

多媒體資訊系統 HW1

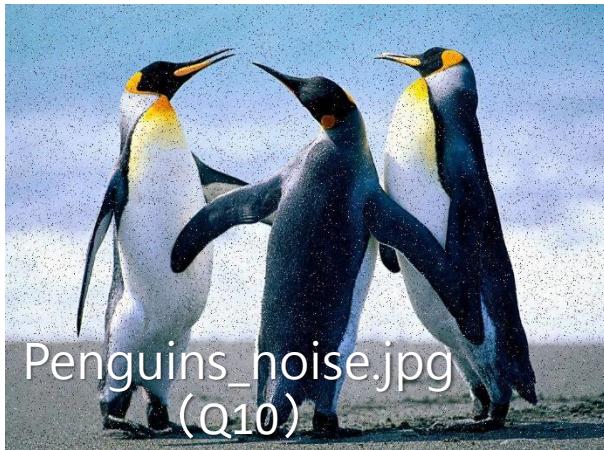
Image

HW1作業說明

- 使用語言：Python (2.7)
- 建議搭配套件：Python Imaging Library
 - Tutorial : <http://effbot.org/imagingbook/introduction.htm>
- 繳交方式：
將程式碼(.py)上傳至WM5數位平台作業區
- 分組：1~2人一組，同組作業分數相同（一組交一份）
- 繳交期限：2017/11/05 23:59 前

HW1作業說明

- 程式名稱：[HW1_學號.py](#) (若多人一組，則為[HW1_學號一_學號二.py](#))
- 輸入：Q1~Q11為 [Penguins.jpg](#) (Q10為 [Penguins_noise.jpg](#))
Q12為 [Elsa.jpg](#) 及 [Chrysanthemum.jpg](#)
- 輸出：
對於每個題目，輸出影像檔名依序為Q1.jpg、Q2.jpg、Q3.jpg至Q12.jpg。
若題目中有分小題(Q5, 6, 7, 11)，檔名則為Q5-1.jpg、Q5-2.jpg……



PIL簡易基本操作 — 讀入存取

```
from PIL import Image
```

```
im = Image.open("Penguins.jpg") # 讀入影像  
print im.format, im.size, im.mode # 影像格式、尺寸、模式
```

```
im.save("Penguins_new.jpg") # 儲存影像
```

PIL簡易基本操作 — 使用Loops讀每個pixel

```
from PIL import Image
```

```
im = Image.open("Penguins.jpg")
```

```
pixel = im.load()
```

```
width, height = im.size
```

回傳像素

得到影像的寬度長度

```
for x in xrange(width):
```

```
    for y in xrange(height):
```

```
        R, G, B = pixel[x,y]
```

取得該pixel的值RGB

(這裡可對pixel[x,y]做修改，例如: pixel[x,y]=(255,255,255))

```
im.save("Penguins_new.jpg")
```

Q1. RGB三層企鵝



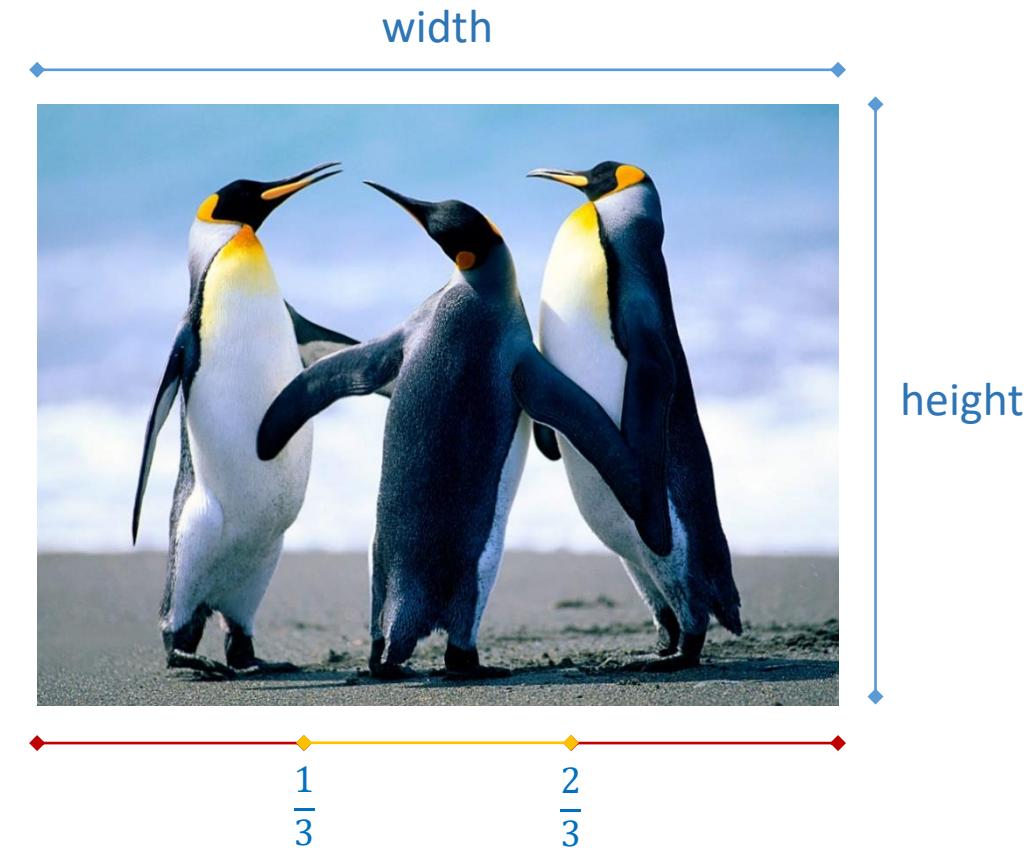
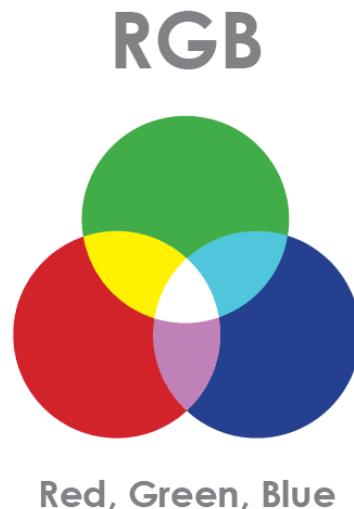
Penguins.jpg



