



# 卷積神經網路 Convolutional Neural Network & 電腦視覺 Computer Vision Part6

林彥宇&教研處

## 「版權聲明頁」

本投影片已經獲得作者授權台灣人工智慧學校得以使用於教學用途，如需取得重製權以及公開傳輸權需要透過台灣人工智慧學校取得著作人同意；如果需要修改本投影片著作，則需要取得改作權；另外，如果有需要以光碟或紙本等實體的方式傳播，則需要取得人工智慧學校散佈權。

# 課程內容

---

**本日課程：**

- 1. Segmentation**
- 2. Mask R-CNN 手把手**

**延伸閱讀 (Optional)：**

- 1. Matching and Co-segmentation**

# 本次課程結束後你 (妳) 應該會什麼？

---

- **軟實力**

- 理解Segmentation的概念
- 了解Mask R-CNN的方法與程式碼內容

- **硬底子**

- 完成手把手Mask R-CNN



---

# Segmentation

# 理論講授 - Semantic Segmentation

## Experimental Results

- Dataset : Pascal VOC

- Training : [Hariharan et al] collected labels for a larger set of 8498 PASCAL training images.
- Testing : Test set of Pascal VOC 2011 and 2012

	mean IU VOC2011 test	mean IU VOC2012 test	inference time
R-CNN [11]	47.9		
SDS [17]	52.6	51.6	~ 50 s
FCN-8s	<b>62.7</b>	<b>62.2</b>	~ 175 ms



---

# Mask-RCNN 手把手 快速上手

此篇程式碼來源為以下網址，並進行修改

[https://github.com/matterport/Mask\\_RCNN](https://github.com/matterport/Mask_RCNN)

# shapes.ipynb





# Mask-RCNN手把手簡介(一)

---

Mask-RCNN是Instance segmentation，輸出屬於Masks，是Two Stage的演算法，而Mask-RCNN是Faster-RCNN的改良版，除了能夠對於物件進行定位，並且也能對物件類別進行多物件前後景分割。



# Mask-RCNN手把手簡介(二)




若是使用Labelme進行Label, 有兩點必須注意。

第一, Mask-RCNN若使用"Labelme"這個Label工具進行Label時, 所輸出的格式為json,




Labelme有提供方法直接將.Json轉為MaskRCNN所需要的格式資料。

轉換後檔案名稱請統一如下, 並依照數據擺放位置擺放。




cv2\_mask(Label)

 1.png	13 days ago
 2.png	13 days ago
 3.png	13 days ago



labelme\_json

 1_json	13 days ago
 2_json	13 days ago
 3_json	13 days ago

pic(原圖)

 1.png	13 days ago
 2.png	13 days ago
 3.png	13 days ago

json

 1.json	13 days ago
 2.json	13 days ago
 3.json	13 days ago



# Mask-RCNN手把手簡介(三)

---

第二個要注意的部分是Labelme所輸出的Mask png格式是每個pixel為16bit, 此程式只能使用8bit, 所以必須將Masks的圖片進行轉換, Hub有提供程式碼  
“16to8.ipynb”方便轉換使用, 使用方法在程式內有說明。

**提醒！轉換後的Mask, 會變成全黑的圖片是正常的。**



# Mask-RCNN手把手簡介(四)

---

由於Mask-RCNN程式碼多較為繁雜，且多為在處理資料的讀取，這邊只列出如果要改使用自己的資料集必須要修改的程式碼，其他部分就不在這邊一一列出，詳細部分在 `shapes.ipynb` 程式碼中已經有註解幫助大家了解。



