

adaptive gamma correction (AGC)

目錄

adaptive gamma correction (AGC)	1
1、 為什麼要做 image correction	2
2、 gamma correction 的基本原理	2
3、 使用 gamma correction 所遇到的問題	2
4、 ChatGPT 所提供的解決方法	2
5、 為什麼使用 AGC	3
A. AGC 的分類	3
B. AGC 的計算流程	4
C. AGC 對於低對比度的處理	5
D. AGC 對於高對比度的處理	5
6、 EXP0.bmp 和 EXP4.bmp，執行 AGC 的比較	6
A. Origin vs AGC	6
B. 與一般 GC 的比較	9
7、 Experience	12
8、 Feedback of using AI guidance	12
9、 Reference	13

1、 為什麼要做 image correction

影像會有過曝或是過暗的狀況，導致影像不清楚，此時使用 image correction 的技術，來改善影像明暗不均衡的狀況。

2、 gamma correction 的基本原理

gamma correction 可以調整圖像的亮度和對比度，因為人眼對亮度的感知是非線性的，可以符合人類視覺系統的特性。

gamma correction 的公式如下：

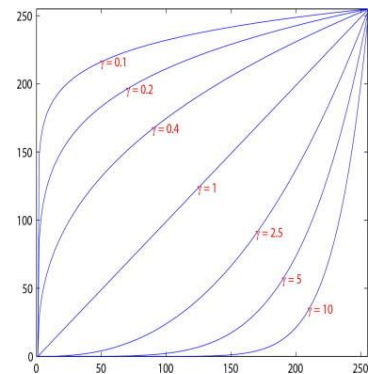
$$I_{out} = I_{in}^{\gamma}$$

I_{out} 是輸出的圖片， I_{in} 是輸入的圖片。

當 $\gamma = 1$ 時，表示線性關係，不進行校正。

當 $\gamma < 1$ 時，圖像變亮並減少對比度，因為在暗區的點會移至較亮的區域。

當 $\gamma > 1$ 時，圖像會變暗並增強對比度，因為在亮區的點會移至較暗的區域。



3、 使用 gamma correction 所遇到的問題

- a、 信息損失：如果 γ 設置得太高，可能會導致圖像陰影中的細節丟失，如果設置得太低，可能會導致亮部細節丟失。
- b、 圖像異常： 不正確或極端的 γ 值可能會引發圖像異常和視覺失真，例如色彩的拼接（顏色的條紋）或亮度值失真。
- c、 主觀性： 選擇正確的 γ 值通常是一個主觀的過程。不同應用和顯示設備可能需要不同的 γ 值以獲得最佳效果。
- d、 兼容性： γ 值的不一致可能導致不同設備之間的顏色變化和不一致。
- e、 範圍有限： 主要處理亮度校正。不處理與顏色校正、色彩空間或其他圖像處理任務相關的問題。
- f、 非線性特性： 伽馬校正引入的非線性特性可能使在經過校正的圖像上執行精確的數學運算變得困難。

4、 ChatGPT 所提供的解決方法

- a、 高級色調映射： GC 可能不夠的情況下使用高級色調映射技術，如 Reinhard、Mantiuk 或 Durand 操作符，以更好地控制色調映射過程。

GC=gamma correction, AGC= adaptive gamma correction

- b、局部對比度增強：實施局部對比度增強方法，如自適應直方圖均衡化（AHE）或有限對比度自適應直方圖均衡化（CLAHE）。這些技術增強圖像的局部區域對比度，防止全局 GC 可能的噪聲和異常放大。
- c、直方圖規格化：使用直方圖規格化或匹配技術來將圖像的直方圖匹配到所需的參考直方圖。這允許更精確地控制圖像的外觀。
- d、線性工作流程：線性工作流程涉及在最終輸出階段之前使用線性色彩空間（伽馬 1.0）工作。這種方法可以減少異常和保持更好的顏色準確性。
- e、用戶交互：在某些情況下，允許用戶手動調整 γ 值以獲得更好的結果。
- f、色彩管理：如果顏色準確性是一個問題，實施色彩管理系統並使用色彩配置文件可以確保不同設備和軟件之間的一致顏色表示。
- g、基於機器學習的方法：機器學習技術，如神經網絡，可以用於增強圖像，考慮圖像增強的複雜性。

