題目概述&研究動機

這個final project是源於Kaggle一個由跨國製藥廠Sartorius舉辦的競賽Cell Instance Segmentation. 比賽的目的是使用計算機視覺算法對細胞進行準確的分割，可以幫助量化神經系統疾病對治療的反應程度。目前Sartorius發現其中一種神經細胞系SH-SY5Y的準確率比較低，因為神經細胞具有獨特，不規則的型態，所以難以使用常用的Mask對細胞進行分割。在這個比賽中，舉辦方提供了606張已label和1972張未label的細胞顯微鏡圖片作為訓練集，其中包括三種不同的細胞，分別是Neurons, Astrocytes和Neuroglioblastoma，而label是細胞中不同object所在pixel的位置。參賽者的任務是需要識別隱藏測試案例中240個細胞的objects的位置. 舉辦方還提供了LIVECell比賽的dataset作為transfer learning之用。

這裏提到的object通常會用於神經系統疾病的研究。而參賽者需要用相位對比顯微鏡下細胞的圖像來進行instance segmentation. 如果有正面結果的話，將有助於神經生物學的進一步研究，研究人員可以利用這次競賽的成果測量不同的疾病和治療對神經細胞的影響，也可以有助於發現新的藥物來治療數百萬的病人。

資料預處理

Semantic segmentation 常用的編碼方式有兩種:

* RLE (Run Length Encode):

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 9 | 17 | 25 | 33 | 41 | 49 | 57 |
| 2 | 10 | 18 | 26 | 34 | 42 | 50 | 58 |
| 3 | 11 | 19 | 27 | 35 | 43 | 51 | 59 |
| 4 | 12 | 20 | 28 | 36 | 44 | 52 | 60 |
| 5 | 13 | 21 | 29 | 37 | 45 | 53 | 61 |
| 6 | 14 | 22 | 30 | 38 | 46 | 54 | 62 |
| 7 | 15 | 23 | 31 | 39 | 47 | 55 | 63 |
| 8 | 16 | 24 | 32 | 40 | 48 | 56 | 64 |

Fig. 1 圖解RLE如何操作: ()

首先將圖片的每個像素都按照由上而下，由左而右編號，當出現一個像素的值不是0時，記錄他的編號，並記錄連續幾個非0像素的長度，如圖中的mask會被encode為3, 4, 10, 2, 27, 4, 35, 4, 52, 1, 62, 3。

* Polygon:

(140.5, 62.3)

(135.5, 58.8)

(130.1, .60.3)

(133.2, 40.6)

(136.7, 40.6)

(150.4, 50.6)

Fig. 2 圖解RLE如何操作: ()

多邊形encode就是直接紀錄多邊形的頂點，並把頂點相連，封閉圖形內的範圍就是mask，值得注意的是polygon的座標單位並不是像素，而是圖片長寬的單位，因此可能會出現小數點的數字。如圖二中的圖形就會被encode為130.1, 60.3, 133.2, 40.6, 135.5, 58.8, 136.7, 40.6, 140.5, 62.3, 150.4, 50.6。

本次實驗給的資料兩種都有使用到，其中三種細胞的資料(即與test set相同的細胞種類)是使RLE，而其他種細胞(用於transfer learning的資料)的標註資料則是使用polygon encoding。RLE轉成mask比較簡單，直接建一個跟圖片一樣大的tensor，接著照著ground truth在mask的部分填上1。Polygon的比較複雜，因此是直接使用cocotool的api將他轉成mask。

