

Introduction to Image Processing HW1

408420083 電機三 B 陳昀顯

HW1.1 Image file format conversion

· 實驗目的

這次功課要求將一個.bmp 的 color image 轉成.jpg .gif 的格式，接著用.jpg 轉換成兩種不同的 Quality Factor(本次取 $QF = 100$ 、 $QF = 0$)，完成之後秀出圖片檔及檔案大小，最後執行 PSNR 分析，以.bmp 檔作為原始圖檔和 2.jpg .gif 依序進行比對。

下圖為執行程式碼：

```
p1 = imread('John Wick.bmp');
imwrite(p1,'John Wick.jpg');
[y,map] = rgb2ind(p1,256); imwrite(y,map,'John Wick.gif');
p2 = imread ('John Wick.jpg');
p3 = imread ('John Wick.gif');

%task1
imwrite(p2,'John WickQF100.jpg','Quality',100);
imwrite(p2,'John WickQF0.jpg','Quality',0);
t1 = imread ('John WickQF100.jpg');
t2 = imread ('John WickQF0.jpg');
%task2
imshow(t1),figure,imshow(t2),figure,imshow(p1),figure,imshow(p3); %QF100,QF0,bmp,gif
colormap(map);
%task4
%bmp -> QF100
p1 = double(p1(:,:,1)*0.299)+double(p1(:,:,2)*0.587)+double(p1(:,:,3)*0.114);
t1 = double(t1(:,:,1)*0.299)+double(t1(:,:,2)*0.587)+double(t1(:,:,3)*0.114);
s = 0;
for i = 1:849
    for j =1:1566
        m = (double(p1(i,j))-double(t1(i,j)))^2;
        s = s+m;
    end
end
mse = double(s)/(849*1566);
psnr = 10*log10(255*255/mse);

%bmp -> QF0
t2 = double(t2(:,:,1)*0.299)+double(t2(:,:,2)*0.587)+double(t2(:,:,3)*0.114);
s1 = 0;
for i = 1:849
    for j =1:1566
        m = (double(p1(i,j))-double(t2(i,j)))^2;
        s1 = s1+m;
    end
end
mse1 = double(s1)/(849*1566);
psnr1 = 10*log10(255*255/mse1);

%bmp -> gif
s2 = 0;
for i = 1:849
    for j =1:1566
        m = (double(p1(i,j))-double(p3(i,j)))^2;
        s2 = s2+m;
    end
end
mse2 = double(s2)/(849*1566);
psnr2 = 10*log10(255*255/mse2);
```

圖 1、HW1.1 之 MATLAB 程式碼

· 程式碼架構

由於 HW1.1 要做四個 Task，故將程式碼分成了四個區塊來執行

1. 負責轉檔案格式

- 用 `imwrite` 將.bmp 轉成.jpg
- 用 `rgb2ind` 將.bmp 轉成.gif，條件加上 256 種顏色
- 用 `imwrite(x, 'lena_QF100.jpg', 'Quality', 100)` 改變.jpg 的 QF 值

2. 秀出四張圖片

3. 前往 Matlab Drive 觀測檔案實際大小

4. 依序執行 bmp -> QF100、bmp -> QF0、bmp -> gif 的 PSNR 計算

· PSNR 簡述&算式設計

一、PSNR 定義

影像經過處理或壓縮之後，輸出的影像都會有某種程度與原始影像不一樣。為了衡量經過處理或壓縮後的影像品質，我們會參考 PSNR 值（峰值信號雜訊比）來認定某個處理程序是否令人滿意。

PSNR 最容易經由均方差（Mean Squared Error，MSE）來定義，假設給一個無雜訊的 $m \times n$ 單色的影像 I 和它有著雜訊的影像 K ，均方差和 PSNR 的定義如下所示：

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$$

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_i^2}{MSE} \right)$$

上式的 mn 代表照片的影像長度 x 寬度 x 通道數(其中 Grayscale image 為 1，Color image 為 3)， MAX_i 代表影像中 pixel 所能表示的最大值，當一張照片的 pixel 值為 unit8 的格式 (0~255) 其最大值就是 255。對於彩色影像的 pixel 有 RGB 三個值，也能適用於 PSNR 的計算，但在計算 MSE 時要除以 3。