5、為什麼使用 AGC

ChatGPT 並沒有提供這個解法，而 bingAI 提供了此解法: adaptive gamma correction(AGC)。使用 gamma correction 時，要手動設定 γ 的數值，在不同圖片時都要重新設定合適的 γ 值，並且必須不斷調整數值，來符合不同圖片的狀況。而 AGC 可以藉由讀取圖片的強度來自行調整 γ 值，做到自動適應不同圖片的狀況。因為調整圖片的強度只會影響到明亮變化，不會影響到顏色的顯示，所以藉由調整強度來改善過暗或過亮的問題。以下將介紹 AGC 的計算與想法。

A. AGC 的分類

AGC 的公式如下：

$$I_{out} = cI_{in}^{\gamma}$$

I_{out} 是輸出圖片的強度， I_{in} 是輸入圖片的強度。

當要進行自動產生 γ 值時，需要先將圖片分類，才能產生出相對適合的 γ 值，從查到的資料中，第一步，先分類對比度，當圖片的強度標準差 σ 的 4 倍小於所設定的對比度參數(資料中提預設為 1/3)，代表圖片中的強度較為集中，對比度較低，反之對比度較高。所以會根據不同的對比度產生不同的 γ 值。

C 常數可以改善圖像的強度，一般情況下也是手動設置，但是一樣用圖像的強度來設定：

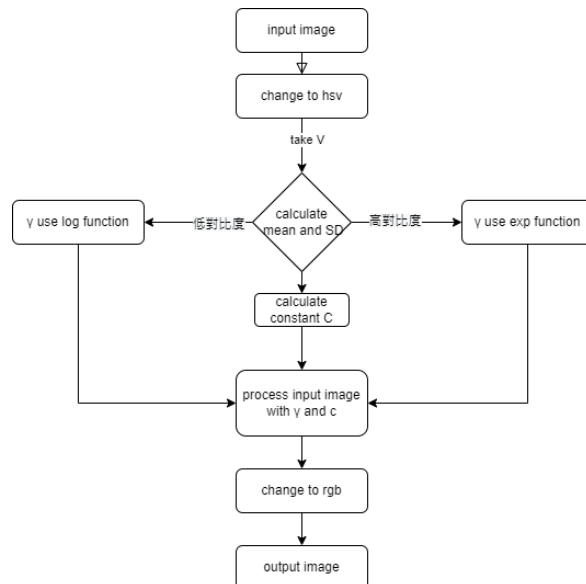
$$c = \frac{1}{1 + u(0.5 - \mu) * [k - 1]}, \quad k = I_{in}^{\gamma} + (1 - I_{in}^{\gamma}) * \mu^{\gamma}, \quad u(x) = \begin{cases} 0, & x \leq 0 \\ 1, & x > 0 \end{cases}$$

這個常數設定適用在高對比度和低對比度中，並且計算出來的 c 是一個矩陣(因為 I_{in} 是一個矩陣)，而 γ 將會有所差異。

GC=gamma correction, AGC= adaptive gamma correction

B. AGC 的計算流程

下圖為 AGC 的判斷流程:



```
inputImage = im2double(rgb2hsv(imread(picturename)));%read image and changed to hsv
outputImage=gammatransform(inputImage);
```

上面程式碼將讀入的圖片轉成 hsv 檔案，並傳入 gamma transform function。

```
tau=3.0;
D=desiredMean+2*desiredStdDev-(desiredMean-2*desiredStdDev);
if D<=(1/tau)
    gamma=-log2(desiredStdDev);
else
    gamma=exp(1-(desiredMean+desiredStdDev)/2);
end
```

上面的程式碼檢查圖片是否為高或低對比度，來設定 γ 值， γ 值的設定會在下面介紹。

```
Value=inputImage(:,:,3);
```

Value 是儲存 V(強度)的資料。

```
k= power(Value,gamma)+(1-power(Value,gamma)).*power(desiredMean,gamma);
if (0.5-desiredMean)<=0
    c=double(zeros(row,column));
else
    c=double(ones(row,column));
end
c=power(double((1+(c).*(k-1))),-1);
```

