

YOLO 應用在 藥物識別分類

109034550 周宛昀



TABLE OF CONTENTS

01

SCENARIO

主題說明

02

MODEL

模型介紹

03

DATA

資料蒐集及標記

04

TRAINING

訓練過程

05

RESULT

結果呈現

06

CONCLUSION

結論及未來展望



01

SCENARIO

主題說明



Background

近年來台灣老年人口比例不斷上升，老年人因年紀漸長，需要服用藥物或是保健食品。但有極大的可能老年人是獨自在家，或是因為年紀較長容易忘記事情，可能會服用錯誤的藥物，進而造成更大的健康上的問題。

5W1H

WHAT

藥物識別

WHEN

老年人吃藥時

WHERE

任何地點

WHY

身旁沒有人協助



HOW

YOLO偵測



02

MODEL

模型介紹

YOLO介紹

YOLO

YOLO(You Only Look Once)是一種關於物件偵測的類神經網路演算法。
本次採用YOLOv3版本。



One Stage

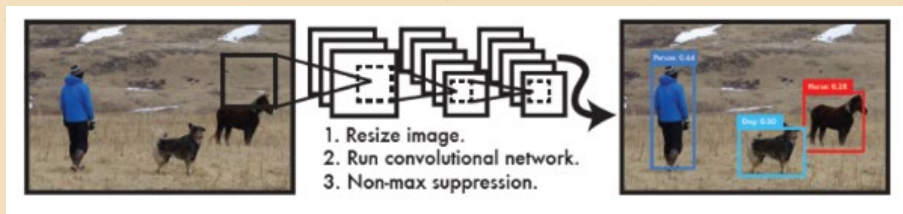
YOLO是one stage的物件偵測方法，物件位置偵測和物件辨識一步到位，也就是一個神經網路能同時偵測物件位置也可以辨識物件。

one stage相較於two stage優點為辨識速度較快，但相對準確度也較低。



Step

1. Resize輸入的圖到32的倍數之正方形
2. 執行一個卷積神經網路
3. 基於模型輸出的信心程度(Confidence)得到偵測結果



YOLO介紹

Grid Cell

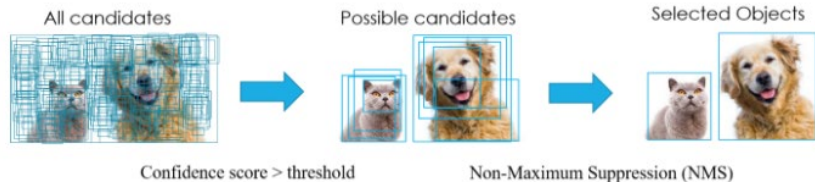
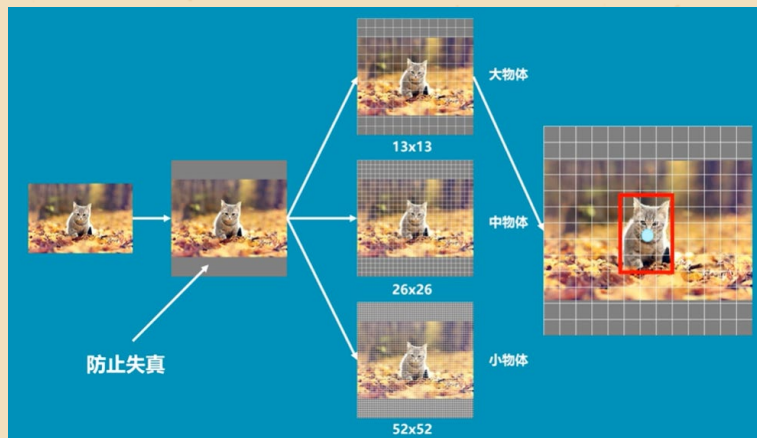
在神經網路內會以三種大小將圖片以網格劃分開來，分別是(13*13)、(26*26)、(52*52)網格。因小物體經過多次卷積後容易失真，故(52*52)網格功用是用來偵測小物體的。

Bounding Box

根據Bounding Box的信心程度，先刪掉一些確定不是物件的Bounding box。再把一些重疊的Bounding Box消除，重覆執行直到每個類別都完成，剩下來的Bounding Box就是選出來的物件。