PSNR 值越高表示品質越好，一般而言當 PSNR 的值 < 30db 時，代表以人眼來看是不能容忍的差異，因此我們會希望影像 PSNR 值 > 30db。然而 PSNR 值較高，並不代表影像品質一定較好，有時候還是必須依靠人眼來輔助判斷影像的品質才較為正確。

二、PSNR 算式 IN MATLAB

考慮到.bmp .jpg 為 color image 形式，其中.gif 為特殊案例，它只能允許 256 種顏色在一張相片內，同時也不支援 Grayscale、Binary 的檔案格式操作，且內容屬性為 uint8 的 8bits 二進位表示法，從 Matlab 上看為十進制轉換期原因是協助我們方便閱讀資料。

以轉.bmp 為例：

```
p1 = double(p1(:,:,1)*0.299)+double(p1(:,:,2)*0.587)+double(p1(:,:,3)*0.114);
```

右式 p1 為原.bmp 檔，將 R、G、B、三個通道分別乘以相對應的係數後，p1 從原先的三維陣列變成二維陣列，其他兩個.jpg 如法炮製，.gif 不用轉的原因是它本身就是二維陣列單一通道。

接著計算 MSE，首先要創建雙層 for 迴圈來取出二維陣列內的每個 pixel 點，由.bmp 相減 QF100、QF0 的.jpg 圖檔和.gif 檔：

```
for i = 1:2160
    for j =1:3840
        m = (double(p1(i,j))-double(t2(i,j)))^2;
        s1 = s1+m;
    end
end
```

最後帶入 PSNR 公式即可得出不同格式下的數值大小。

• 成果展示(原檔為 David Guetta.bmp、John Wick.bmp)



圖 2、David Guetta.bmp



圖 3 、 David Guetta.jpg



圖 4 、 David GuettaQF0.jpg



圖 5 、 David GuettaQF100.jpg



圖 6、David Guetta.gif

 David_guetta.bmp	346 KB
 David_guetta.gif	4.42 MB
 David_guetta.jpg	543 KB
 David_guettaQF0.jpg	135 KB
 David_guettaQF100.jpg	1.83 MB

圖 7、轉換格式後之檔案大小

 psnr	48.5090	1×1	double
 psnr1	27.1880	1×1	double
 psnr2	8.7278	1×1	double

圖 8、以.bmp 為參考點和 2jpg、gif 比較 psnr
(psnr 為 QF100.jpg、psnr1 為 QF0.jpg、psnr2 為.gif)



圖 9、John Wick.bmp



圖 10、John Wick.jpg



圖 11、John WickQF0.jpg



圖 12、John WickQF100.jpg



圖 13、John WickQF100.gif






 John Wick.bmp	5.07 MB
 John Wick.gif	713 KB
 John Wick.jpg	144 KB
 John WickQF0.jpg	25 KB
 John WickQF100.jpg	497 KB

圖 14、轉換格式後之檔案大小




 psnr	36.7374	1×1	double
 psnr1	25.2672	1×1	double
 psnr2	8.6520	1×1	double

圖 15、以.bmp 為參考點和 2jpg、gif 比較 psnr
(psnr 為 QF100.jpg、psnr1 為 QF0.jpg、psnr2 為.gif)

HW1.2 Binary to Gray code

· 實驗目的

這次功課要求執行 binary code & Gray code bit plane decomposition，將指定圖片匯入後要先轉成灰階影像(如果照片本身就是灰階則不用轉)，利用 binary code to gray code 遞迴式把 8bits(binary)表示的 Grayscale image 改成用 gray code 表示，最後秀出兩種情況下的 bit plane 並進行比較。

下圖為執行程式碼：

```
p = imread('DG.bmp');
p = rgb2gray(p);

imshow(p);

posx = [0 1 2 0 1 2 0 1]/3;
posy = [2 2 2 1 1 1 0 0]/3;
figure
for i = 1:8
    subplot("position",[posx(i),posy(i),0.3,0.3]);
    imshow(logical(bitget(p,i)));
end

figure
for i = 1:7
    subplot("position",[posx(i),posy(i),0.3,0.3]);
    imshow(logical(xor(bitget(p,i),bitget(p,i+1))));
end
subplot("position",[posx(8),posy(8),0.3,0.3]);
imshow(logical(bitget(p,8)));
```