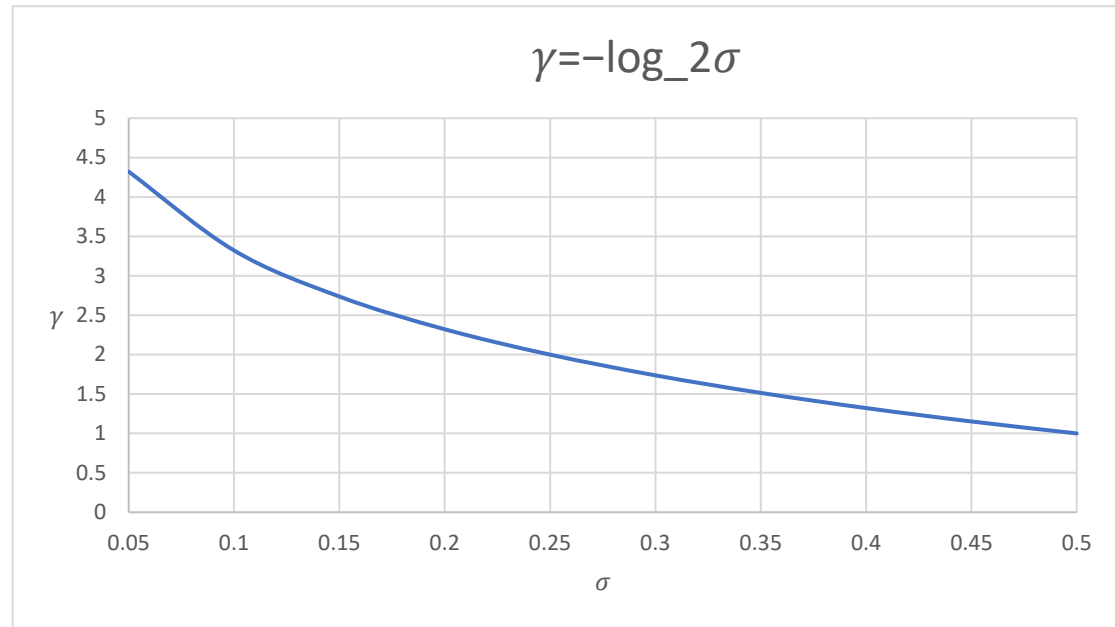
而上面的程式碼根據公式計算出 c 的值。

GC=gamma correction, AGC= adaptive gamma correction

C. AGC 對於低對比度的處理

當對比度低時，已下方公式計算 γ 值，對比度低時代表 σ 較小，代表當對比度越低 γ 越大，可將對比度拉高。

$$\gamma = -\log_2 \sigma$$



D. AGC 對於高對比度的處理

對於高對比的影像需要對亮度調整，並非繼續提高對比度，所以用下方公式設計 γ 值：

$$\gamma = e^{[1 - \frac{\mu + \sigma}{2}]}$$

$(\mu + \sigma)$ 的最大值會是 $\mu = 0.5$ 時， $\sigma = \frac{1}{\sqrt{2}}$ ，推算出 $\gamma = 0.902$ ，所以 γ 值不會低於0.9，而最大不會高於 $e = 2.718$ ，所以 γ 不會有太大的變化，對於對比度的調整不會太大。

6、 EXP0.bmp 和 EXP4.bmp，執行 AGC 的比較

A. Origin vs AGC

EXP0.bmp

Name	entropy	brisque	niqe
origin	5.661374	25.485214	2.858891
AGC	6.327986	18.151377	3.093332



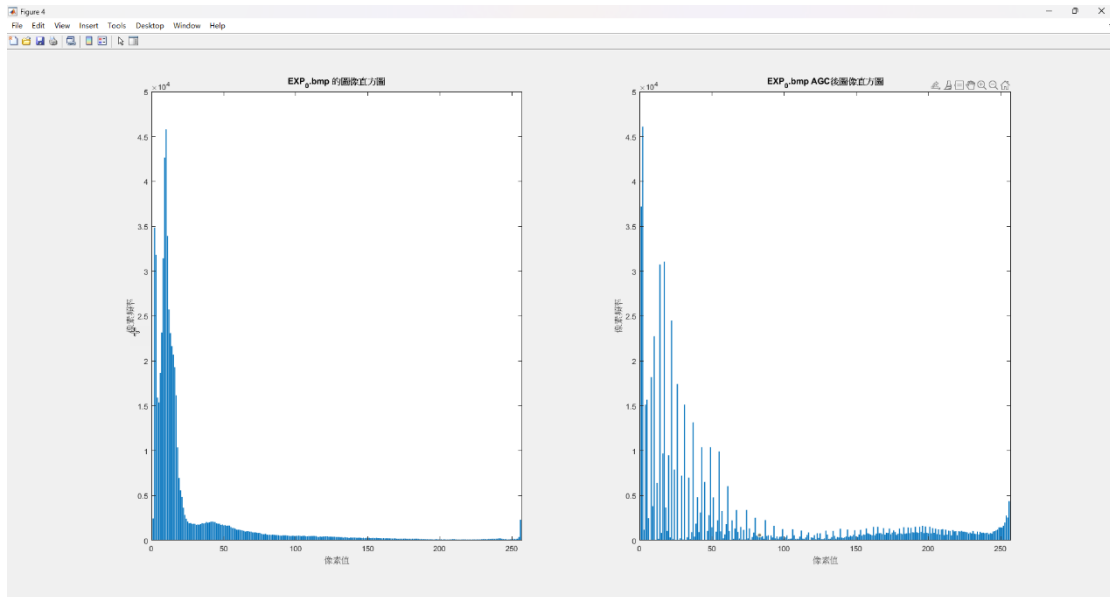
左邊是原始的圖像，右邊是經過 AGC 後的圖像，從主觀角度，會發現經過 AGC 的圖像細節更多，entropy 和 brisque 都顯示 AGC 後的圖像更好。原始圖像的

