Comments

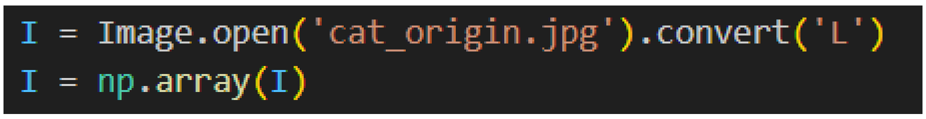
(a) How to determine the entry values of a dither matrix?

Dn 矩陣大小為(2xn)x(2xn)，所以 D1 矩陣為 2x2，D2 為 4x4…以此類推，從此次作業得到的結果，個人覺得經過 D2 處理的的圖像比 D1 更還原。

(b) How to arrange the positions of entry values?

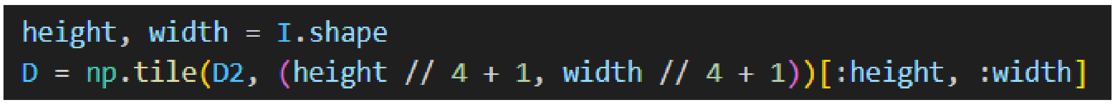
在生成抖動矩陣（dither matrix）的方法中，常見的有 Floyd-Steinberg Dithering、Jarvis-Judice-Ninke Dithering、Ordered Dithering 和 Random Dithering 等。此次作業選擇使用 Random Dithering，由於其隨機性，能有效減少圖像中的人工感，並降低生成抖動矩陣的計算複雜度。

(c) During extending dither matrix to dither array, how to fit the boundaries of dither array and image array?



```
I = Image.open('cat_origin.jpg').convert('L')
I = np.array(I)
```

先算出圖片的大小

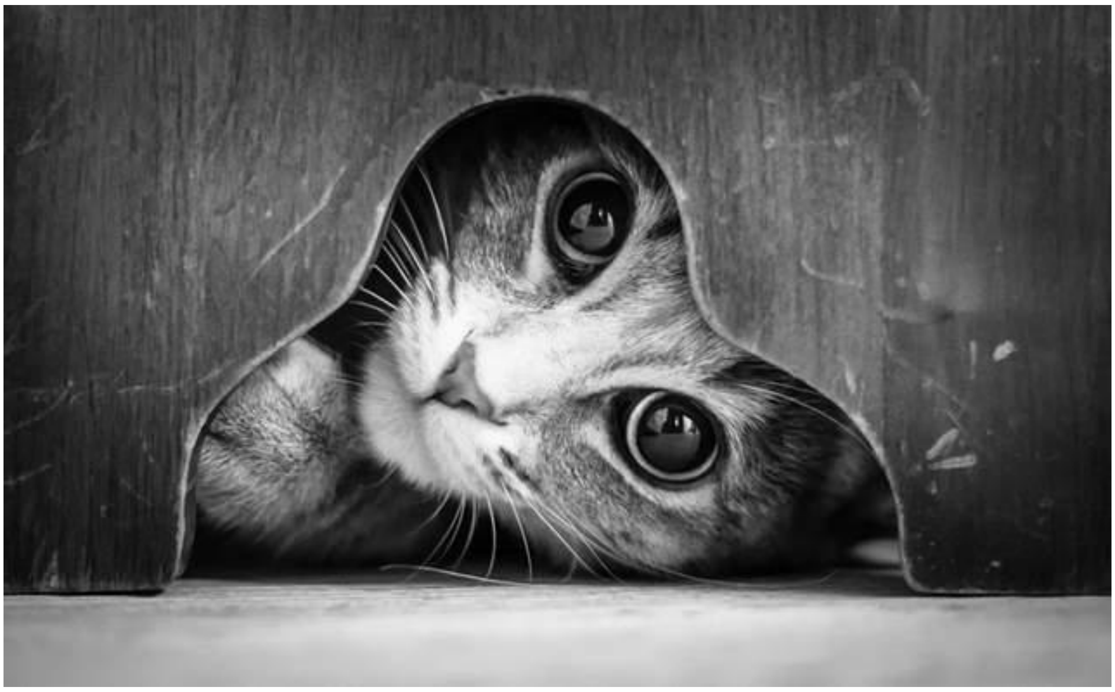


```
height, width = I.shape
D = np.tile(D2, (height // 4 + 1, width // 4 + 1))[:height, :width]
```