Q1.jpg

Q1. RGB三層企鵝

- 將影像垂直切成三等份
 - 第一等份內影像的pixel **R**變為原來的三倍
 - 第二等份內影像的pixel **G**變為原來的三倍
 - 第三等份內影像的pixel **B**變為原來的三倍



Q2. CMYK四層企鵝



Penguins.jpg



Q2.jpg

Q2. CMYK四層企鵝

- 將RGB影像藉由PIL函式(convert)或其他方法轉成CMYK
`im = Image.open("Penguins.jpg").convert("CMYK")`
- 可以用`im.mode`檢查影像模式
- 將影像垂直切成四等份
 - 第一等份內影像的pixel，C變為原來的三倍
 - 第二等份內影像的pixel，M變為原來的三倍
 - 第三等份內影像的pixel，Y變為原來的三倍
 - 第四等份內影像的pixel，K變為原來的三倍



Q3. 負片(互補色)企鵝



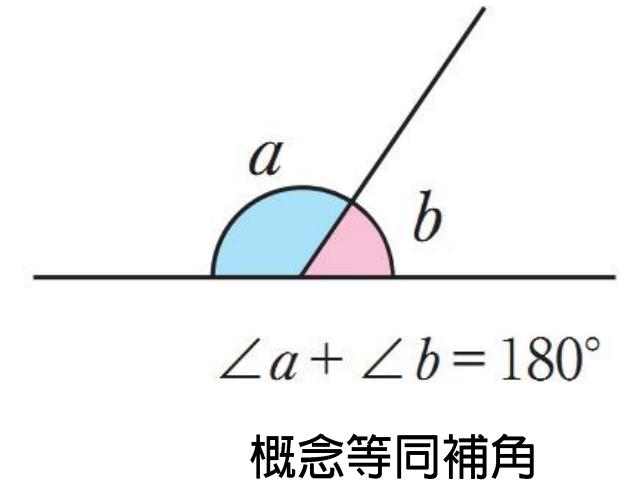
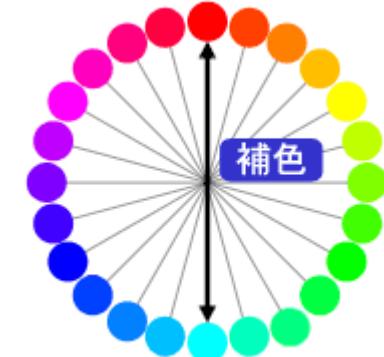
Penguins.jpg



Q3.jpg

Q3. 負片(互補色)企鵝

- 色相環中，相對的顏色即為補色
也就是兩個顏色的RGB值相加會等於 $(255,255,255)$
- 舉例來說：
白色 $(255,255,255)$ 的互補色為黑色 $(0,0,0)$
黃色 $(255,255,0)$ 的互補色為藍色 $(0,0,255)$
- 負片：將影像每一個pixel轉成它的互補色
- 分別將用255減去原R、G、B值即可



