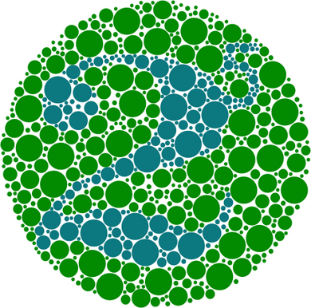
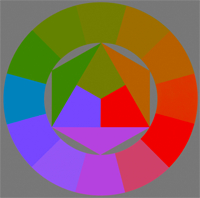
**Computer Vision HW1 Report**

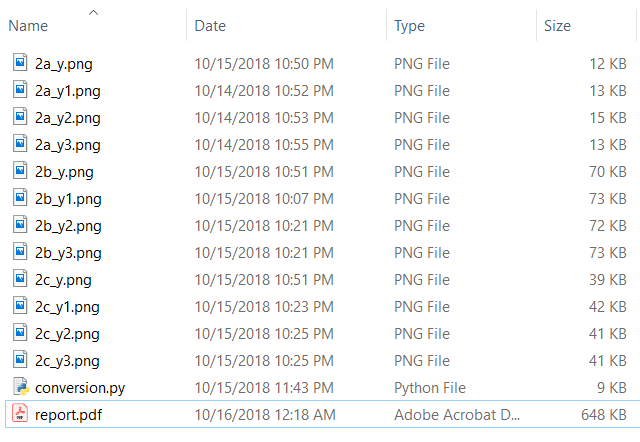
B05901182 電機三 潘彥銘

1. **Input images**



2a.png 2b.png 2c.png

1. **How to run my code?**

****

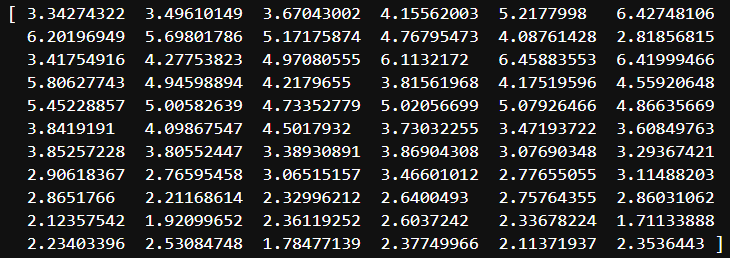
進入本作業資料夾後，會看到上圖的圖檔與文件檔

**Step1:** 將測試圖片拖曳進此資料夾。

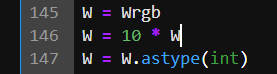
**Step2:** 進入終端機，cd到此作業資料夾後，輸入**conversion.py [intput file name]** ，程式執行完後會有前三高票的係數組合的灰階圖，檔名分別是 **’first\_place\_’** + **[intput file name]** 、 **‘second\_place\_’ + [intput file name]** 以及 **‘third\_place\_’ + [intput file name]**

1. **How I implement the local minima selection?**

在每一場比賽裡，66個候選人經過joint bilateral filter之後，得到66張圖片，將這些圖片與原圖做joint bilateral filter得到的圖片矩陣相減取norm得到cost，我把這66個cost存入一個numpy array叫做**cost\_board** (如下圖一)。

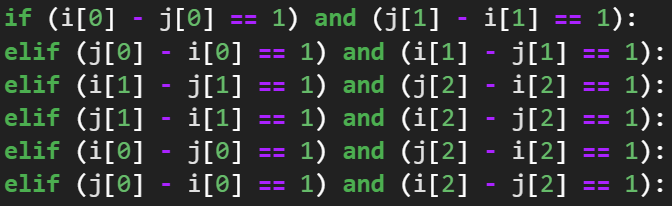


圖一 (以2a.png在σs = 1、σr = 0.05為例)