Final

結合「選出的物件」和「對應的grid cell是什麼類別」，就可以決定這個選出的物件屬於什麼類別





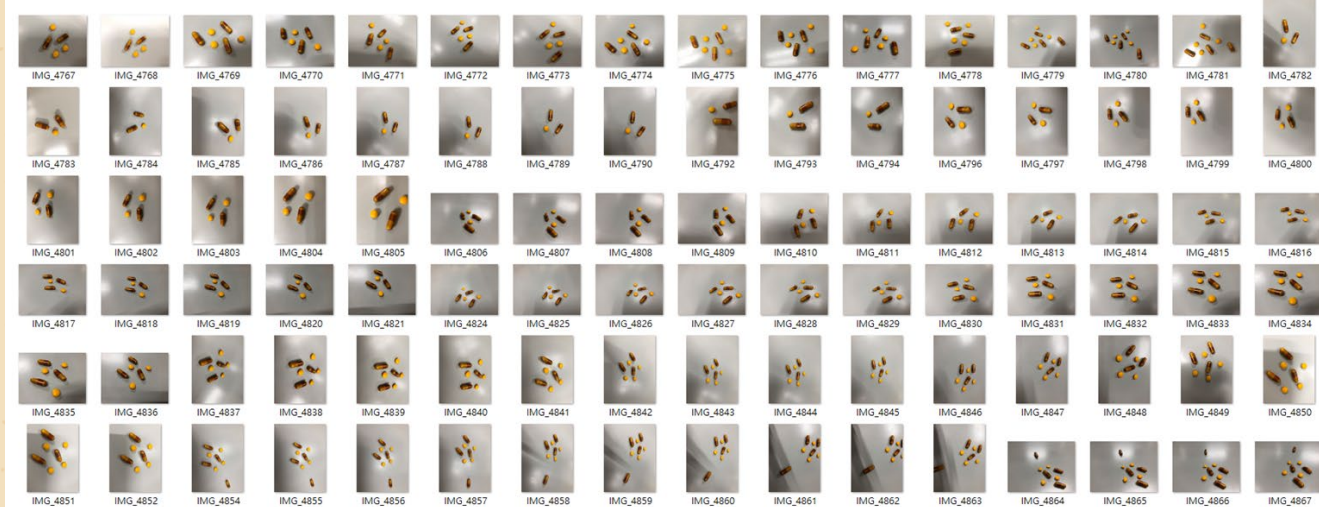
03

DATA

資料蒐集及標記

資料蒐集

- 資料集為自行拍攝之藥物群照片共12張，包含維他命B群及維他命C群。



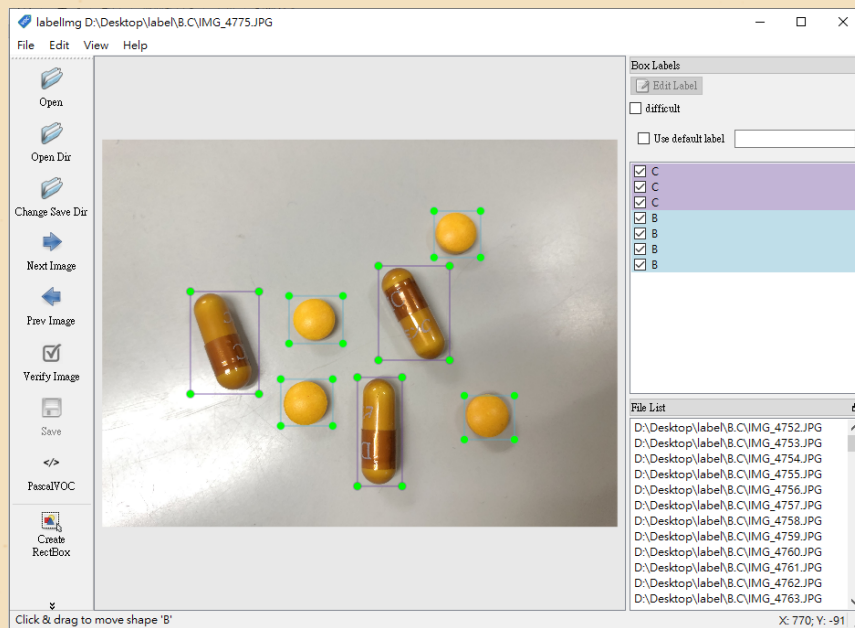
- 109張為訓練集、12張為驗證集、訓練批量為6。

Train on 109 samples, val on 12 samples, with batch size 6.