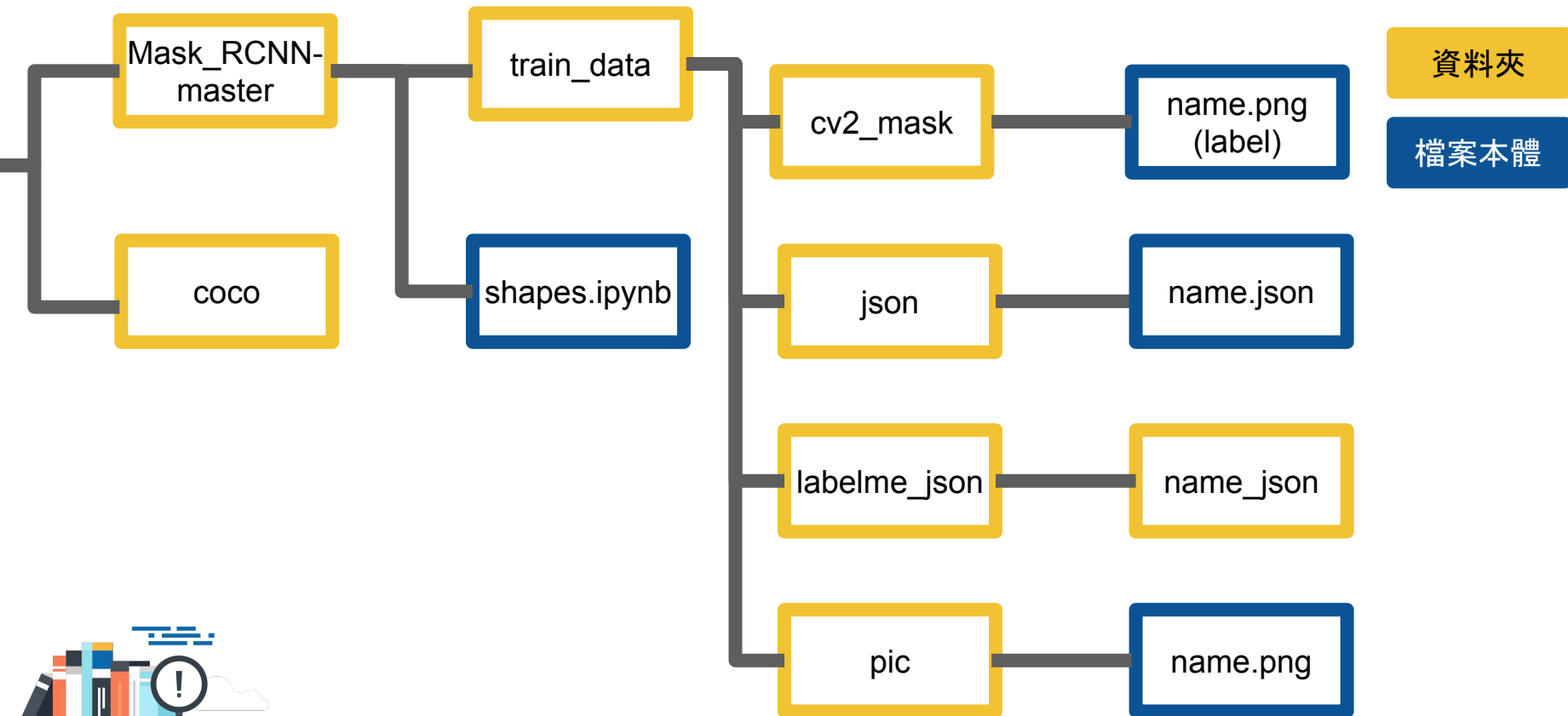
# Mask-RCNN手把手 主要步驟

---

1. 從Label工具得到輸出後，先將Mask(遮罩圖)利用"train\_data" 資料夾底下的"16to8.ipynb"依據裡面的內容說明，將16bit的遮罩轉換成8bit的遮罩供Model使用。
2. 將8 bit的Mask(遮罩)、轉換後原圖、.Json、轉換後的.Json這四樣依序放到指定資料夾。
3. 如果有增加類別或使用自己的新類別，請依照後面的投影片的提示說明進程式碼上的修改。
4. 以上都完成後開始進行訓練
5. 測試辨識時，請依照投影片的提示修改程式碼，之後就可以進行測試。



# 數據置放位置



# shapes.ipynb

---

程式碼總共分為兩個部分，分別為訓練的Cell，以及測試辨識圖片的Cell兩個部分進行。



# shapes.ipynb(一)

以下只列出使用自己的資料集訓練，必須要修改的程式碼部分，紅字部分為類別，若要增加新的訓練資料類別，請依範例依序增加。

```
#####
```

```
#####
```

```
#####
```

```
self.add_class("shapes", 1, "truck") #
```

```
self.add_class("shapes", 2, " Own class_1 ") #
```

```
self.add_class("shapes", 3, " Own class_2 ") #
```

```
self.add_class("shapes", 4, " Own class_3 ") #
```

.....依此類推

```
#####
```

```
#####
```

```
#####
```





# shapes.ipynb(二)

```
#####
```

```
#####
```

```
#####
```

```
for i in range(len(labels)):
```

```
    if labels[i].find("truck") != -1:
```

```
        labels_form.append("truck")
```

```
    elif labels[i].find("your class_1") != -1:
```

```
        labels_form.append("Own class_1")
```

```
    elif labels[i].find("your class_2") != -1:
```

```
        labels_form.append("Own class_2")
```

```
    elif labels[i].find("your class_2") != -1:
```

```
        labels_form.append("Own class_3")
```

.....依此類推

```
#####
```

```
#####
```

```
#####
```



# shapes.ipynb(三)

第一次訓練，請填coco，在產生訓練後的模型後，如果想繼續沿用請改成last。

```
#####
```

```
#####
```

```
#####
```

```
# 第一次訓練，請填coco，在產生訓練後的模型後，如果想繼續沿用  
請改成last
```

```
init_with = "coco" # coco, or last
```

```
#####
```

```
#####
```

```
#####
```



# shapes.ipynb(四)

請依照類別**所有物件種類數+1(背景)**來修改。

```
#####  
#####  
#####  
# Number of classes (including background)  
NUM_CLASSES = 2 # background + 1 class  
#####  
#####  
#####
```

最後，啟動這個Cell的程式碼“**train\_model()**”  
開始訓練吧！



# shapes.ipynb(測試)

請在 shapes.ipynb 的程式碼中，找到執行測試的Cell，並修改以下的程式碼。

```
#####  
#####  
#####  
# All category names  
class_names = ['BG', 'truck']  
  
#####  
#####  
  
#####  
#####  
#####  
  
# Load a random image from the images folder  
file_names = r'train_data/Your_image.png'
```

```
#####  
#####  
#####
```



延伸閱讀-1

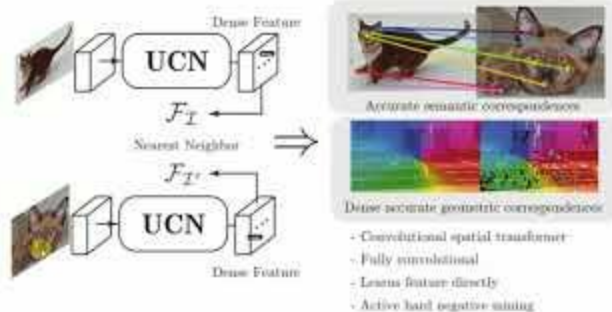
---

# Matching and Co-segmentation

# 理論講授 - Matching and Co-segmentation (Optional)

## Universal Correspondence Network (UCN)

[Choy et al, NIPS'16]



Research Center for Information Technology Innovation, Academia Sinica

170