Q4. 馬賽克企鵝



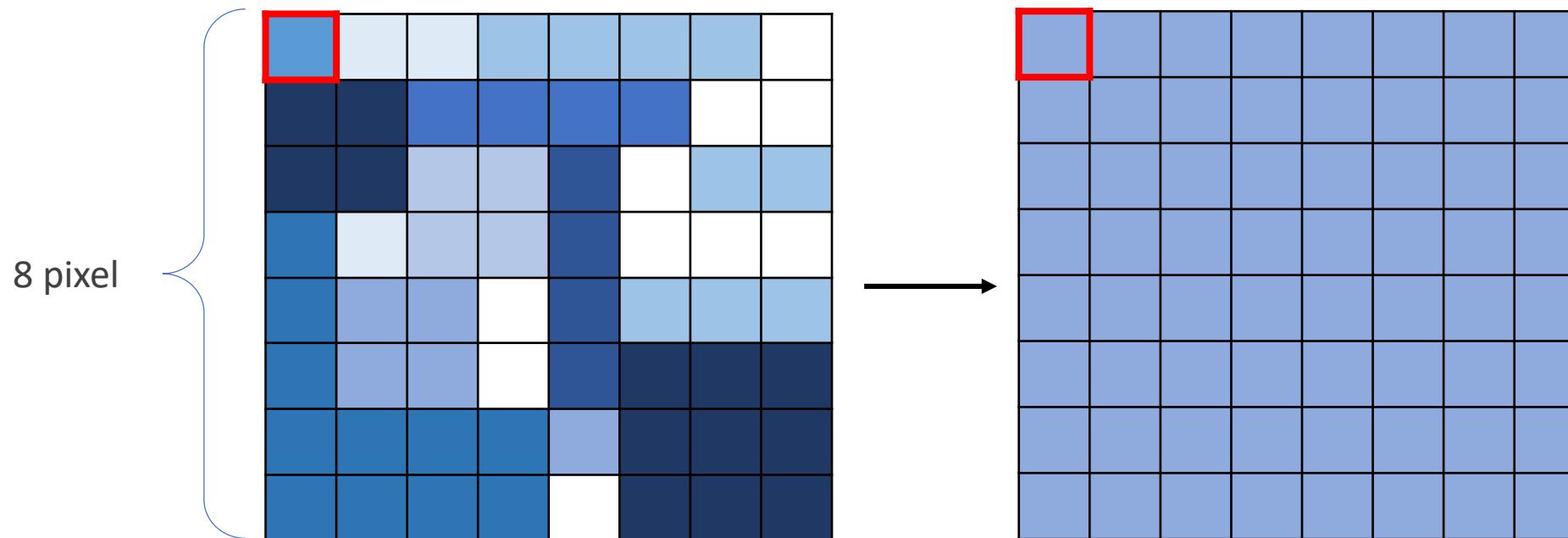
Penguins.jpg



Q4.jpg

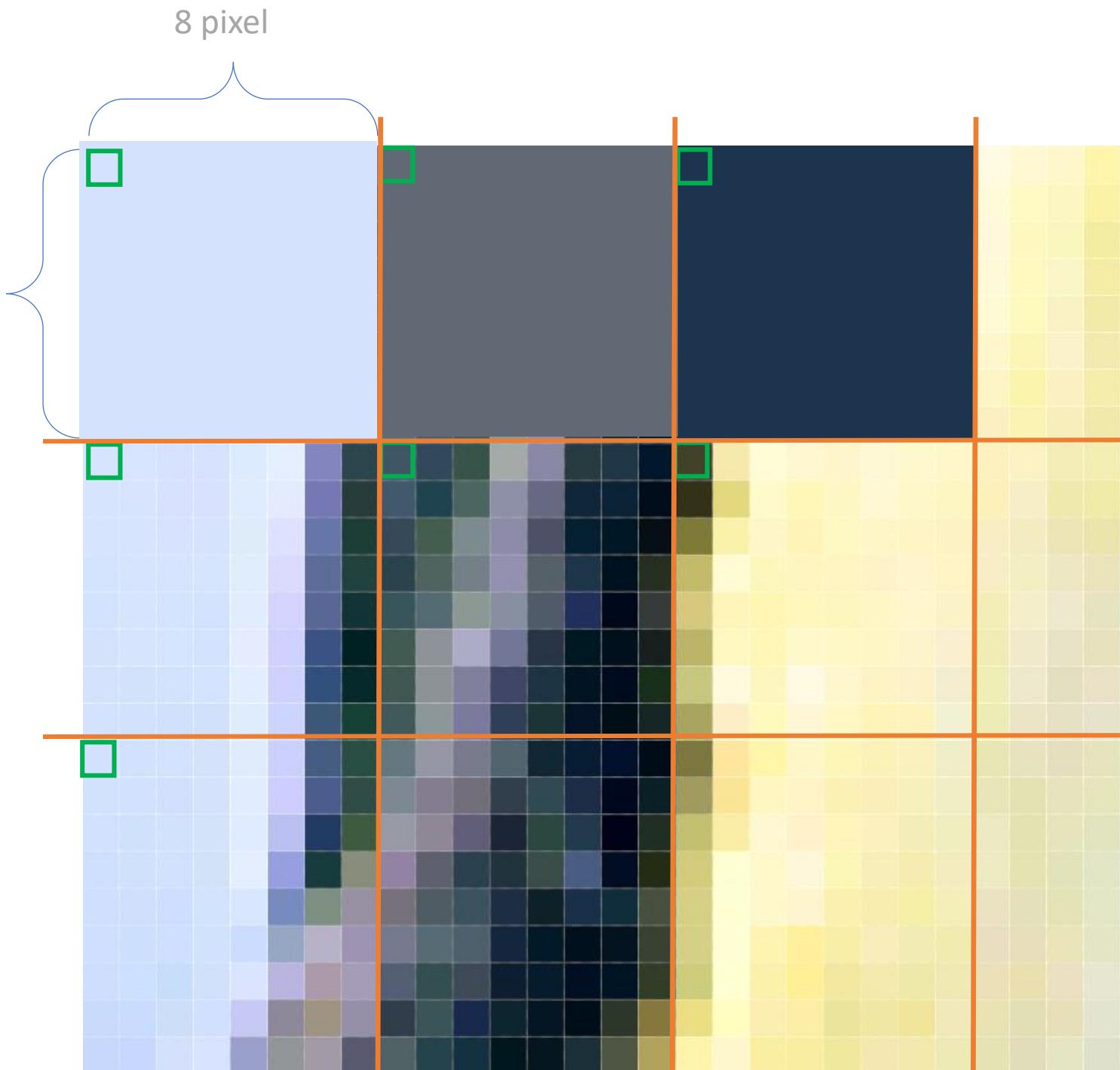
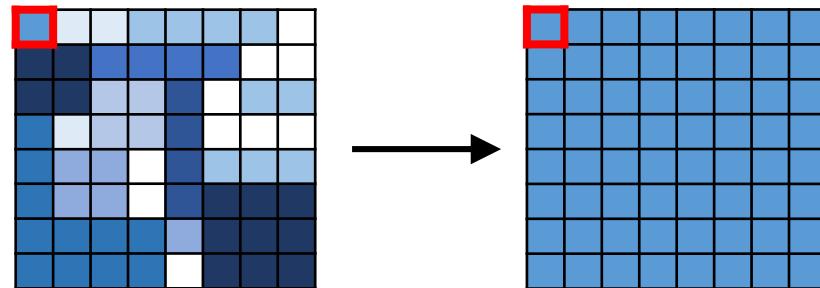
Q4. 馬賽克企鵝