資料標記

- 利用labelImg套件進行手動標記
- 將所有照片中的藥丸給予對應的標記項目

項目	標記
維他命B群	B
維他命C群	C



標記內容

- `<name>`標記類別
 - `<xmin>`標記項目座標
 - `<ymin>`標記項目座標
 - `<xmax>`標記項目座標
 - `<ymax>`標記項目座標
- 左下
- 右上

標記項目的絕對位置

```
- <object>
  <name>B</name>
  <pose>Unspecified</pose>
  <truncated>0</truncated>
  <difficult>0</difficult>
  - <bndbox>
    <xmin>1284</xmin>
    <ymin>1487</ymin>
    <xmax>1626</xmax>
    <ymax>1801</ymax>
  </bndbox>
</object>
- <object>
  <name>C</name>
  <pose>Unspecified</pose>
  <truncated>0</truncated>
  <difficult>0</difficult>
  - <bndbox>
    <xmin>2012</xmin>
    <ymin>1336</ymin>
    <xmax>2727</xmax>
    <ymax>1751</ymax>
  </bndbox>
</object>
```



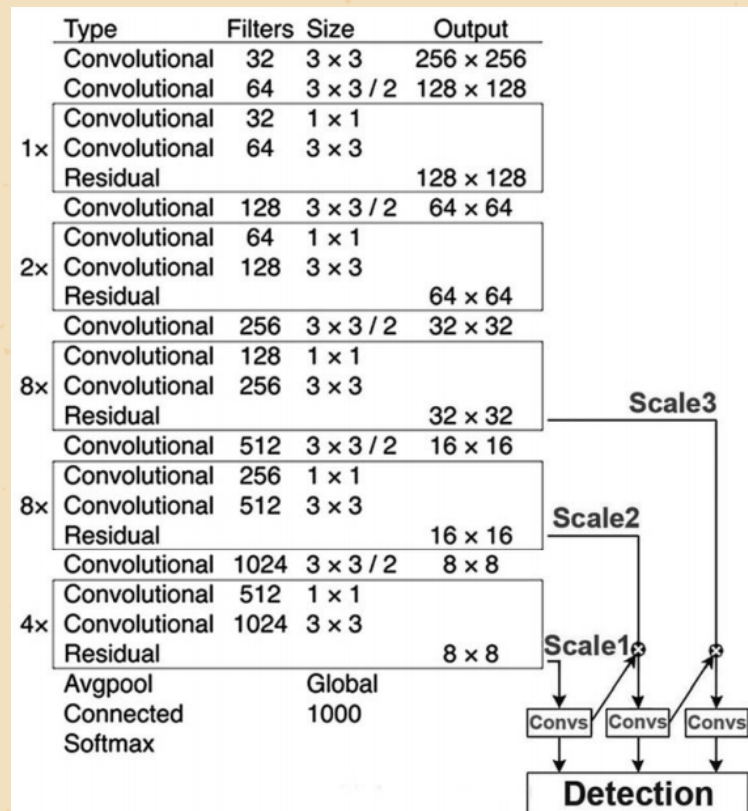
04

TRAINING

訓練過程

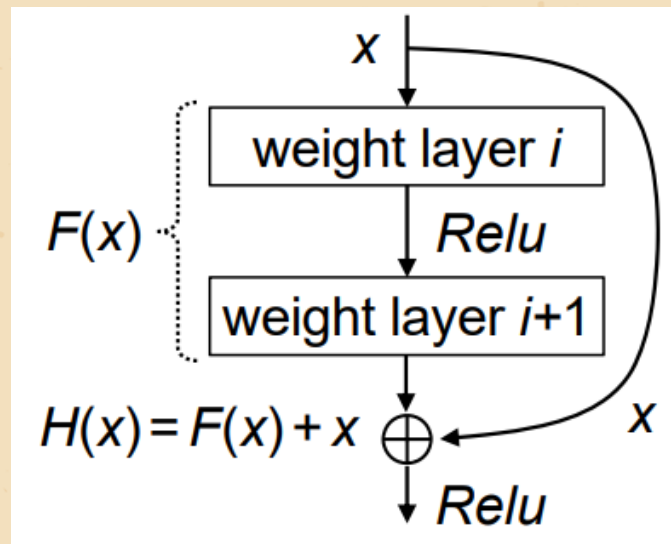
卷積神經網路架構

- 將輸入圖的 bounding box 長寬進行正規化，使得長寬介於 0~1 之間。
- 使用 ResNet 的 residual block (殘差模型)，建構其核心 Darknet-53。
- 每層之積活函數選擇 leaky rectified linear activation (Leaky ReLu)。
- 損失函數為 binary cross-entropy loss。



Residual Block (殘差模型)

- 有短路連接的結構稱為殘差(剩餘)區塊 (residual block)。
- 殘差區塊的輸入端直接拉一條短路線連到輸出端。若輸入端的數值是 x ，殘差區塊產生的數值是 $F(x)$ ，則輸出端的數值即為 $H(x) = F(x) + x$ 也就是在倒傳遞的訓練過程中，若發現 $F(x)$ 值變得很小，代表殘差區塊所找的特徵是沒用的，這時倒傳遞就會弱化這一個殘差區塊。
- 避免出現梯度消失或者爆炸等不利於訓練的情形。