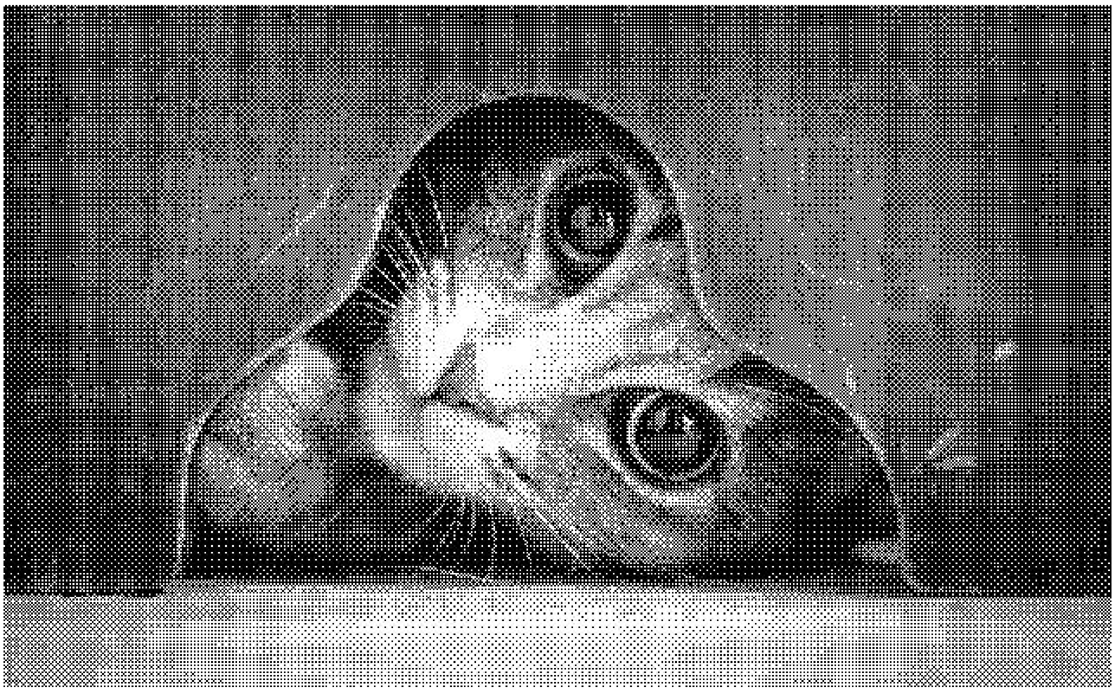
得到圖片長寬後，除以一個 dither matrix 的維度大小(例: D2 就是除以 4)，怕除完有小數點，所以多+1，確保整張圖都有被 dither matrix 覆蓋到

Photos

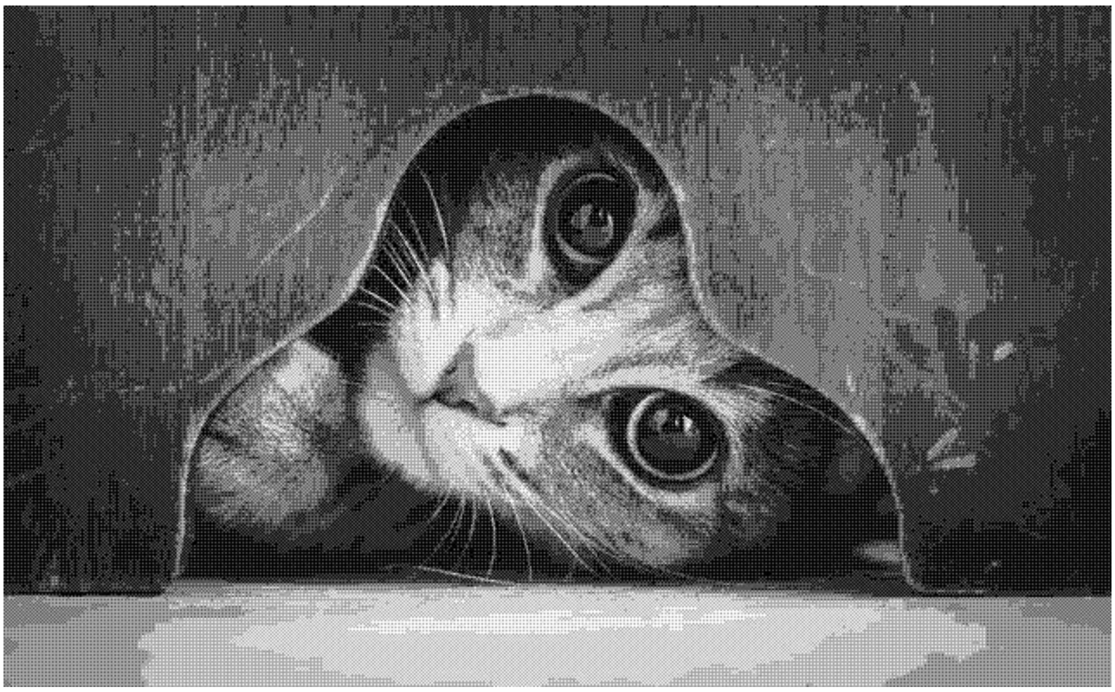
Origin



2-level dithered



4-level dithered



Program

```
import numpy as np

from PIL import Image

import matplotlib.pyplot as plt



I = Image.open('cat_origin.jpg').convert('L')

I = np.array(I)



# 2-level dither

D2 = np.array([

    [0, 128, 32, 160],

    [192, 64, 224, 96],

    [48, 176, 16, 144],
    [240, 112, 208, 80]
])


height, width = I.shape
D = np.tile(D2, (height // 4 + 1, width // 4 + 1))[:height, :width]
I_2prime = np.where(I > D, 255, 0).astype(np.uint8)


# 4-level dither
D1 = np.array([
    [0, 56],
    [84, 28]
])
height, width = I.shape
D = np.tile(D1, (height // 2 + 1, width // 2 + 1))[:height, :width]
step = 85
Q = (I // step).astype(int)
I_4prime = Q + (I - step * Q > D).astype(int)
I_4prime = (I_4prime * step).astype(np.uint8)


Image.fromarray(I_2prime).save('cat_dithered_2_level.jpg')
Image.fromarray(I_4prime).save('cat_dithered_4_level.jpg')


plt.figure(figsize=(15, 5))
plt.subplot(1, 3, 1)

plt.imshow(I, cmap='gray')

plt.title('Original Image')

plt.axis('off')


plt.subplot(1, 3, 2)

plt.imshow(I_2prime, cmap='gray')

plt.title('2-level Dithered Image')

plt.axis('off')


plt.subplot(1, 3, 3)

plt.imshow(I_4prime, cmap='gray')

plt.title('4-level Dithered Image')

plt.axis('off')


plt.show()
```