

# 高等影像處理

## 作業#1

姓名：\_\_\_\_\_蘇柏凱\_\_\_\_\_

學號：\_\_\_\_\_111c71007\_\_\_\_\_

指導老師：\_\_\_\_\_李曉祺、蔣欣翰\_\_\_\_\_

## 1.1 Raw image file format understanding

Figure

Lena256.raw



Discussion

在工具的設置中需要把長度寬度都設成 256，色深為 8

## 1.2 Raw image file input/output

Discussion

a.(1) : 213

a.(2) : 72

Figure

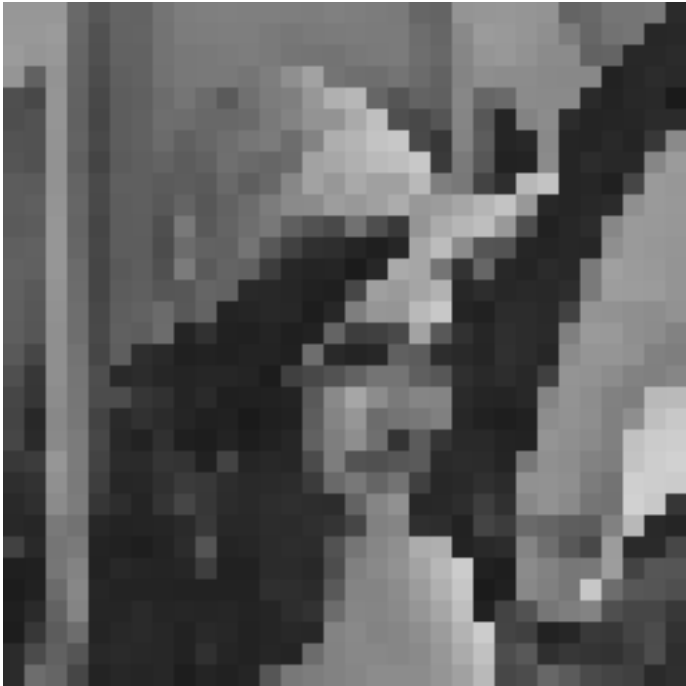
b.lena256\_out.raw



Figure

c.lena256\_out\_divide



Discussion
<p>c.分割部分採用 6 條直線的斜率把圖片分成八個區塊，在分割好後觀察圖片，利用旋轉矩陣 <math>x = x * \cos(\theta) - y * \sin(\theta)</math>, <math>y = x * \sin(\theta) + y * \cos(\theta)</math> 對各圖片順時針旋轉 90 度、180 度、270 度，最後平移回原點再合併。值得注意的是因為 <math>\pi</math> 的誤差還有 pixel 為偶數，影像在切割時會有黑線殘留，這時候需要採用雙精浮點數計算、注意 pixel 起始位置。</p>
Figure
<p>d. lena256out_minpool.raw</p> 
Discussion
<p>原始圖片大小為 256*256 依據題目需求採用一個 8*8 大小的 filter，並在 filter 內的 8 個 pixel 找到其中的最小值，並用最小值替代掉 8*8 方框內的所有 pixel，可以看見採用此種方式進行 down sampling 讓圖片看起來較為模糊，也同時讓圖片看起來較為黯淡</p>

### 1.3 Brightness

Figure

a. lena256out\_bright.raw



b. lena256out\_bright\_random.raw



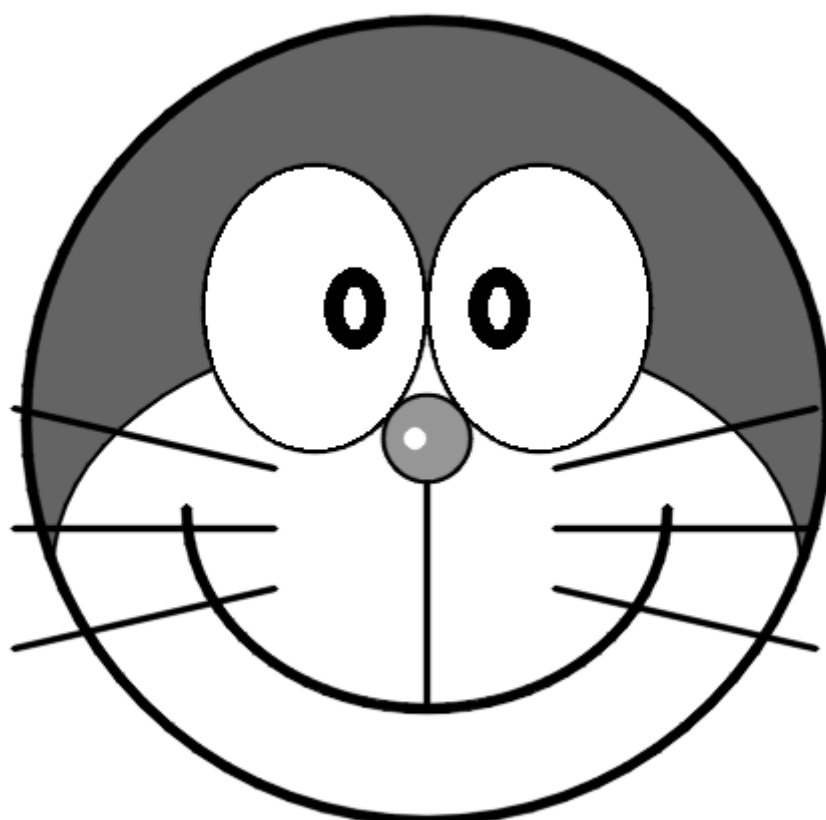
Discussion

c.在進行增加明亮度的行為時可以注意到部分 pixel 會到 308 而超出原始圖片深度 8 bits 最多到 256 大小的設定造成 overflow，而在 b 部分隨機對每個 pixel 增減 60 的時候會也會造成 underflow，為了解決這個問題我找到每一個圖片中最大值、最小值，並採用 min-max 的方式進行 normalized,即  $\text{pixel} - \text{min} / (\text{max} - \text{min})$ ，最後把圖片乘上 255，把強度還原成 0~255

## 2 OpenCV image I/O(c/c++ with OpenCV)

Figure

111c71007.png



111C71007

### Discussion

利用 OpenCV 繪製一個哆拉 A 夢可以發現最重要的是圖片的繪製順序問題，起初在繪製時以為會是一個複雜的問題，但是逐一分解後可以發現所有的圖案都可以由圓形、橢圓形、直線完成，自己手動畫完一個可愛的動畫人物讓人感到成就感十足。