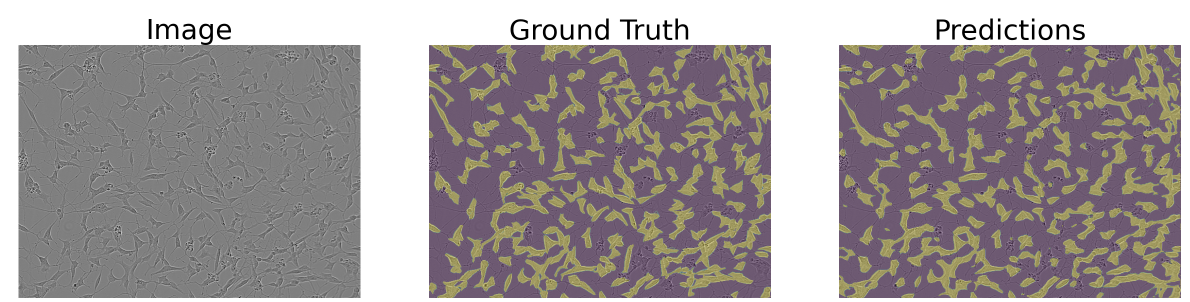
其實給定的資料不多，約600張圖片而已(不含transfer learning的)，因此原本預計要使用較多的data augmentation，例如放大、縮小、平移、左右上下反轉等，藉此防止模型overfitting。這次實驗要預測的是他的mask，因此要同時對原本的image還有mask (甚至使用Mask R-CNN還要對box做augment)做同樣的transformation，但是使用了過多的data augmentation會造成消耗大量的CPU資源，訓練時間會被拉的非常長，因此最後的實驗只有使用左右、上下反轉的data augmentation。

模型

實驗結果與討論

1. 使用Unet架構:





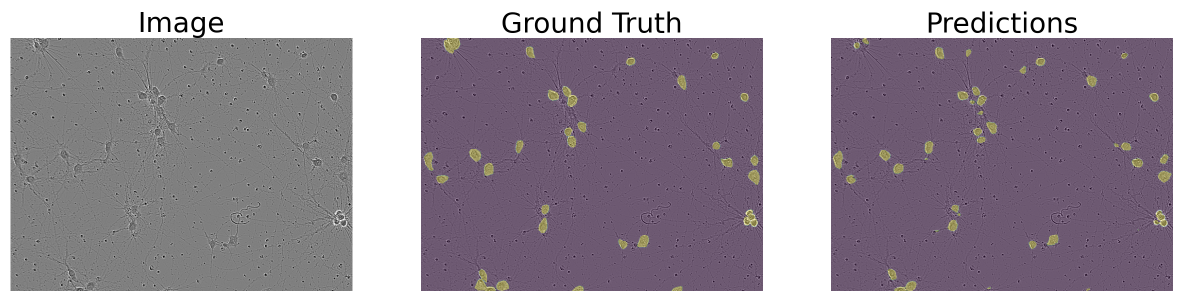
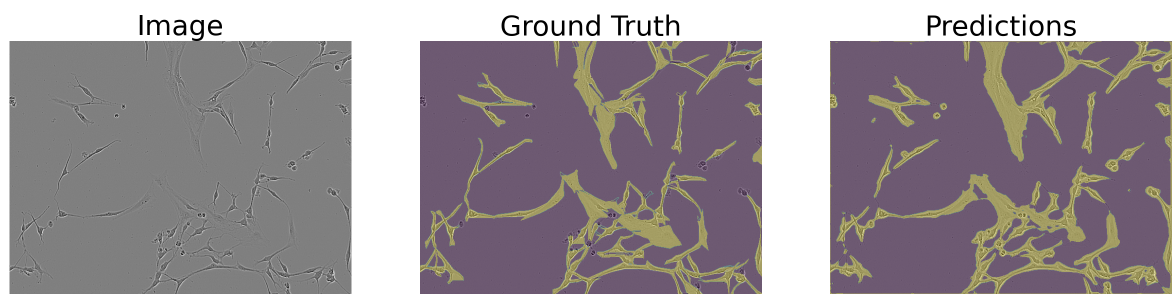
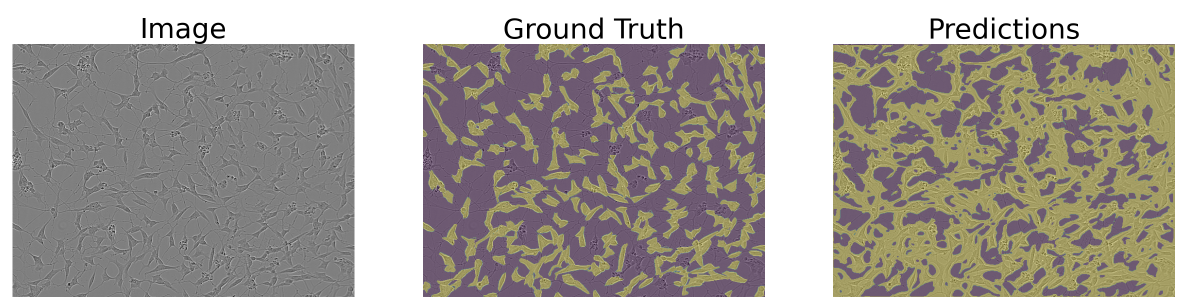
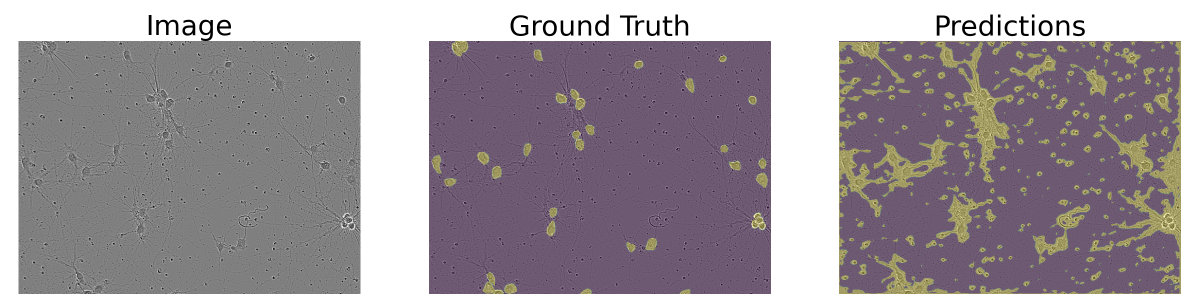