$\mu = 0.110047$ 、 $\sigma = 0.176341$ ， $4\sigma = 0.705363$ 會發現圖像比 $\frac{1}{3} = 0.333333$ 還要大，所以屬於對比度高的圖片，算出來的 $\gamma = 2.355626 < e=2.718$ 符合假設，其中 c 是一個經過轉換後強度變化的矩陣。

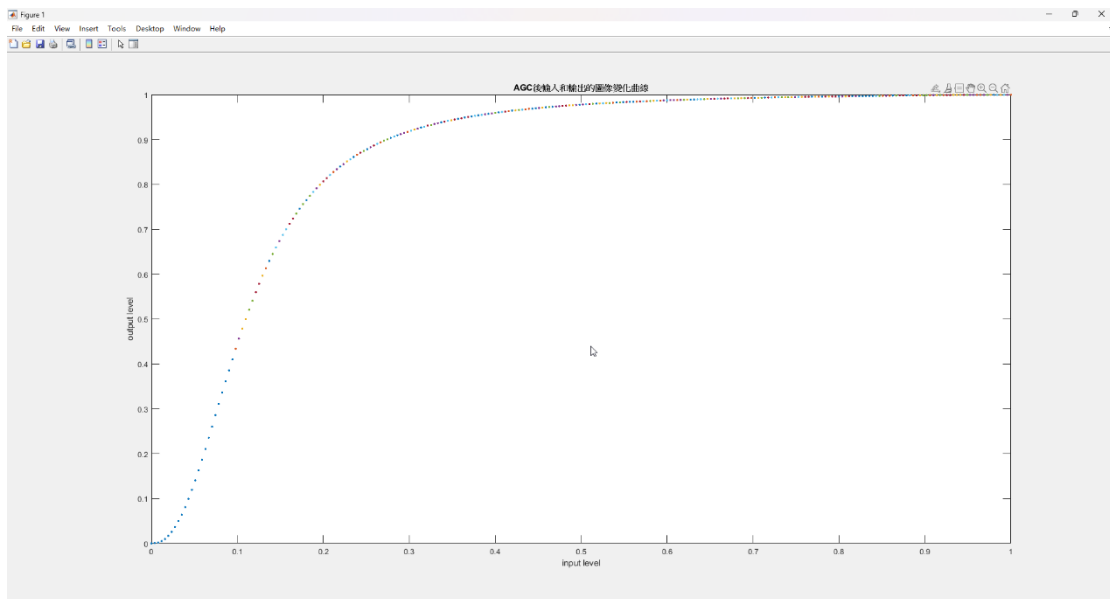
$$Output = c * Input^{2.355626}$$

```
outputImage=inputImage;
outputImage(:,:,3) =double( c .* double(imadjust(inputImage(:,:,3), [], [], gamma)) );
```

上面的程式碼，先將 input 的 H(色调)和 S(饱和度)複製到 output 裡面，再根據 AGC 的公式計算出 output 的 V 值。



上圖為 origin 和 AGC 後的直方圖，可以看出經過 AGC 後有分散了較黑的部分，並增加亮區的比例。



上圖為 output level and input level 的曲線圖，可以看出輸入的點會對應到較亮的區域，輸出的圖片會有較亮的表現，當在高對比度時，根據 c 的公式，輸入的 μ 越低 c 會越大，使得曲線更為陡峭，使暗區的點會有更多移至亮區。