- 每8x8個pixel為一區塊，每一區塊用它最左上角的顏色取代



Q4. 馬賽克企鵝

- 每 8×8 個像素一組
- 取得區塊最左上的顏色
- 用這個顏色取代其它像素



Q5. 灰階企鵝



Penguins.jpg



Q5-1.jpg



Q5-2.jpg

Q5. 灰階企鵝



- 灰階即代表亮度：0～255

- $R = G = B$

- 彩色轉灰階可取平均亮度
 $(R+G+B) \div 3$

- 人類對顏色的敏感度不同

- 綠色 0.587

- 紅色 0.299

- 藍色 0.114

- 標準灰階公式：

$$0.299 * R + 0.587 * G + 0.114 * B$$

將影像用上述兩種方法將每一個pixel轉成它的灰階值

1. $(R, G, B) \rightarrow \left(\frac{R+G+B}{3}, \frac{R+G+B}{3}, \frac{R+G+B}{3}\right)$

2. $(R, G, B) \rightarrow (0.299R+0.587G+0.114B, 0.299R+0.587G+0.114B, 0.299R+0.587G+0.114B)$

Q6. 黑白企鵝



Threshold = 180



Q6-1.jpg

Threshold = 64



Q6-2.jpg

Threshold = 20



Q6-3.jpg

Q6. 黑白企鵝

- 像素非黑即白
- 取平均亮度： $(R+G+B) \div 3$
- 設定門檻值(Threshold)：
平均值大代表較亮，將其顏色設為白色 (255, 255, 255)
平均值小代表較暗，將其顏色設為黑色 (0, 0, 0)
- 門檻愈大代表要夠亮才能設為白色（即黑點會較多）
 - Threshold = 20
 - Threshold = 64
 - Threshold = 180