Fig. 1 使用Unet (threshold=0.4) 的結果

可以看到Unet在下面兩種細胞的辨識率其實蠻高的，但是最上面那個細胞做的很差，第一種細胞有很多線條，幾乎都沒有被辨識出來，因此我們試著把threshold調小。







使用Unet (threshold=0.05) 的結果

當我們把threshold調小之後，第一個細胞有顯著的進步，但是對於另外兩種，卻反而退步很多，因此我們推測不同種類的細胞應該會需要不同的threshold。

1. 使用Mask R-CNN架構:

Mask R-CNN與Unet最大的不同是Mask R-CNN除了辨識mask本身之外，還會辨識bounding box，也就是會把每個細胞都框起來，這樣可以有助於模型偵測細胞的大致位置，推測應該可以降低模型學習的難度。

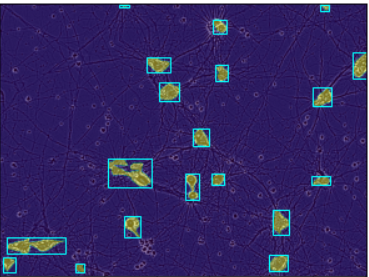
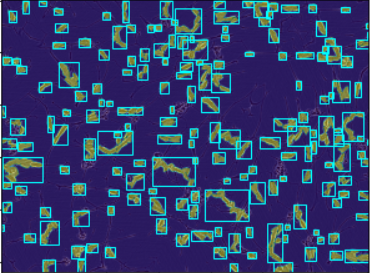
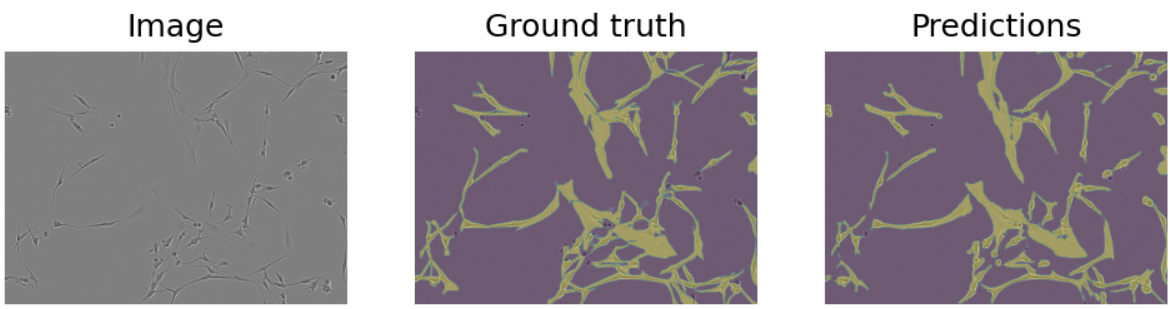
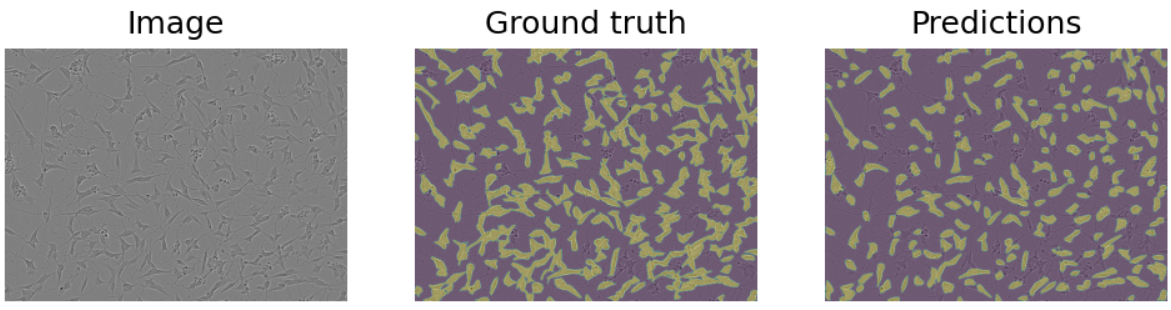