圖 16、HW1.2 之 MATLAB 程式碼

· 程式碼架構

上述程式分成兩個部分：

1. 用 bitget 取出 binary code 進行 bit plane decomposition
2. 用 bitget 取出 binary code 先進行轉 gray code 迴圈，再秀出 bit plane decomposition

· 成果展示(原檔為 David Guetta.bmp、John Wick.bmp)



圖 17、David Guetta (grayscale version)

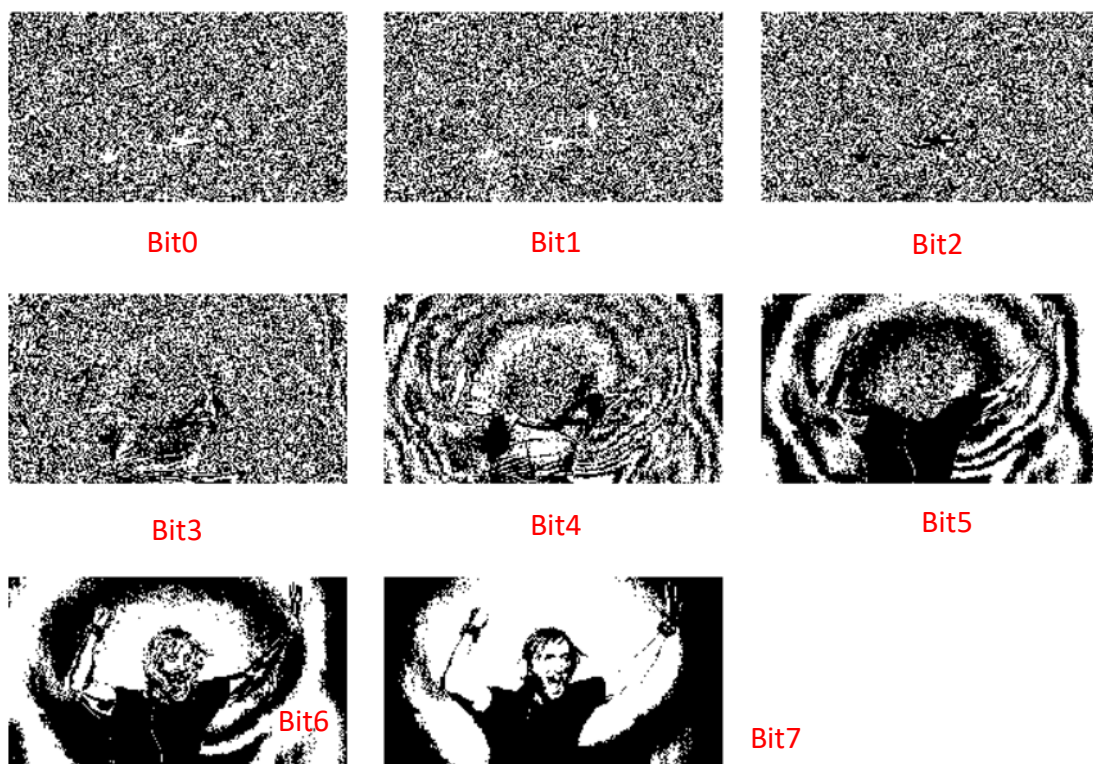


圖 18、David Guetta (binarycode bit plane decomposition)