Threshold = 180



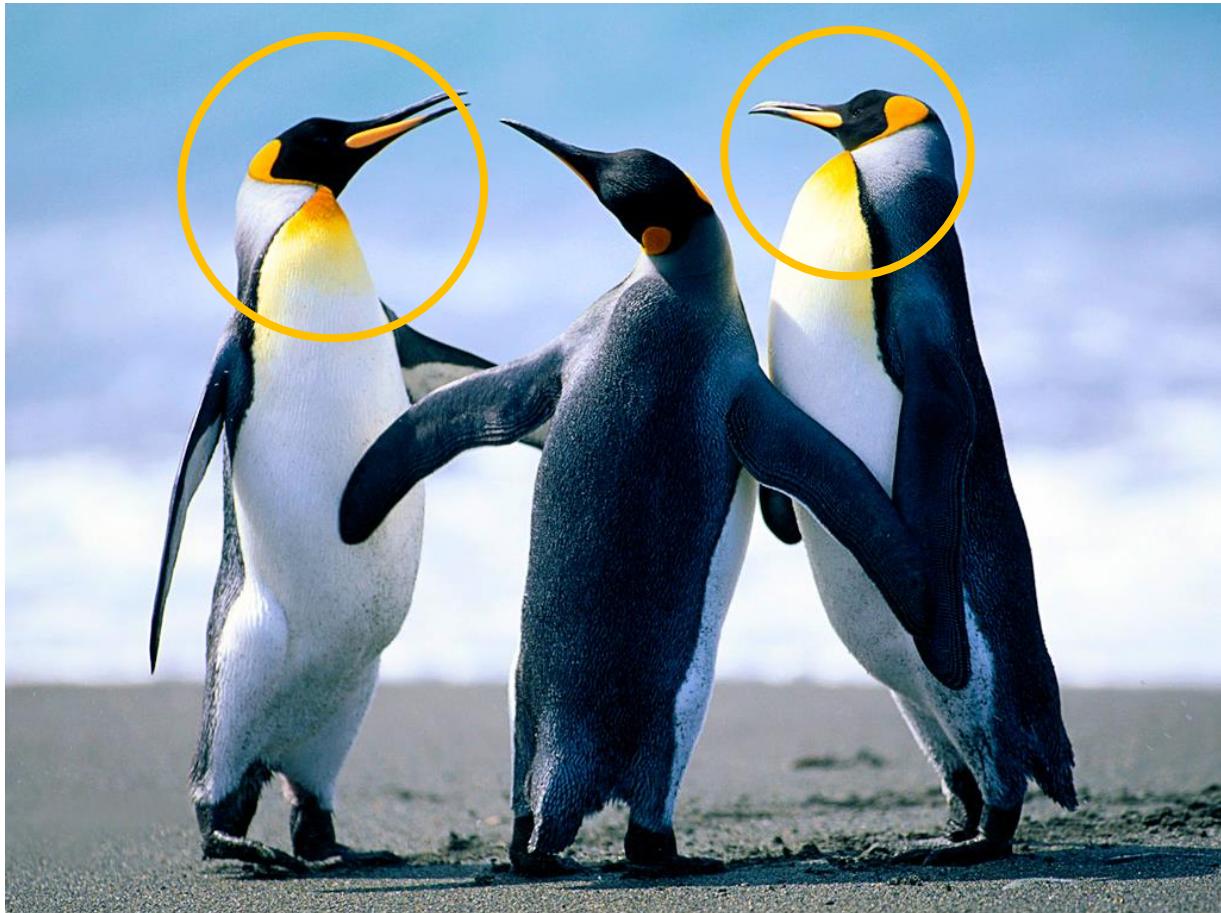
Q7. RGB三色互換企鵝



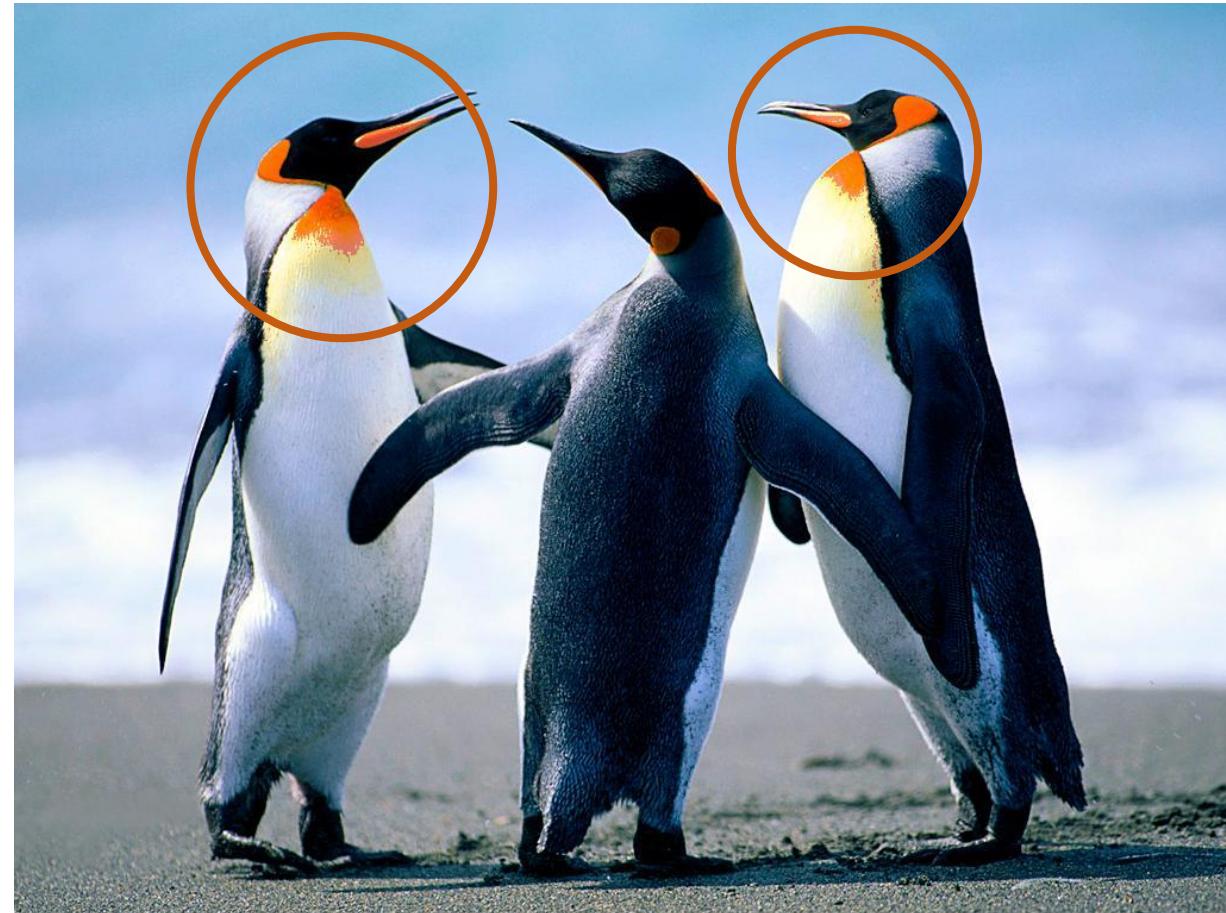
Q7. RGB三色互換企鵝

- 將影像每一Pixel的RGB互換
 - RGB → RBG
 - RGB → GRB
 - RGB → GBR
 - RGB → BRG
 - RGB → BGR
- 組合數： $3 \times 2 \times 1 = 6$
- 扣除原圖：5張圖