Fig . 視覺化Mask R-CNN的學習目標 (黃色為mask，籃框為box)

****

****

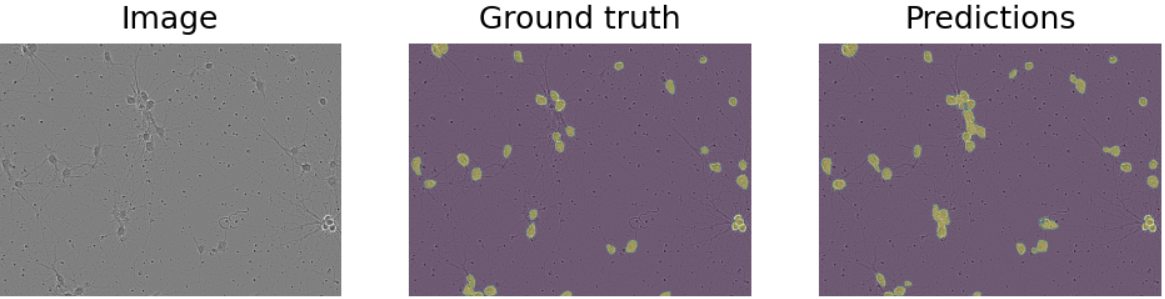
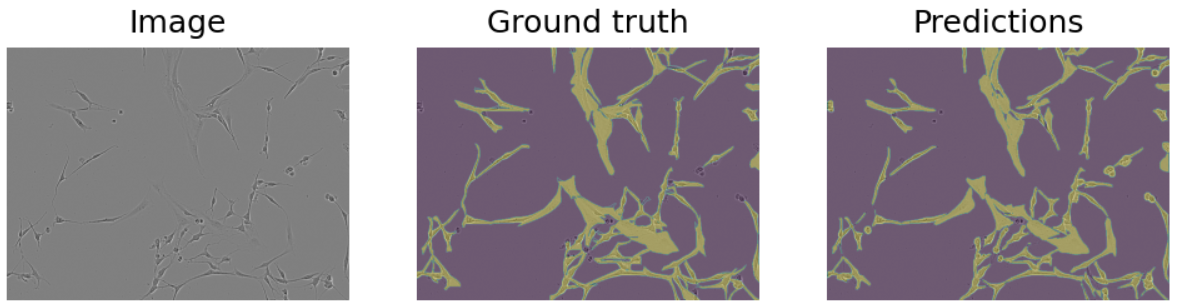
****