YOLO執行程式碼

```
1 [net]
2 # Testing
3 batch=1
4 subdivisions=1
5 # Training
6 # batch=64
7 # subdivisions=16
8 width=416
9 height=416
10 channels=3
11 momentum=0.9
12 decay=0.0005
13 angle=0
14 saturation = 1.5
15 exposure = 1.5
16 hue=.1
17
18 learning_rate=0.001
19 burn_in=1000
20 max_batches = 500200
21 policy=steps
22 steps=40000,450000
23 scales=.1,.1
24
25 [convolutional]
26 batch_normalize=1
27 filters=32
28 size=3
29 stride=1
30 pad=1
31 activation=leaky
```

```
33 # Downsample
34
35 [convolutional]
36 batch_normalize=1
37 filters=64
38 size=3
39 stride=2
40 pad=1
41 activation=leaky
42
43 [convolutional]
44 batch_normalize=1
45 filters=32
46 size=1
47 stride=1
48 pad=1
49 activation=leaky
50
51 [convolutional]
52 batch_normalize=1
53 filters=64
54 size=3
55 stride=1
56 pad=1
57 activation=leaky
58
59 [shortcut]
60 from=-3
61 activation=linear
62
```

```
63 # Downsample
64
65 [convolutional]
66 batch_normalize=1
67 filters=128
68 size=3
69 stride=2
70 pad=1
71 activation=leaky
72
73 [convolutional]
74 batch_normalize=1
75 filters=64
76 size=1
77 stride=1
78 pad=1
79 activation=leaky
80
81 [convolutional]
82 batch_normalize=1
83 filters=128
84 size=3
85 stride=1
86 pad=1
87 activation=leaky
88
89 [shortcut]
90 from=-3
91 activation=linear
92
93 [convolutional]
94 batch_normalize=1
95 filters=64
96 size=1
97 stride=1
98 pad=1
99 activation=leaky
```

```
500 size=3
501 stride=1
502 pad=1
503 activation=leaky
504
505 [shortcut]
506 from=-3
507 activation=linear
508
509 [convolutional]
510 batch_normalize=1
511 filters=512
512 size=1
513 stride=1
514 pad=1
515 activation=leaky
516
517 [convolutional]
518 batch_normalize=1
519 filters=1024
520 size=3
521 stride=1
522 pad=1
523 activation=leaky
524
525 [shortcut]
526 from=-3
527 activation=linear
528
529 [convolutional]
530 batch_normalize=1
531 filters=512
532 size=1
533 stride=1
534 pad=1
535 activation=leaky
536
537 [convolutional]
538 batch_normalize=1
539 filters=1024
540 size=3
541 stride=1
542 pad=1
543 activation=leaky
544
545 [shortcut]
546 from=-3
547 activation=linear
```


訓練過程

- 共訓練50代
- 每代花費時間約32秒
- loss: 每代的損失
- val_loss: 驗證集的損失

```
Epoch 47/50  
18/18 [=====] - 32s 2s/step - loss: 23.5566 - val_loss: 25.4391  
Epoch 48/50  
18/18 [=====] - 32s 2s/step - loss: 22.1900 - val_loss: 22.7103  
Epoch 49/50  
18/18 [=====] - 33s 2s/step - loss: 23.9353 - val_loss: 24.6406  
Epoch 50/50  
18/18 [=====] - 32s 2s/step - loss: 22.5898 - val_loss: 30.6628
```