Q8. 顏色替換企鵝



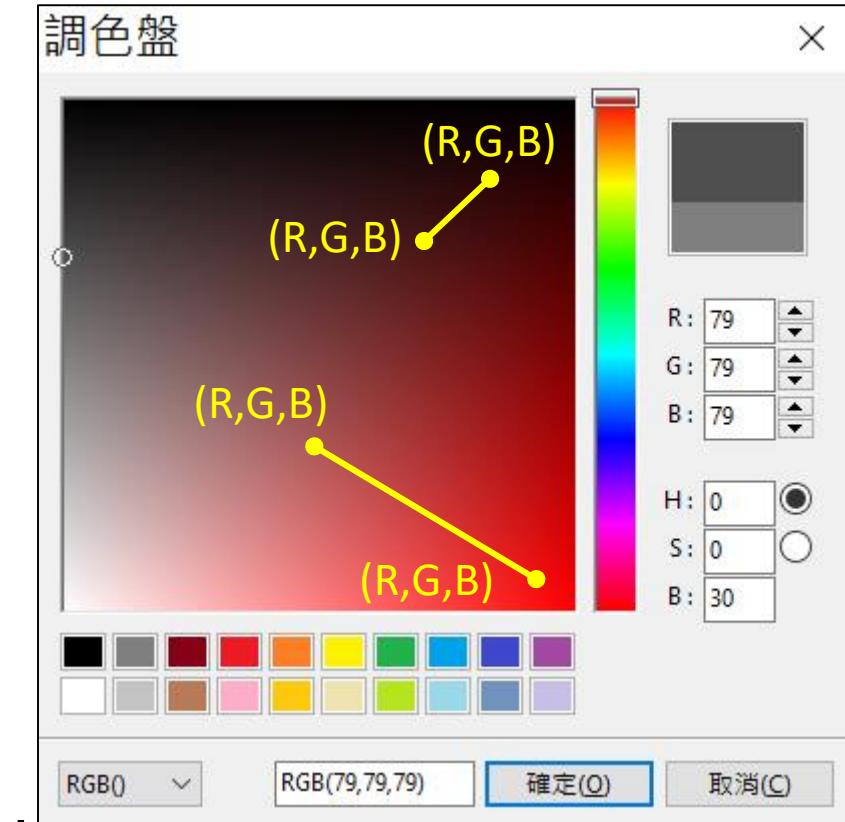
Penguins.jpg



Q8.jpg

Q8. 顏色替換企鵝

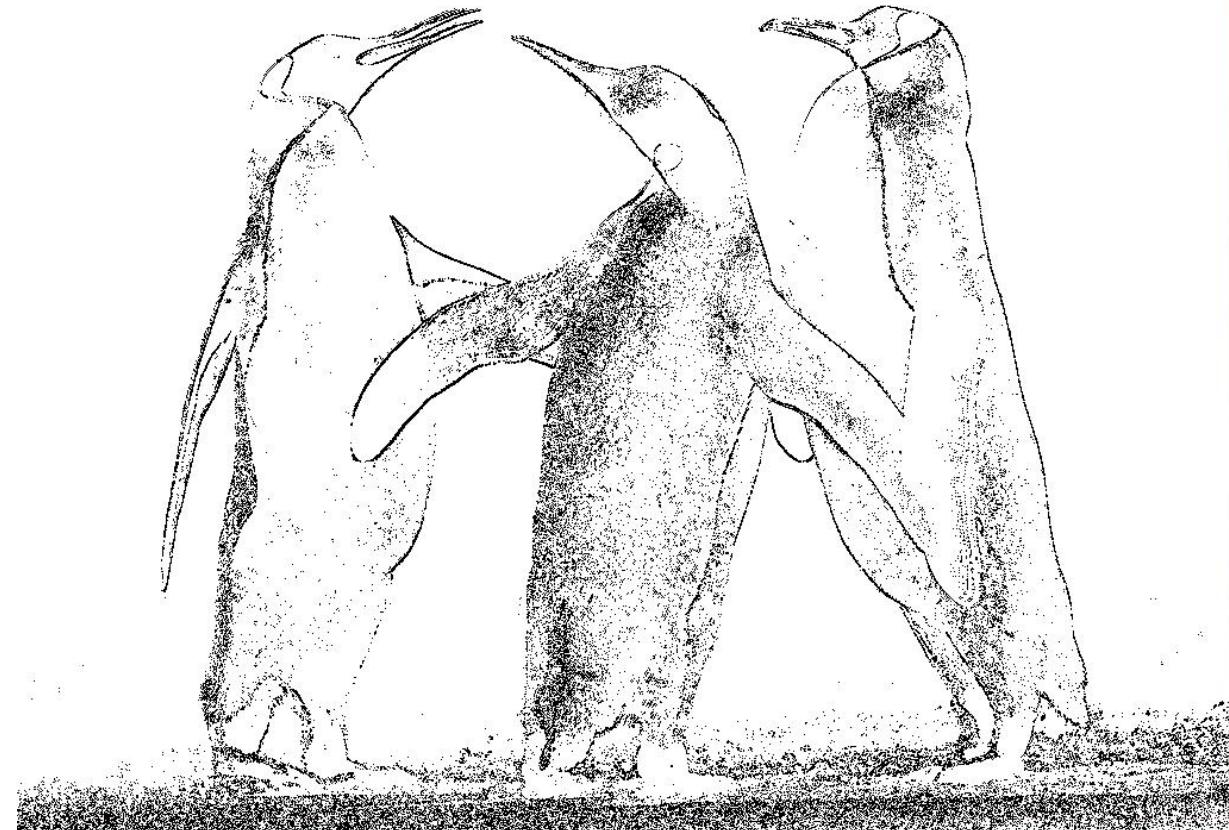
- 不只是替換單一顏色，而是取代相似的顏色
- 將RGB看成空間中的XYZ三維度
- 相似的顏色可以用不同距離來定義
- 此題使用歐氏距離 $\sqrt{(\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2}$
並定義若兩色距離小於100.0，則它們為相似的顏色
- 目標：
將顏色「相似」於 (R:255, G:190, B:25) 的像素，G*0.6