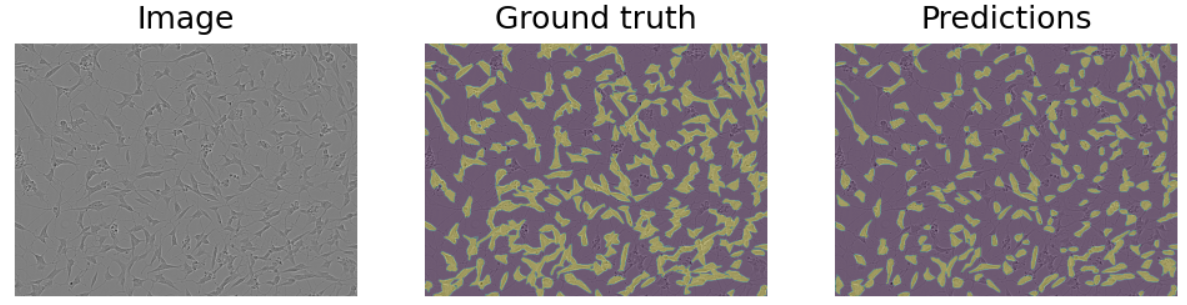
Fig . 使用Mask R-CNN的預測結果

在使用了Mask R-CNN之後，在第一種細胞上的辨識效果大幅進步，另外兩種的辨識效果還是很高，證明了學習bounding box對於辨識mask對於第一種細胞有很大的幫助，也能拉近不同細胞之間mask的分布 (三種細胞都使用同樣的threshold、score，但是都有一定的辨識能力)。

1. 對不同的細胞使用不同的decode規則:

從Unet的例子中，我們推測不同細胞應該要給予不同的threshold。而Mask R-CNN中，我們可以在學習目標中加入分類的Loss，因此可以從我們預測出的細胞種類，給予不同的threshold跟score。





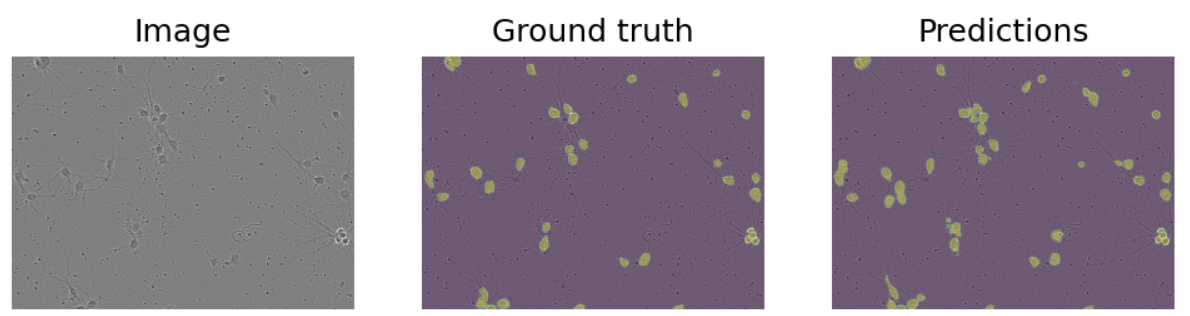


Fig . 對於不同細胞給予不同的threshold、score

由於原本使用Mask R-CNN就已經對於不同細胞都有差不多的辨識能力，因此當我們給了不同的threshold，在視覺化上看起來差不多 (以下實驗皆不放圖，因為視覺化結果差不多)，但是最後計算分數時，卻有很大的差別 (見表)。

1. Transfer learning (不使用transfer data訓練cell type classifier (3 classes))

除了使用test set有的三種細胞資料，這個競賽提供了另外8種細胞的資料。雖然細胞種類不同，但他們要預測的目標是接近的，因此我們推測若讓模型看過這些接近的資料，應該能對辨識效果有所提升。

但是因為這8種細胞並沒有出現在training set裡面，因此若是讓model直接跑在這上面，他細胞分類那邊可能反而有惡化的效果，因此我們在使用transfer data的時候，並不會計算classifier loss的gradient。

1. Transfer learning (使用transfer data訓練cell type classifier (11 classes))

另外，我們使用了另一種方法來處理transfer data種類的問題。我們直接設定模型辨識的細胞種類數為11，接著直接跑在transfer data上，最後再fine-tune在原本的training set。