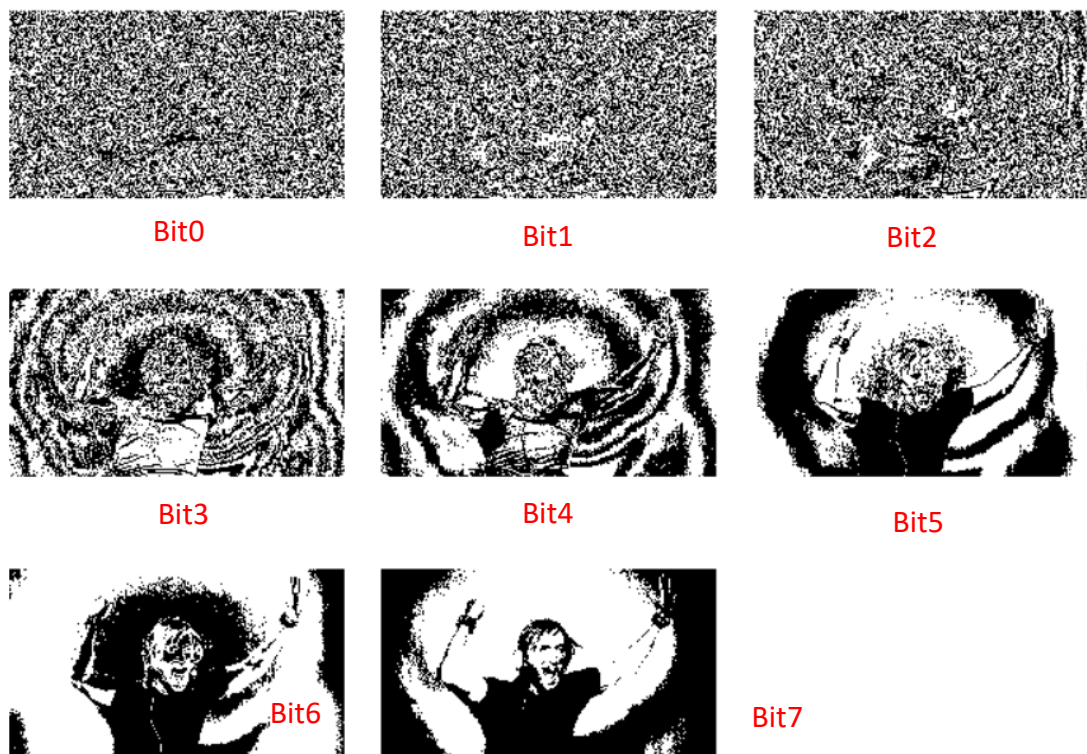


圖 19、David Guetta (graycode bit plane decomposition)



圖 20、John Wick (grayscale version)

