

非O型端子



```
1 # USAGE
2 # python test_network.py --model santa_not_santa_model --image images/examples/santa_01.png
3
4 # Import the necessary packages
5 from keras.preprocessing.image import img_to_array
6 from keras.models import load_model
7 import numpy as np
8 import argparse
9 import imutils
10 import cv2
11
12 # construct the argument parser and parse the arguments
13 ap = argparse.ArgumentParser()
14 ap.add_argument("-m", "--model", default="santa_not_santa_model", required=True,
15                 help="path to trained model")
16 ap.add_argument("-i", "--image", default="test2.jpg", required=True,
17                 help="path to input image")
18 args = vars(ap.parse_args())
19
20 # load the image
21 image = cv2.imread(args["image"])
22 orig = image.copy()
23
24 # pre-process the image for classification
25 image = cv2.resize(image, (28, 28))
26 image = image.astype("float") / 255.0
27 image = img_to_array(image)
28 image = np.expand_dims(image, axis=0)
29
30 # load the trained convolutional neural network
31 print("[INFO] loading network...")
32 model = load_model(args["model"])
33
34 # classify the input image
35 (notSanta, santa) = model.predict(image)[0]
36
37 # build the label
38 label = "O Terminal" if santa > notSanta else "Not O Terminal"
39 proba = santa if santa > notSanta else notSanta
40 label = "{}: {:.2f}%".format(label, proba * 100)
41
42 # draw the label on the image
43 output = imutils.resize(orig, width=400)
```

O型端子

小組分工表

小組成員	分工項目
陳泳傑	程式設計、文書處理
林暉峻	程式設計、文書處理
薛嘉華	拍攝照片、蒐集資料
吳承憲	甘特圖、蒐集資料

謝謝觀賞

