

Digital Image Processing

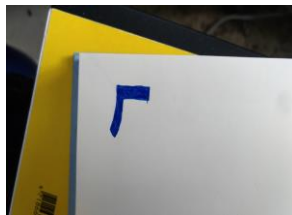
Homework 2

0510002

袁鈺勛

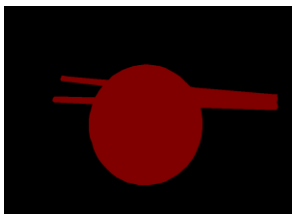
- Part 1

下面 6 張圖是自己拍的照片，最後面兩張自己畫的圖因為前面的圖片較為複雜，所以畫了簡單的圖形。

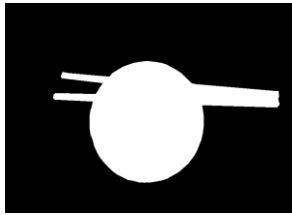


- Part 2

下圖為啟動 LabelMe，open file 再 create polygons 以滑鼠點出的區域產生的 json 經 labelme_json_to_dataset 生成的 ground truth .png file，以 indoor_1 為例。



下圖為將 RGB 的 ground truth file 經 photoshop 轉成的灰階圖片。以魔術棒選取紅色區域，再以油漆桶將其填滿為白色，最後轉換成灰階，以 indoor_1 為例。



● Part 3

訓練所需時間約為 35 分鐘，以 CPU 來執行 training，training data 為 MSRA10K 的前 2000 個 pairs

以下為電腦規格：

1. 作業系統: Windows 10 家用版 64-bit (10.0, Build 17763) (17763.rs5_release.180914-1434)
2. 處理器: Intel(R) Core(TM) i5-6300HQ CPU @ 2.30GHz (4 CPUs), ~2.3GHz
3. 記憶體: 12288MB RAM
4. 內部顯示卡: Intel(R) HD Graphics 530
5. 外部顯示卡: NVIDIA GeForce GTX 950M

以下為 training 環境規格(VirtualBox)：

1. 作業系統: Ubuntu 18.04.2 LTS
2. 處理器: Intel(R) Core(TM) i5-6300HQ CPU @ 2.30GHz, 1 physical processor, 2 cores, 2 threads
3. 記憶體: 4039MB RAM
4. 圖形控制器: VMSVGA

● Part 4

以下為 test 取得的 accuracy：

1. Indoor_1: 0.7577319587628866
2. Indoor_2: 0.6945812807881774
3. Outdoor_1: 0.9754901960784313
4. Outdoor_2: 0.9705882352941176
5. Artificial_1: 0.9831460674157303
6. Artificial_2: 0.9666666666666667

心得：

這次拍的照片在 indoor 和 outdoor 上面並沒有特別考慮什麼，就是想到了就拍，但 artificial 那想說前面的兩種照片都是偏向較複雜的圖案或背景，所以就手畫了簡單的圖案，而且背景基本上保持乾淨與統一，使圖案

和背景對比加大，看成果會不會真的比前面的 4 個照片好很多。訓練出來後 artificial 的準確率確實都滿高的，可見對比明顯且細節較少是可以簡單辨別出來的，而 outdoor 的拍攝都是以周邊可看到的環境為背景，雖然細節比較多，但在 training 的照片中也有出現類似以自然為背景的照片，所以 outdoor 測試出來成果也不差。可是 indoor 的測試結果就很差，我認為主要有 1. 以肉眼點出 salient object 位置，2. 照片特殊，3. 圈選的位置不太對，4. 用來 training 的照片沒有類似的這四點。第一點也可以套用到 outdoor 和 artificial，可能是因為以肉眼方式點出的輪廓其實並沒有貼合 salient object，所以導致訓練出來的神經網絡找到的和肉眼點出來的有所差異，使準確率下滑。第二點是針對 indoor_2，這張照片我認為特殊的點是塑膠袋，因為塑膠袋不太明顯，所以我在以肉眼點輪廓時有些區域其實我不太確定有沒有塑膠袋存在，而且在 training 照片中鮮少看到有類似這張照片部分區域是和背景相似的，所以我認為可能是這原因導致這張照片對訓練出來的神經網絡來說是不太會處理的狀況。第三點則是針對 indoor_1，因為這張照片其實有一個區域是筷子和碗夾出的小區塊，而這個區塊在點 salient object 時是沒有點出來的，但在沒有用來 training 的照片中有看到類似的，所以我認為可能是這原因導致他準確度下降。第四點在第二、三點也有提到過，artificial 因為構造簡單所以很好辨別，而 outdoor 因為在 training 的照片中有出現大量類似的圖片，所以準確率高，但 indoor 的照片在 training 中卻比較少類似的，使得訓練出來的參數不太能應對 indoor 的情況，所以用來 training 的照片要愈多樣化愈好，而且數量也不能太少，才可以使得神經網絡能更準確的辨認出各式照片中的 salient object。經過這次的作業，我學到了在 ubuntu 上利用 tensorflow 以及其推薦的 virtual environment 來跑 python 的 code，並以此環境訓練影像處理相關的神經網絡來觀察 salient object 的成效，以及在前處理圖片時會將圖片切成 superpixels 來簡化和加速訓練。這次作業遇到的 CNN 深度學習比我之前在機器學習遇到的三層神經網絡複雜多了，光是 2000 個資料在 2 cores 的 vm 下就要跑 35 分鐘，而且在機器學習時只是學 feedforward propagation 和 backward propagation，這次居然還出現了 convolution，果然深度學習相較於機器學習複雜程度大幅上升。這也是我第一次遇到 superpixels 這個概念，果然以這神經網絡的複雜度來看，如果不將圖片切成 superpixels，說不定用 GPU 加速來跑也要跑很久，更何況我的環境還是雙核心的 vm。只是神經網絡感覺就像個黑盒子，跑一跑就跑出了結果，感覺看不太到裡面到底發生了什麼。