EXP4.bmp

Name	entropy	brisque	nique
origin	7.455342	32.750829	2.621499
AGC	7.276748	34.576359	2.814619

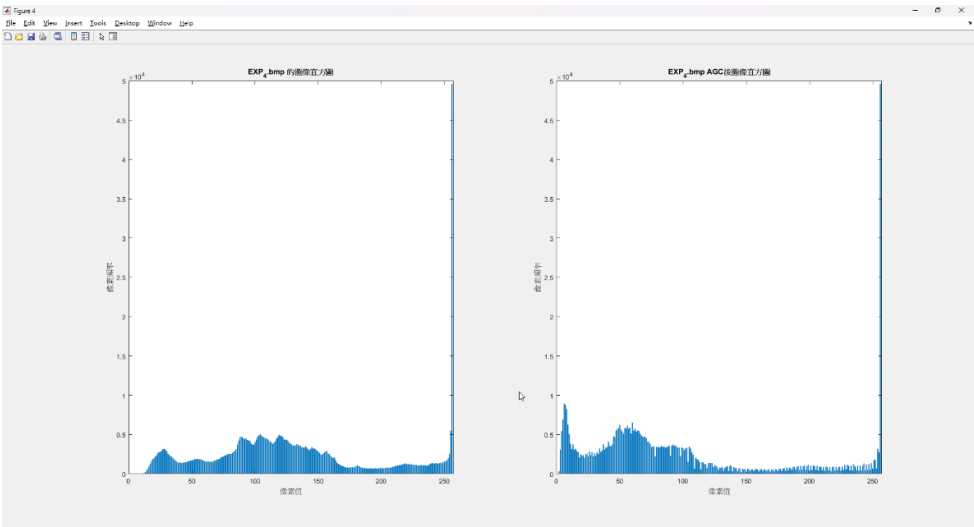


上圖，左邊是原始的圖像，右邊是經過 AGC 後的圖像，從主觀角度，會發現經過 AGC 的圖像有減少過曝的問題，但是在窗外的細節部分，並沒有變好，在評分制中也沒有較好的表現。原始圖像的 $\mu = 0.562095$ 、 $\sigma = 0.252853$ ，

$4\sigma = 1.011410$ 會發現圖像比 $\frac{1}{3}$ 還要大，所以屬於對比度高的圖片，算出來的 $\gamma = 1.808551 < 2.718$ 符合假設，其中 c 是一個經過轉換後強度變化的矩陣。

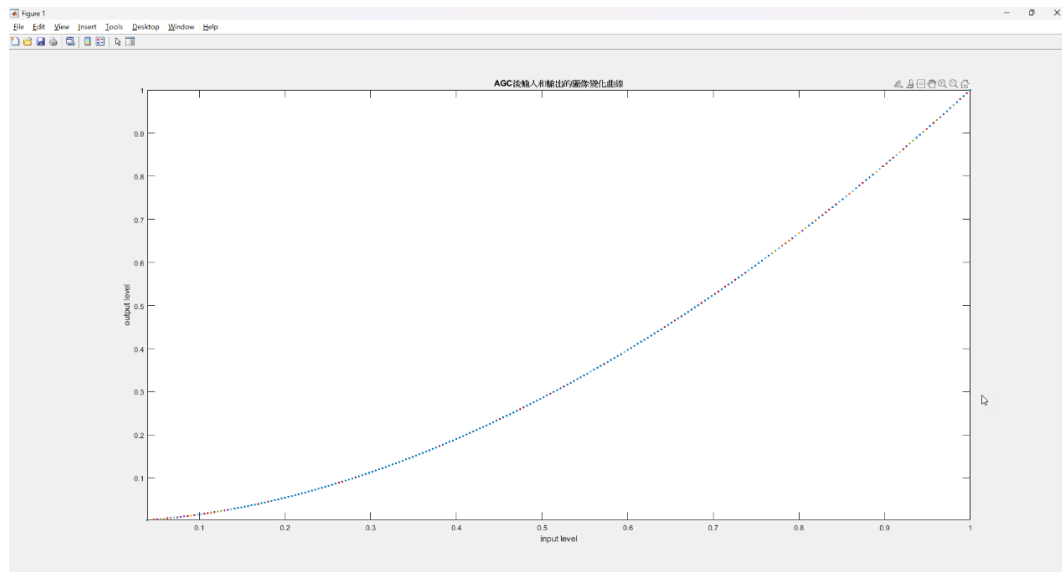
$$Output = c * Input^{1.808551}$$

與之前 EXP0.bmp 的計算方式一樣，先複製 H 和 S ，再將計算出來的 V 存入 $output$ 。



上圖為 $origin$ 和 AGC 後的直方圖，可以看出經過 AGC 後有將部分亮區的點分散到暗區，整體並沒有被改變很多，依然保留了較高的對比度。

GC=gamma correction, AGC= adaptive gamma correction



上圖為 output level and input level 的曲線圖，可以看出輸入會對應到較暗的區域，不過曲線變化不明顯，圖片雖然過曝，但沒有太大的強度差異，所以曲線較為平滑。

B. 與一般 GC 的比較

EXP0.bmp ($\gamma = 0.5$)

Name	entropy	brisque	niqe
origin	5.661374	25.485214	2.858891
AGC	6.327986	18.151377	3.093332
GC	5.611756	27.073485	2.856179

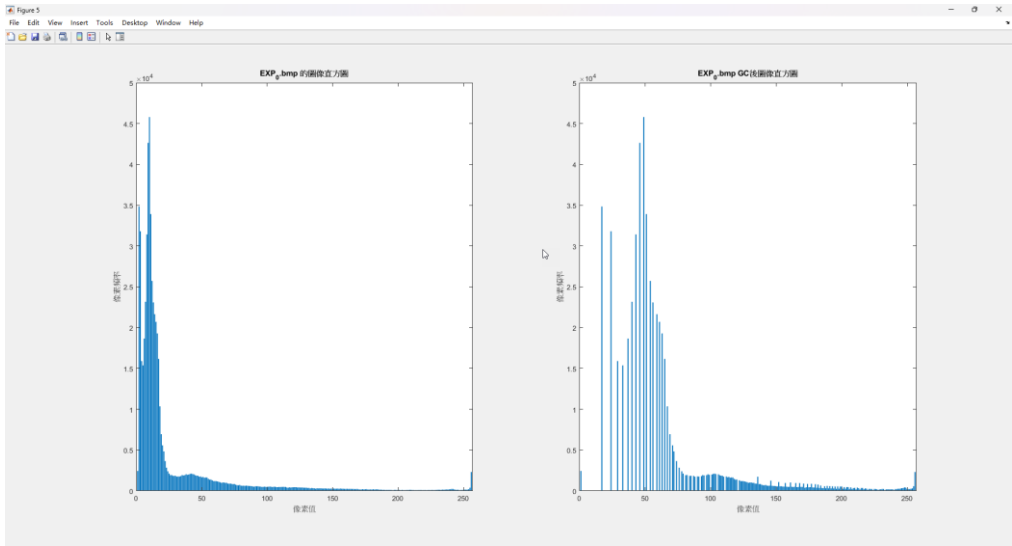


上圖，左邊是原始的圖像，右邊是 GC 後的圖像，從主觀角度，會發現 GC 的圖像有改善亮度，但是對於細節的呈現並沒有改善，因為外面的強度應該更高，但是 GC 並沒有辦法凸顯出來。

GC=gamma correction, AGC= adaptive gamma correction



上圖，左邊是 GC 後的圖像，右邊是 AGC 後的圖像，會發現 AGC 更能反映對比度，因為圖片強度平均值較低，經過運算，會使得圖片對比增加，而呈現更多資訊。



上

圖為 origin 和 GC 後的直方圖，可以看出經過 GC 後將暗區的點分開到較亮的區域，但是對比度的呈現沒有 AGC 好。

EXP4.bmp ($\gamma = 1.5$)

Name	entropy	brisque	niqe
origin	7.455342	32.750829	2.621499
AGC	7.276748	34.576359	2.814619
GC	7.243880	32.770779	2.708688

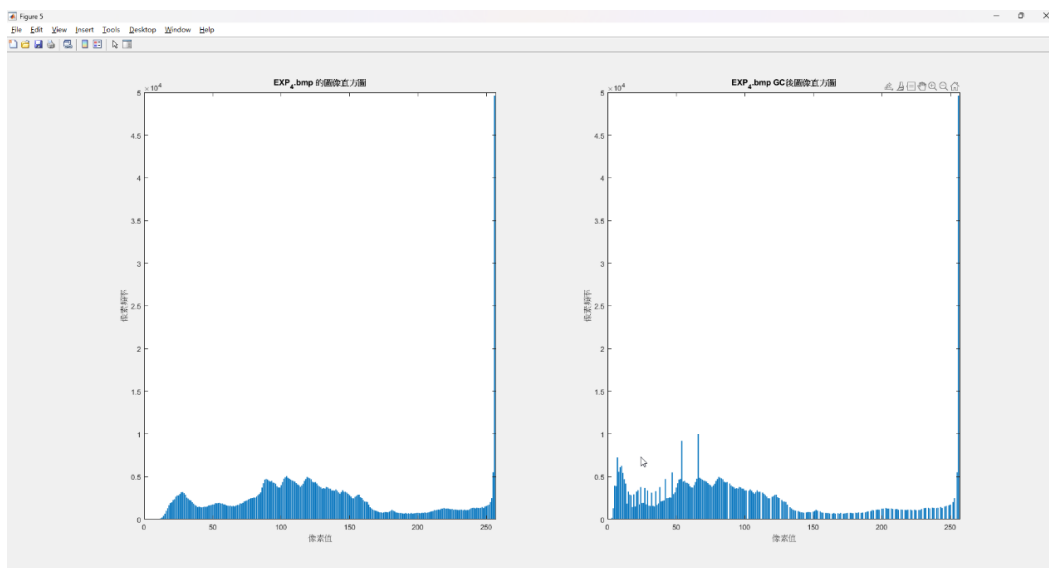
GC=gamma correction, AGC= adaptive gamma correction



左邊是原始的圖像，右邊是 GC 後的圖像，從主觀角度，覺得 GC 的圖像更好，並且與原圖比較在評分上，比 AGC 的表現更好。



上面這張，左邊是 GC 後的圖像，右邊是 AGC 後的圖像，會發現圖片後面的亮度一樣過曝，不過在前面的亮度呈現，GC 後的圖片有更好的對比度。



上圖為 origin 和 GC 後的直方圖，可以看出經過 GC 後有將部分亮區的點分散到暗區，整體變化不大，而且在暗區的顯示上有一些值特別多，因為 AGC 的算法減少改變對比度，所以看起來 GC 後的圖片更為銳利。

GC=gamma correction, AGC= adaptive gamma correction

7、 Experience

Challenge

在這次作業中我遇到的最大的困難是在處理常數 c 的部分，因為不熟悉 `hsv` 的規格和顯示，所以在執行 c 的運算時，時常會發生矩陣無法相乘的問題，與 AI 反覆詢問和用 `matlab debug` 的工具，才成功解決問題。

Summarize

從論文和 AI 中發現，並沒有最完美的 `correction`，每當增加某方向的加強，時常就會犧牲掉其他的資訊，並且在後續查資料時發現 `AGC` 的方法有很多種，同樣是根據圖片來改變 c 和 γ ，但是不同的算法，計算出的結果也不一樣，而我選擇的方案只是眾多方案的其中之一，未來可以分析不同的計算方式哪個更好，或是整合出更優解。

Cooperation

都是從三個主題出發，為了有更廣的學習機會，我選擇了和朋友不同的研究方向，可以在做完後互相分享所學到的知識，並且在製作過程和同學討論程式碼，一起解決遇到的問題。

8、 Feedback of using AI guidance

使用 `ChatGPT`，詢問缺點和改善方法，提供給我不同增強 `gamma correction` 的方法，剛開始提供了結合 `HE` 和 `GC` 的方法，當我繼續深入詢問不同方法，我發現它會開始回覆相同的答案，而我轉而使用 `bingAI` 就提供我 `AGC` 的解法，所以當使用其中一個 AI 工具無法得到預期的效果時，可以使用不同的工具和詢問方法，可以得到不一樣的答案，不過有時會是錯誤的答案，還需要後續查證。在程式碼方面，有些時候會遇到不會撰寫的部分，也可以問 AI，它會提供簡單的範例與解釋，讓我在自學上更方便。

未來在使用 AI 方面，它減少了我們從問題到答案的路徑，可以更快速的找到需要的資料，不論是 `kaggle` 的比賽程式碼解釋，或是期末作業，它都能給予幫助，幫助我探索更多知識。

9、 Reference

<https://jivp-urasipjournals.springeropen.com/articles/10.1186/s13640-016-0138-1>
[chatgpt](#)
[bingAI](#)