1. Transfer learning，並使用額外訓練的cell type classifier

這個競賽還有提供大量額外的未標註資料，但是這些資料還是有cell type的，因此若是我們用這些資料另外訓練移個分類器，也許效果可以更好。

1. Refine Mask: 對mask的decode 規則加入更多限制

我們推測一般細胞在圖片裡佔的比例應該是不會差太多的，因此我們統計了圖片裡各種細胞面積的比例，在decode的時候，若該細胞超過training set裡pr98圖片的面積比例，則增加該次decode的threshold，直到他面積的占比小於pr98圖片的mask比例；反之，若面積占比小於training set裡pr2圖片的面積比例，則降低該次decode的threshold，直到他面積占比大於pr2圖片的比例。

期望這個方法可以解決一些離均值的問題。

1. 實驗結果:

|  |  |  |
| --- | --- | --- |
| Methods | Public scores | Private scores |
| UNet | 0.101 | 0.106 |
| Mask R-CNN | 0.201 | 0.198 |
| Mask R-CNN  (with different threshold) | 0.283 | 0.292 |
| Mask R-CNN  (Transfer learning, fixed classifier) | 0.285 | 0.294 |
| Mask R-CNN  (Transfer learning, trainable classifier) | **0.290** | 0.302 |
| Mask R-CNN  (Transfer learning, outer classifier) | 0.288 | **0.303** |
| Mask R-CNN  (Transfer learning, refine mask) | 0.283 | 0.292 |

Table . 所有進行過的實驗在leaderboard上的結果

UNet的效果明顯比Mask R-CNN的效果差，因此可以推論對於細胞的semantic segmentation而言，直接使用類似auto-encoder的效果並不好，可能是因為學習目標太困難。

Mask R-CNN方面，可以看到當我們給予不同種類的細胞不同的threshold跟score，他的效果就有很大的提升 (儘管在視覺化方面的成效不明顯)，證實了我們的推論: 不同細胞的mask有不同的分布，因此需要有不同的decode方式。

Transfer Learning對於辨識也有微幅的進步，因此其他種類的細胞或許對於辨識也有一些幫助。另外使用 11 classes的模型與使用額外訓練的分類器來decode，效果並沒有顯著的差異。但是Refine Mask卻反而對於辨識率惡化，表示test set裡，可能有很多圖片具有較極端的占比，但卻會被新的decode方式限制住，因此辨識率變差。

結論

在本次專案中，我們主要嘗試了UNet以及Mask R-CNN在cell instance segmentation上的效果。初步實驗發現Mask R-CNN透過bounding box的辨識能夠大大地降低模型學習的難度，在表現上也比UNet好很多。之後，我們便以Mask R-CNN為基礎，嘗試了使用不同的threshold、transfer learning、refine mask等方法，也嘗試了使用不同方式訓練出來的classifier。最後以Mask R-CNN搭配transfer learning以及outer classifier以些微之差取得了最好的private score。在此專案中，我們將機器學習用於computer vision，並實作出cell instance segmentation的應用，可以發現電腦雖然能夠達到不錯的辨識率，但距離完美仍然有一段路，辨識結果太容易受到影像中的一些雜訊或是些微改變而有顯著的差距。如何使得computer vision的結果能夠更加穩定，也是未來一個非常重要的課題。

參考資料

Unet: [Updated - Cell Instance Segmentation [pytorch] | Kaggle](https://www.kaggle.com/anastasiiaselezen/updated-cell-instance-segmentation-pytorch)

Mask R-CNN: [🦠 Sartorius - Starter Torch Mask R-CNN [LB=0.273] | Kaggle](https://www.kaggle.com/julian3833/sartorius-starter-torch-mask-r-cnn-lb-0-273)、

[🦠 Sartorius - Torch Mask R-CNN | Kaggle](https://www.kaggle.com/rluethy/sartorius-torch-mask-r-cnn)

Refine mask: [🦠 Sartorius - Classifier + Mask R-CNN [LB=0.28] | Kaggle](https://www.kaggle.com/julian3833/sartorius-classifier-mask-r-cnn-lb-0-28)