

# 画像工学 課題 3

E1728 永田大和

提出日：2021/12/06

## ◇実行方法

言語は Python3.8 で書かれており、実行には最新版の numpy, pillow, scipy を必要とする。各種関数を適切に組み合わせたのち、記述された python 実行ファイルをコマンドラインから実行するものとする。

## ◇コード

課題内容すべての前に、以下のコードが記述されているものとする。つまり、以下のコードはすべてのプログラムで使用する。

```
1. import math
2.
3. import numpy as np
4. from PIL import Image
5. from scipy.fft import dct, idct
6.
7.
8. def read_img(filename: str) -> np.ndarray:
9.     return np.asarray(Image.open(filename))
10.
11.
12. def save_img(image: np.ndarray, filename: str):
13.     img = Image.fromarray(image)
14.     img.save(filename)
15.
16.
17. def dct2(a):
18.     return dct(dct(a, axis=0, norm='ortho'), axis=1, norm='ortho')
19.
20.
21. def idct2(a):
22.     return idct(idct(a, axis=0, norm='ortho'), axis=1, norm='ortho')
23.
24.
25. def psnr(original: np.ndarray, edited: np.ndarray) -> float:
```

```

26.     mse = np.mean((original.astype(float) - edited.astype(float)) ** 2)
27.     return 10 * np.log10((255 ** 2) / mse)
28.
29.
30. def dct_each_block(array: np.ndarray, dct_callback: callable, size=8) ->
    np.ndarray:
31.     if len(array.shape) != 2:
32.         raise ValueError(f'array is not a 2-Dimensional array. given shape:
            {array.shape}')
33.     height, width = array.shape
34.     if not isinstance(height, int) or not isinstance(width, int):
35.         raise ValueError(f'array is not a valid 2-Dimensional array. given shape:
            {array.shape}')
36.     if height % size or width % size:
37.         raise ValueError(f'array is not dividable into {size}-sized block. given
            shape: {array.shape}')
38.
39.     new = np.zeros(array.shape)
40.     for y in range(height // size):
41.         for x in range(width // size):
42.             dct_val = dct2(array[y * size:(y + 1) * size, x * size:(x + 1) * size])
43.             dct_val = dct_callback(dct_val)
44.             new[y * size:(y + 1) * size, x * size:(x + 1) * size] = dct_val
45.     return new
46.
47.
48. def idct_each_block(array: np.ndarray, size=8) -> np.ndarray:
49.     height, width = array.shape
50.     new = np.zeros(array.shape)
51.
52.     for y in range(height // size):
53.         for x in range(width // size):
54.             new[y * size:(y + 1) * size, x * size:(x + 1) * size] = ¥
55.             idct2(array[y * size:(y + 1) * size, x * size:(x + 1) * size])
56.     return new.round().astype(np.uint8)

lena = read_img('LENA.png')

```

## ◇課題 1

```
1. def callback_1(array: np.ndarray) -> np.ndarray:
2.     return np.rot90(np.tril(np.rot90(array)), 3)
3.
4.
5. quiz1 = dct_each_block(lena, callback_1)
6. i_quiz1 = idct_each_block(quiz1)
7. save_img(i_quiz1, 'quiz1.png')
8. print(psnr(lena, i_quiz1))
9.
```

結果:

SN 比 : 36.39037633155074

再構成画像:



## ◇課題 2

```
1. def callback_2(minimum: int) -> callable:
2.     def wrapper(array: np.ndarray) -> np.ndarray:
3.         new = array.copy()
4.         new[np.abs(new) <= minimum] = 0
5.         return new
6.
7.     return wrapper
8. quiz2 = dct_each_block(lena, callback_2(10))
9. i_quiz2 = idct_each_block(quiz2)
10. save_img(i_quiz2, 'quiz2.png')
11. print(psnr(lena, i_quiz2))
12.
```

結果:

SN 比 : 38.85567114145475

再構成画像:



### ◇課題 3

```
1. def callback_3(ratio: float) -> callable:
2.     def wrapper(array: np.ndarray) -> np.ndarray:
3.         flat = np.abs(array.flatten())
4.         flat.sort()
5.         min_index = math.ceil(len(flat) * ratio) - 1
6.         if min_index == len(flat):
7.             min_index -= 1
8.         minimum = flat[min_index]
9.
10.        new = array.copy()
11.        new[np.abs(new) <= minimum] = 0
12.        return new
13.
14.    return wrapper
15.
16. quiz3 = dct_each_block(lena, callback_3(0.6))
17. i_quiz3 = idct_each_block(quiz3)
18. save_img(i_quiz3, 'quiz3.png')
19. print(psnr(lena, i_quiz3)) # 39.71817190326393
20.
```

結果:

SN 比 : 39.71817190326393

再構成画像: (次ページ)



## ◆課題 4

```
1. def callback_4(ratio: float) -> callable:
2.     def wrapper(array_: np.ndarray) -> np.ndarray:
3.         array = array_.copy()
4.         size, _ = array.shape
5.         index = 0
6.         from_index = array.size * (1 - ratio)
7.         for trial in range(size * 2 - 1):
8.             y = trial
9.             y = min(y, size - 1)
10.            for offset in range(trial - y, min(y, size) + 1):
11.                y_ = y - offset + max(0, trial - size + 1)
12.                if trial % 2:
13.                    y_, offset = offset, y_
14.                if index >= from_index:
15.                    array[y_, offset] = 0
16.                    index += 1
```

```
17.         return array
18.
19.     return wrapper
20.
21. quiz4 = dct_each_block(lena, callback_4(0.6))
22. i_quiz4 = idct_each_block(quiz4)
23. save_img(i_quiz4, 'quiz4.png')
24. print(psnr(lena, i_quiz4))
25.
```

結果:

SN 比 : 33.345978863543294

再構成画像:



## ◇課題 5

なお、この課題のみ、二次元逆 DCT 変換の過程でほかのプログラムと異なった動作をする必要があるため、特別に定義している。



```

1. def callback_5(q: float, is_dct=True) -> callable:
2.     T = np.asarray([[16, 11, 10, 16, 24, 40, 51, 61],
3.                     [12, 12, 14, 19, 26, 58, 60, 55],
4.                     [14, 13, 16, 24, 40, 57, 69, 56],
5.                     [14, 17, 22, 29, 51, 87, 80, 62],
6.                     [18, 22, 37, 56, 68, 109, 103, 77],
7.                     [24, 35, 55, 64, 81, 104, 113, 92],
8.                     [49, 64, 78, 87, 103, 121, 120, 101],
9.                     [72, 92, 95, 98, 112, 100, 103, 99]])
10.    if not 0 <= q < 100:
11.        raise ValueError(f'{q} is not valid for argument `q`. 0 <= q < 100 is only
12.        acceptable.')
13.    if q < 50:
14.        Q = T * 50 / q
15.    else:
16.        Q = T * (100 - q) / 50
17.    def wrapper(array: np.ndarray) -> np.ndarray:
18.        if is_dct:
19.            return np.round(array / Q)
20.        return np.round(array * Q)
21.    return wrapper
22.
23.
24.
25. def idct_each_block_quiz5(array: np.ndarray, q: float, size=8) -> np.ndarray:
26.     height, width = array.shape
27.     new = np.zeros(array.shape)
28.
29.     for y in range(height // size):
30.         for x in range(width // size):
31.             new[y * size:(y + 1) * size, x * size:(x + 1) * size] = ¥
32.             idct2(callback_5(q, False)(array[y * size:(y + 1) * size, x *
33.             size:(x + 1) * size]))
34.     return new.round().astype(np.uint8)

```

```
35. q = 30
36. quiz5 = dct_each_block(lena, callback_5(q))
37. i_quiz5 = idct_each_block_quiz5(quiz5, q)
38. save_img(i_quiz5, 'quiz5.png')
39. print(psnr(lena, i_quiz5))
40.
```

結果:

SN 比 : 31.626378319520843

再構成画像:

