

# 知能情報実験 1 : レポート課題 5

205713B 朝比奈 太郎

2021 年 6 月 10 日

## 目次

1	目的	2
2	方法	2
2.1	コーディング環境 . . . . .	2
2.2	検証内容 . . . . .	2
2.3	画像の絵画方法 . . . . .	2
3	結果	3
3.1	2.3 の画像の絵画方法のソースコード . . . . .	3
3.2	問 1 . . . . .	3
3.3	問 2 . . . . .	4
3.4	問 3 . . . . .	4
4	考察	5

## 1 目的

Numpy は行列処理に適しており、画像は画素を 2 次元的に配置した集まりである。従って、Numpy を用いて画像処理を行うことは適切であると確かめるため。また、画像を Numpy 行列として扱う理由は画像の大きさは画像の画素の数、すなわち高さ方向の画素数 (行列)\*幅方向の画素数 (列数) で表現されるから。

## 2 方法

### 2.1 コーディング環境

- PC のスペック: MacBook Air Core i5
- 使用言語: Python
- 使用ライブラリ: matplotlib.image, matplotlib.pyplot, numpy

### 2.2 検証内容

まず講義資料から、img\_x.png と img\_y.png をダウンロードした。講義内容や講義資料を参考にしながら img\_x.png と img\_y.png を加算、減算、白黒反転をそれぞれの問い (問 1,2,3) に沿って行った。

### 2.3 画像の絵画方法

\_\_main\_\_.py とは別に同じディレクトリ下に module\_img.py というファイルを作成し、その中に画像を絵画するための関数 plot\_img(img) を作成した。

## 3 結果

### 3.1 2.3 の画像の絵画方法のソースコード

Listing 1 module\_img.py

```
1 import matplotlib.pyplot as plt
2 import matplotlib as mpl
3 mpl.use('TkAgg')
4
5
6 def plot_img(img):
7     fig = plt.figure()
8     ax = fig.add_subplot(111)
9     ax.imshow(img, cmap="gray")
10    plt.show()
```

### 3.2 問 1

Listing 2 問 1

```
1 from matplotlib.image import imread, imsave
2 from matplotlib.pyplot import plot
3 import numpy as np
4 from . import module_img as m_img
5
6 img_x = imread(fname="./report5_1/data_input/img_x.jpg",
7                format="jpg")[:, :, 0].astype(np.float64)
8 img_y = imread(fname="./report5_1/data_input/img_y.jpg",
9                format="jpg")[:, :, 0].astype(np.float64)
10
11 img = np.add(img_x, img_y)
12 img_inv = np.subtract(255, img)
13
14 m_img.plot_img(img_inv)
```

### 3.3 問2

Listing 3 問2

```
1 from matplotlib.image import imread, imsave
2 from matplotlib.pyplot import plot
3 import numpy as np
4 from . import module_img as m_img
5
6 img_x = imread(fname="./report5_2/data_input/img_x.jpg",
7               format="jpg")[:, :, 0].astype(np.float64)
8 img_y = imread(fname="./report5_2/data_input/img_y.jpg",
9               format="jpg")[:, :, 0].astype(np.float64)
10
11 img_x_inv = np.subtract(255, img_x)
12
13 img_y_inv = np.subtract(255, img_y)
14
15 img = np.add(img_x_inv, img_y_inv)
16
17 m_img.plot_img(img)
```

### 3.4 問3

Listing 4 問3

```
1 from matplotlib.image import imread, imsave
2 from matplotlib.pyplot import plot
3 import numpy as np
4 from . import module_img as m_img
5
6 img_x = imread(fname="./report5_3/data_input/img_x.jpg",
7               format="jpg")[:, :, 0].astype(np.float64)
8 img_y = imread(fname="./report5_3/data_input/img_y.jpg",
9               format="jpg")[:, :, 0].astype(np.float64)
10
11 img_x_inv = np.subtract(255, img_x)
12
13 img = np.subtract(img_x_inv, img_y)
14
15 m_img.plot_img(img)
```

## 4 考察

まず、画素の濃度は 0-255 までの 256 階調あり、0 に近づくほど濃度が低くなり (暗くなり), 255 に近づくほど濃度が高くなる (明るくなる)。Numpy の減算関数 (`subtract(X,Y)`) を用いると、 $X - Y$  が行われる。従って、`subtract(255, x)` を行うと、画像は Numpy 行列として扱えることから白黒反転した画像が出力される。問 1 では、画像  $x$  と画像  $y$  の加算画像を `subtract` を用いて白黒反転したので、画像  $x$  と画像  $y$  を加算した画像を画像  $P'$  とすると、 $P'$  に `subtract` の処理を加えて画像  $P$  が生成されたということである。問 2 では、画像  $x$  を白黒反転したものを  $x'$ 、画像  $y$  を白黒反転したものを  $y'$  とすると、 $x'$  と  $y'$  の加算画像は  $P$  となり、 $P$  が生成されたということである。問 3 では、画像  $x'$  に画像  $y$  を減算するという事は、画像  $x'$  に画像  $y'$  を加算することと意味が等しいことから問 2 同様、画像  $P$  が生成されたということである。問 1 と問 2 では、白黒反転することと加算することの順番が異なるだけであるから、本質的には問 1 も問 2 同様に画像  $P$  を生成したと言える。ゆえに、問 1,2,3 は演算プロセスは異なるものの同一の画像が生成されるといえる。

## 参考文献

- [1] 國田 樹, 2021\_StuLab1\_理工系のレポート作成技術, 2021/06/10.
- [2] Latex 入門/図表, <https://texwiki.texjp.org/?LaTeX>, 2021/06/10
- [3] LaTeX 簡条書き, [http://www.yamamo10.jp/yamamoto/comp/latex/make\\_doc/item/item.php](http://www.yamamo10.jp/yamamoto/comp/latex/make_doc/item/item.php), 2021/06/10