Q9. 企鵝線條描繪



Penguins.jpg

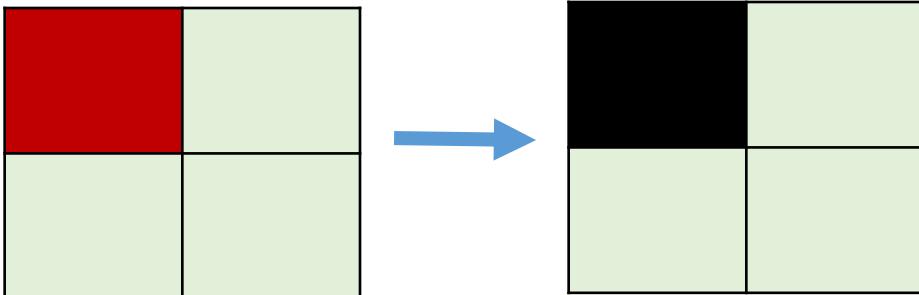


Q9.jpg

Q9. 企鵝線條描繪

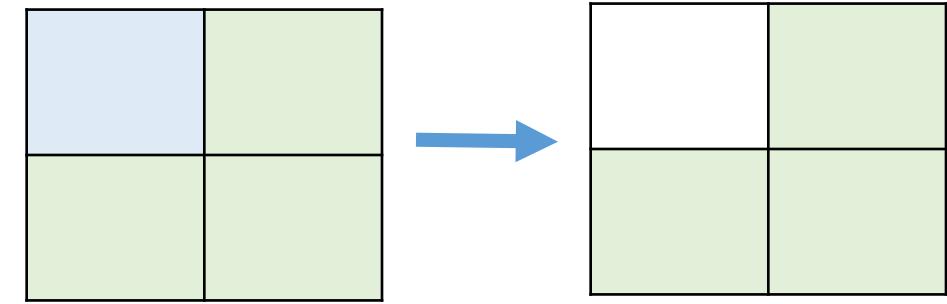
- 若一個的pixel與其鄰近的pixel差異很大，可以推測它可能是輪廓的部分
- 我們定義「pixel」與「下方pixel」和「右方pixel」的 $(R+G+B)/3$ 絕對值差均大於10，則為黑色(邊)，反之則為白色

