**作業二：影像處理與顯著性偵測實習**

0611031 資工10 謝至恆

1. **原理介紹**

**-Deep learning**

深度學習是希望把資料透過多個處理層(layer)中的線性或非線性轉換(linear or non-linear transform)，自動抽取出足以代表資料特性的特徵(feature)，也可以說是模擬出類神經網路。在傳統的機器學習中，特徵通常是透過由人力撰寫的演算法產生出來的，需要經過各領域的專家對資料進行許多的分析及研究，了解資料的特性後，才能產生出有用、效果良好的特徵。這樣的過程就是特徵工程(Feature engineering)。  
深度學習具有自動抽取特徵(feature extraction)的能力，也被視為是一種特徵學習(Feature Learning, representation learning)，可以取代專家的特徵工程所花費的時間。帶著強大的自動特徵抽取的能力，深度學習在以往機器學習一直無法突破的應用，得到非常驚人的優異表現，使以往認為電腦無法做到的事，變成了可能。

深度學習是在最近幾年得到飛躍性的發展，但其實觀念很早就出現了

出處

<http://www.cc.ntu.edu.tw/chinese/epaper/0038/20160920_3805.html>

**-Salient object detection**

圖像顯著性的檢測，這次作業是利用labelme來將圖片作顯著性處理，將圖片中最主要要表達的物體剪取出來，在做Super pixel分割再套入CNN模型判斷PIXEL是否顯著。

**-Super pixel / Super pixel Segmentation**

聽助教講解應該是將圖片中RGB值相近，也就是相素顏色相似的框成一塊，這樣就進而可以減少運算的複雜度，這樣可以降低為度和題除一些異常像速點，缺點就是會損失一些圖片中較為特殊的標示之類的東西，應該有點類似模糊化的概念

概念圖 (左上的取得比較保守，右下娶得比較寬鬆



**-SLIC Super pixel Segmentation**

一種新的生成超像素的方法，比現有方法更快，更高的記憶效率，展示了目前最優的邊界依從性，並提高了分割算法的性能。（SLIC）採用K均值算法生成超像素，相較與其他算法具有兩個重要的區別：

1）通過將搜索空間限制為與超像素大小成比例的區域，顯著地減少了優化中的距離計算的數量。這降低了像素數N的線性複雜度，並且與超像素k的數量無關。

2）加權距離度量組合顏色和空間接近度，同時提供對超像素的尺寸和緊湊性的控制。

出處: <https://blog.csdn.net/zhj_matlab/article/details/52973723>



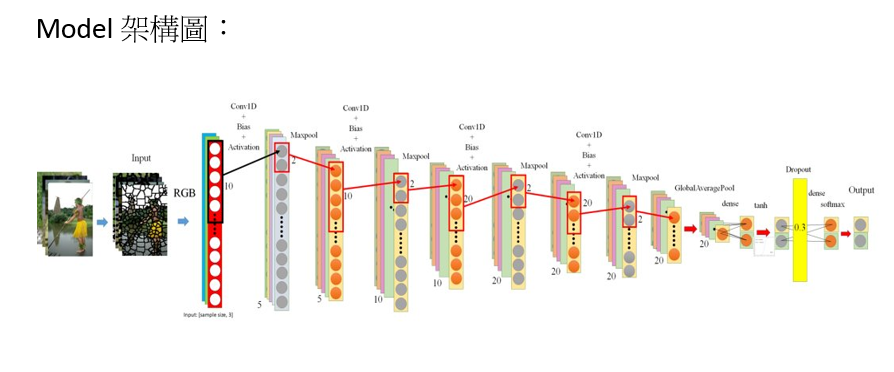
**(二)實作方法**

**-labelme**

照助教獎的方法先下載完打開將圖片丟進去使用

Creat polygons 將圖片中想選取的物件點出來，存成JSON檔，再跑助教給的python程式碼就可以生出黑紅的遮罩，再自行使用軟體將紅色的地方改成白色，即產生Binary mask

**-深度學習網路**



先將要使用的圖片使用matlab程式生成superpixel的RGB張量

再將此一維的圖片資料使用助教給的訓練模型 (Keras.model.fit 方法)以每個 superpixel 的特徵作為輸入，判斷每個 superpixel 經過所時做的 CNN 模 型後，該 superpixel 是否被標為視覺顯著 目標為所有superpixel 之特徵透過所建的 CNN 模型所判斷的顯著結果，會和 groundtruth 所分類的該區域顯著性一致。

先將原圖使用變成一維的RGB樣式後，跑Conv1D + Activation + bias

Nb\_filter代表有幾種filter也就是圖片中下面的數字

Filter\_length代表filter一次會跟多少pixel作處理也就是圖片中框框中的數量

基本上就參考給的程式碼作一些數量的修正還有多添加幾層就完成了

跑之前要先將Epoch設成20，這樣就可以一次產生10,15,20Epoch的model了，

(因為training的時候跑一次要很久我是使用2100張圖片來training)

而且在自己的電腦上跑會比較慢，使用學校的GPU會跑得比較快，

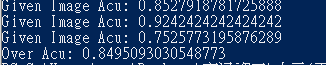
TEST因為測資比較少所以我就直接在自己的電腦上跑了，把想要套的MODEL設定好，改好路徑，就會依照MODEL產生其圖片對應的準確值，最後會產生平均值。也就是這次作業要完成的事情了

**(三)深度學習模型訓練/評估結果**

Epoch 10



Epoch 15



Epoch 20



不同Epoch數產生的準確率略有不同，但大致還在85%上下

比較特別的是在Epoch10的時候，準確度最高的是Epoch最少的10次

15最低，到20又回彈一些，應該是因為參數還在調整當中，有時候會不小心調太多(雖然也只有0.05)，導致Epoch10~15中間準確度下降，隨著Epoch數越來越多，應該會趨近一個相似的值。

**心得:**

這次的作業蠻有趣的，算是第一次接觸這種類神經網路程式，覺得很新奇，原來可以這樣來產生這種類似機器學期的效果。原本看到助教給的程式碼的時候覺得完蛋了，又是新的東西，都看不懂，但後來上網找了一些簡單的範例來看後就比較了解了，又因為程式是助教給極大部分架構，只要能看懂一些關鍵的部分其實就能看懂程式在幹嘛，一開始跑Training的時候還以為我電腦要壞掉了，後來才知道他本來就會跑一段時間。最後做出來結果的時候成就感蠻大的，但如果之後要自己寫程式的話一定會爆炸。這個作業讓我對於影像處理和人工智慧中的類神經網路都有進一步的了解，之前只是聽聽觀念，也覺得這些東西要實際做出來好像複雜到不是我能想像的，沒想到可以實際操作這樣一個架構來將給的圖像顯著化的程式。