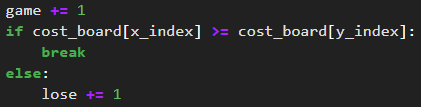
圖二

接下來把原來的66個候選人(Wr, Wg, Wb)全部乘以10，再用astype(int)這個函式把numpy array轉成int(如上圖二)，**以避免之後的浮點數精度問題**。然後用兩個for loop將候選人們的cost一一比對，並用圖三的六個判斷式判斷j是否為i的鄰居 (i為j皆為一組(Wr, Wg, Wb)，i的鄰居找法就是把i的Wr 、Wg、 Wb分別+1或-1，且還要符合Wr +Wg+ Wb=1)。



圖三

每個判斷式內都會有圖四的架構，game可以視為i與鄰居比cost大小的次數，而lose代表i總共輸了幾次，當game == lose時，則表示i每次比大小都輸，則可把i的cost視為local minimum。

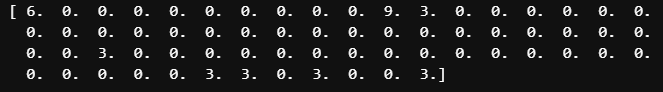


圖四

1. **Output images and corresponding weight combinations**

**(1)2a.png:**

**2a.png**投票結果



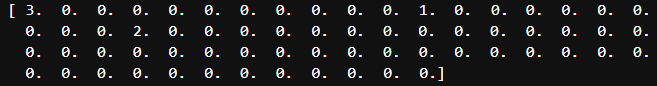
**2a\_y1.png 2a\_y2.png 2a\_y3.png**



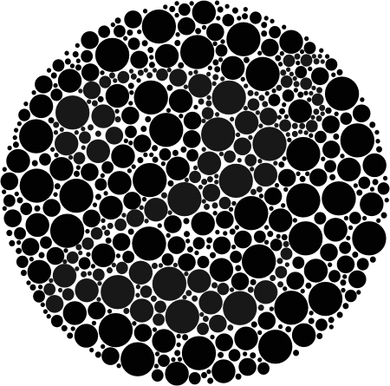
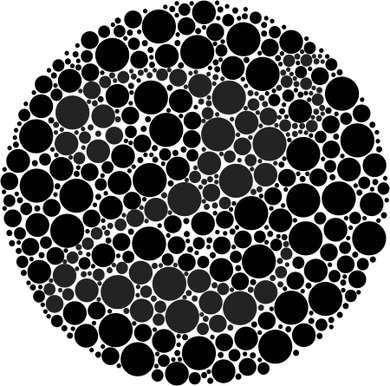
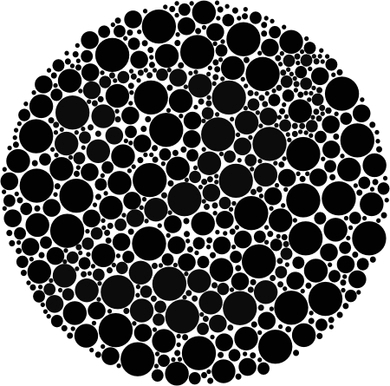
**9**票, **[ 0. , 1. , 0. ] 6**票, **[ 0. , 0. , 1. ] 3**票, **[ 1. , 0. , 0. ]**

**(2)2b.png:**

**2b.png**投票結果



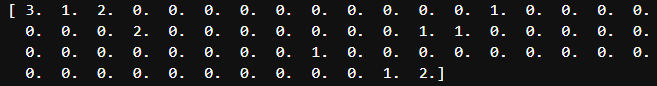
**2b\_y1.png 2b\_y2.png 2b\_y3.png**



**3**票, **[ 0. , 0. , 1. ] 2**票, **[ 0.2 , 0. , 0.8 ] 1**票, **[ 0.1 , 0. , 0.9 ]**

**(3)2c.png:**

**2c.png**投票結果



**2c\_y1.png 2c\_y2.png 2c\_y3.png**



**3**票, **[ 0. , 0. , 1. ] 2**票, **[ 0.2 , 0. , 0.8 ] 2**票, **[ 0. , 0.2 , 0.8 ]**

1. **Comments to this assignment**

想請問一下助教之後會不會公布您寫的code，想看看助教的coding style和寫程式的思維，謝謝助教!