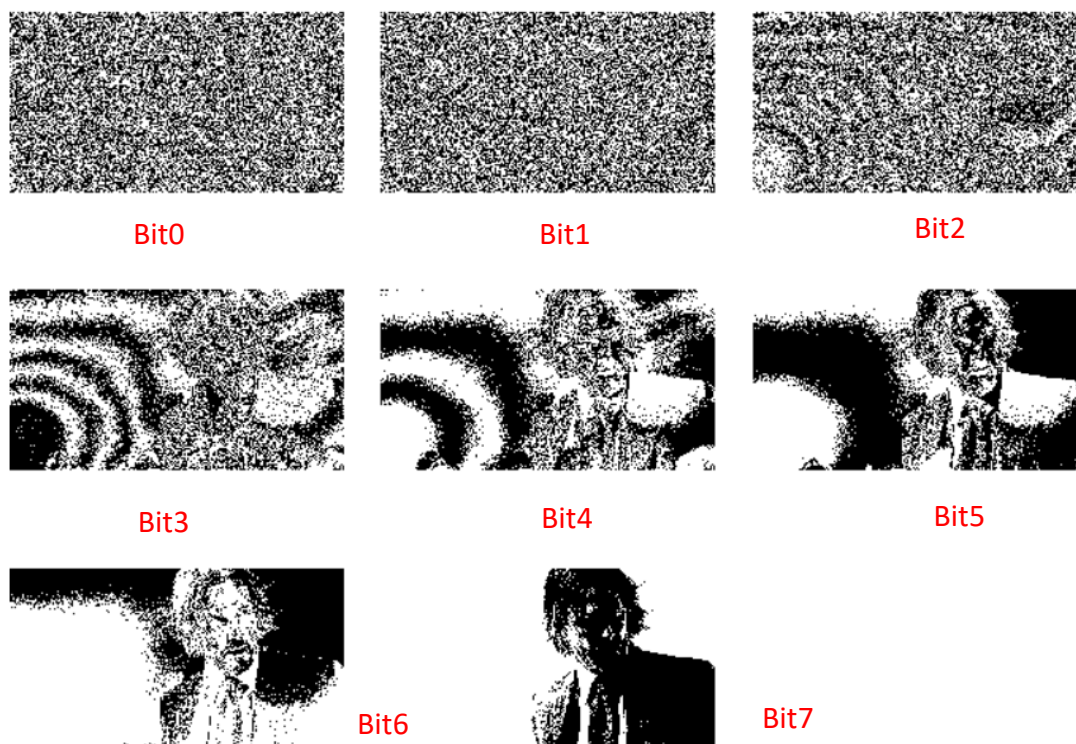


圖 21、John Wick (binarycode bit plane decomposition)

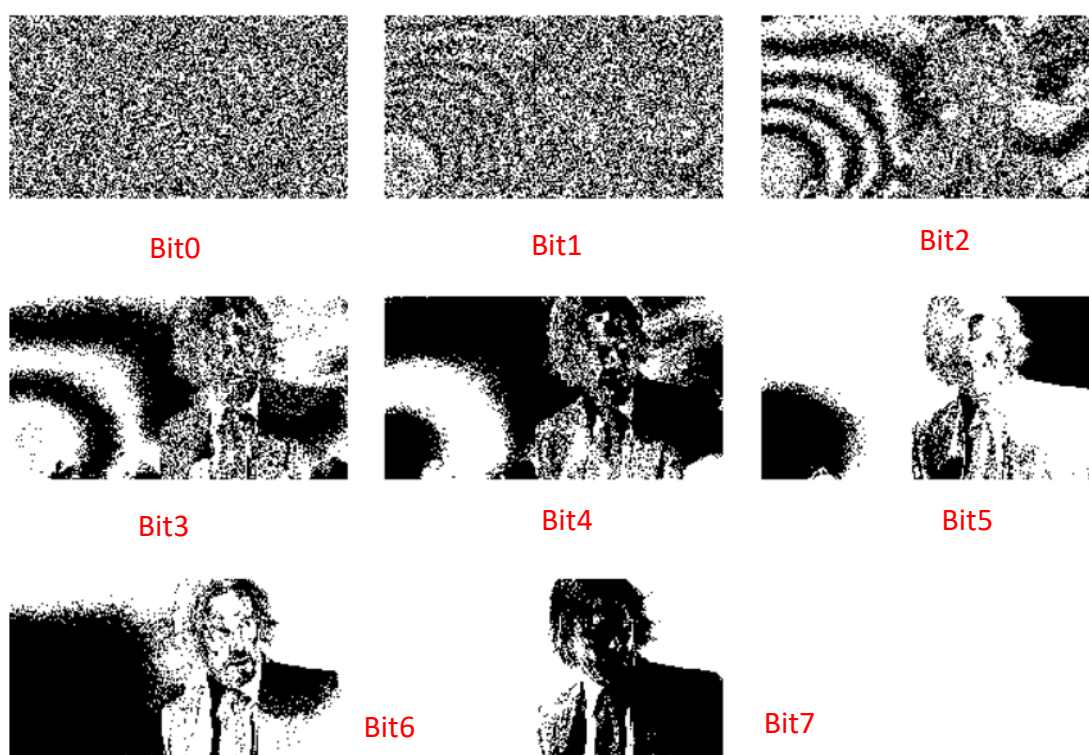


圖 22、John Wick (graycode bit plane decomposition)

・討論 binary gray 的 bit plane 差異

從上圖中可以發現除了 bit7 最高位元外，其他 bit 有著明顯的互補色差異，以 bit6 來看更能感覺到差異，此外 gray code 的特點為相鄰兩數間只會差一個 bit，其 bit plane 圖案變化程度來的比 binary code 還要大。

· 本次作業心得

這次作業 1.1 想的比較快，主要都是上課之前提到過的東西，像是讀取圖檔跟轉圖檔格式，只需專注在 PSNR 的計算上面；1.2 是我卡蠻久的一題，而且一開始方向就已經錯了，原本以為是要取出每個 pixel 的 binary code 轉成 gray code，其實用 bit plane 整個矩陣的方式來進行遞迴會更有效率。最後再感謝李助教的指點讓我完成這次作業。