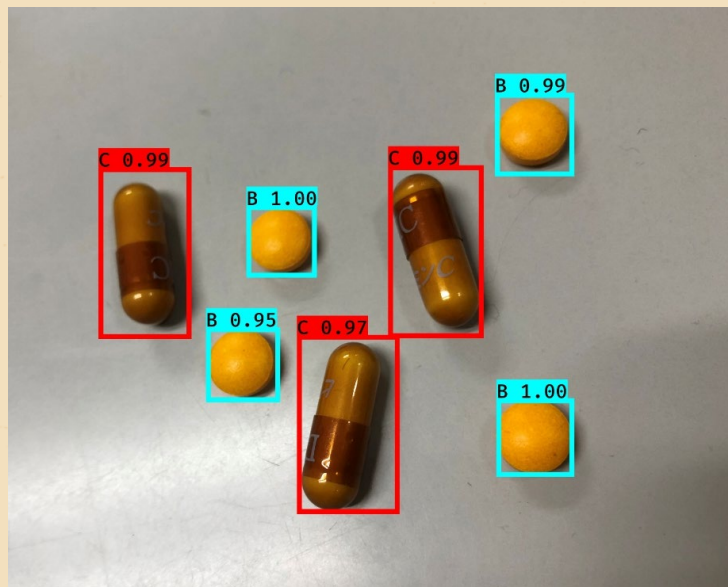


05

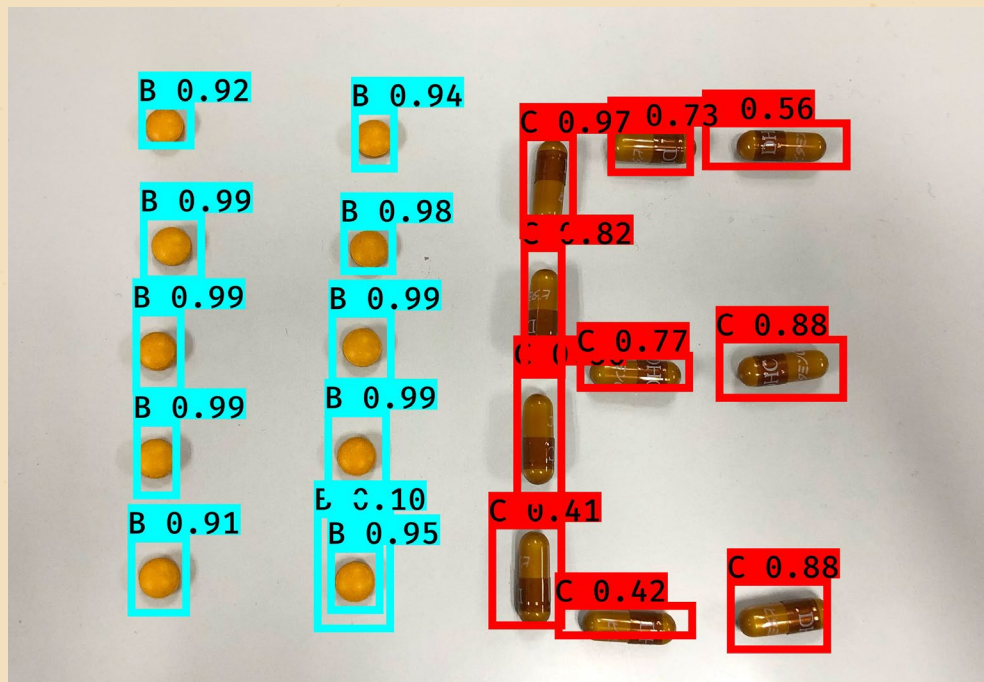
RESULT

結果呈現

測試成果



測試成果



```
Found 21 boxes for img
C 0.41 (2036, 1986) (2260, 2300)
C 0.42 (2230, 2221) (2643, 2329)
C 0.56 (2109, 1529) (2260, 1935)
C 0.56 (2661, 811) (3095, 959)
C 0.73 (2384, 831) (2637, 976)
C 0.77 (2294, 1490) (2634, 1606)
C 0.82 (2130, 1176) (2261, 1568)
C 0.88 (2734, 2151) (3038, 2370)
C 0.88 (2701, 1443) (3085, 1635)
C 0.97 (2127, 852) (2294, 1160)
B 0.10 (1525, 1953) (1759, 2309)
B 0.91 (980, 2024) (1162, 2263)
B 0.92 (1011, 760) (1175, 898)
B 0.94 (1634, 775) (1770, 965)
B 0.95 (1564, 2054) (1728, 2255)
B 0.98 (1602, 1113) (1766, 1265)
B 0.99 (1556, 1658) (1747, 1929)
B 0.99 (994, 1364) (1146, 1615)
B 0.99 (1016, 1083) (1206, 1289)
B 0.99 (996, 1686) (1133, 1920)
B 0.99 (1568, 1368) (1761, 1588)
2.7356699000000013
Yolo inital: 13.501978 sec
Image load: 0.013134 sec
Detect object: 2.736209 sec
```





06

CONCLUSION

結論及未來展望

研究限制及未來展望

- 
- 資料集之樣本數量稍嫌不足，因此在訓練的類別辨識上受限於樣本過少無法達到百分百精準。
 - 藥物本身變異過大，若藥物本身的髒污或色差可能會影響判斷結果。

- 
- 可增加樣本數，提升準確率。未來也想加上更多不同類別之藥物辨識。
 - 污漬顏色影響的問題，日後可以在增強圖像前處理增加對比度，幫助判斷。



THANKS!

Do you have any questions?