此紅色色塊與下方和右方差異大，故改為黑色



情形一

此藍色色塊與下方和右方差異小，故改為白色



情形二

Q10. 企鵝中值濾波去雜訊



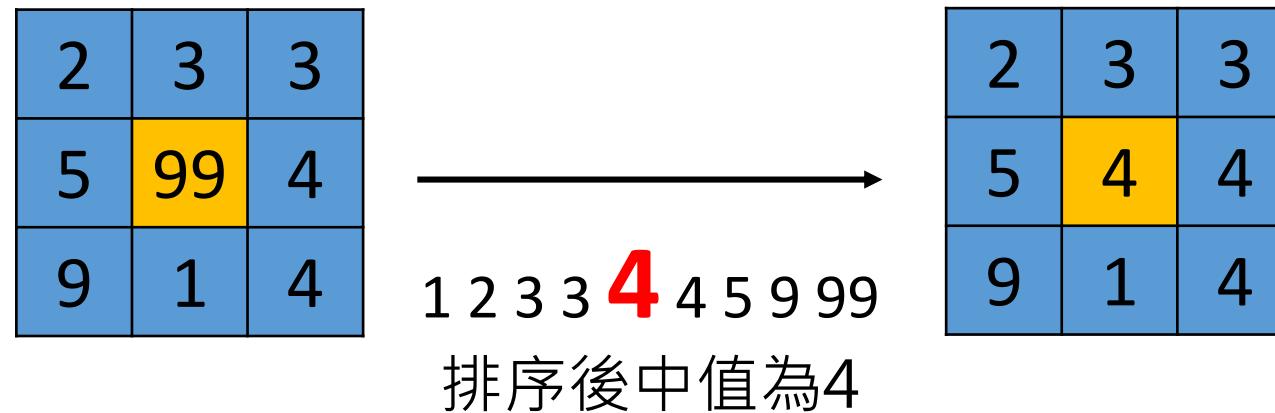
Penguins_noise.jpg



Q10.jpg

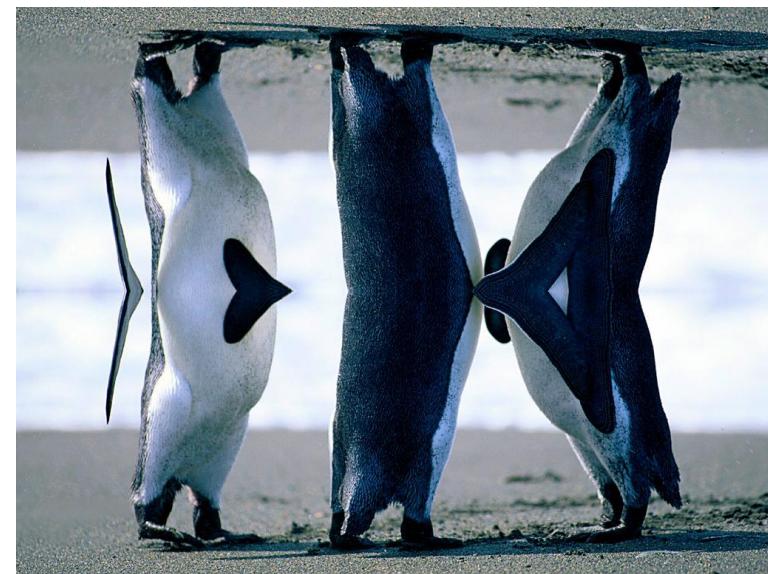
Q10. 企鵝中值濾波去雜訊

- 替每個Pixel取(R+G+B)/3當作判定值
- 針對每個Pixel，取該pixel(黃色)及周圍八個(藍色)Pixel之判定值



- 將其中位數所指之像素所代表的 (R , G , B) 取代原RGB值

Q11. 鏡像企鵝

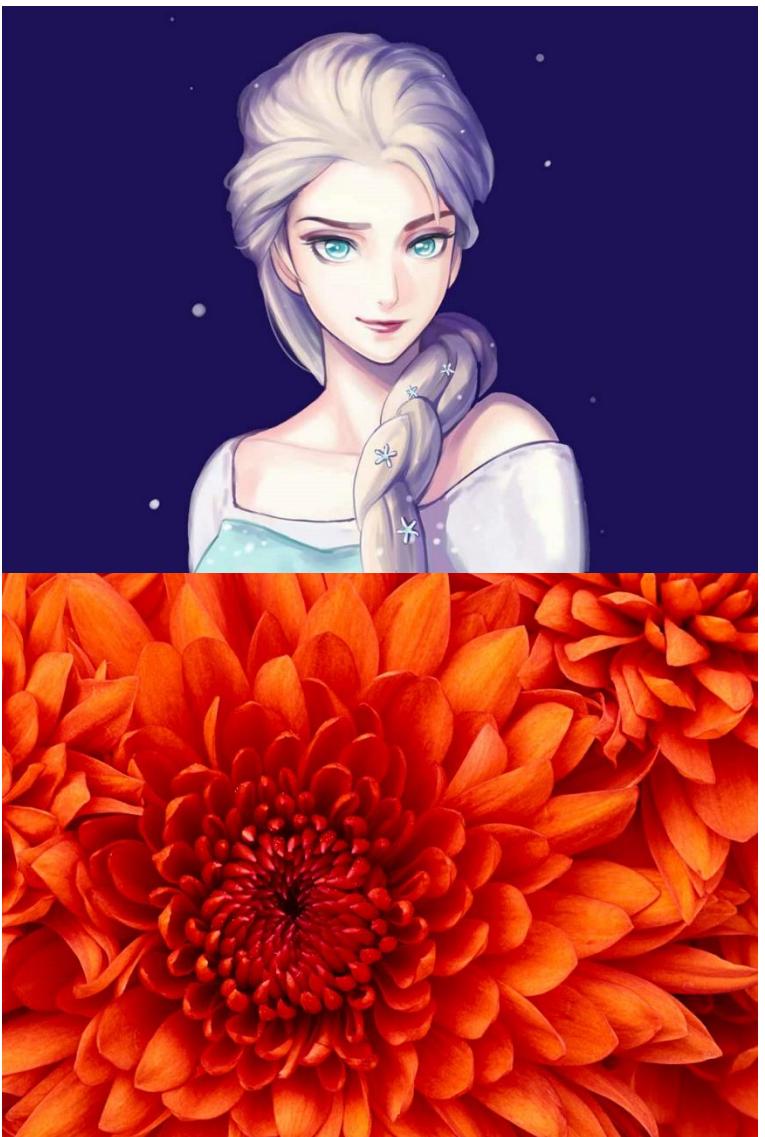


Q11. 鏡像企鵝

- 做出不同方向的鏡像效果
 - 右到左、左到右、上到下、下到上
- 兩種方法
 - 用LOOP讀取像素，並在欲鏡射的位置反序寫入來取代原像素
 - 或利用PIL的函式（crop、paste）進行切割旋轉複製



Q12. Elsa換背景



Q12.jpg

Q12. Elsa換背景

- 將影像原先藍色背景替換為新的背景
- HINT：依序讀取對應的Pixel來取代相似色
 - 毛邊太多或是吃色太多會扣分