

# レポート提出票

科目名: 情報工学実験3

実験課題名: 課題4 画像変換

実施日: 2021年4月22日

学籍番号: 4619055

氏名: 辰川力駆

共同実験者:

_____	_____
_____	_____
_____	_____
_____	_____

# 1 実験の要旨

実験環境を整備した上で、Jupyter notebook で Python 言語とそのライブラリの使い方を理解し、画像処理の基礎を学習する。

## 2 実験の目的

画像変換の処理を題材に、画像処理のプログラミングと評価を通じて、画像処理の基本的な考え方を理解することを目的とする。

第1回目の実験では、濃淡変換の実装と実験を通じて、Python 言語とライブラリの使用方法、画像処理の基本を理解することを目的とする。

## 3 課題1

### 3.1 実験方法

次のようなプログラムを作成する。ただし、指定の領域は、2点 (64,25),(192,220) を対角とする矩形である。

1. 画像ファイル data/Mandrill.bmp を読み込み画像 im1 とする。
2. 画像 im1 の指定の領域をグレースケールにした画像 im2 を作成する。
3. 画像 im1 の指定の領域を上下反転した画像 im3 を作成する。
4. 画像 im1、im2、im3 にタイトルを付けて、横に並べて表示する。

### 3.2 実験結果

作成したプログラムは付録のソースコード 1 に載せた。このプログラムを実行すると、以下のようになった。

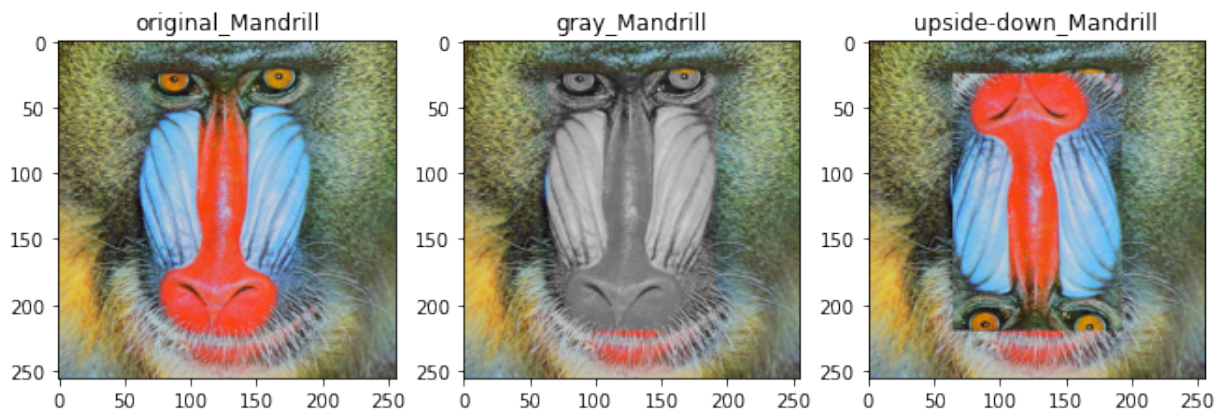


図 1: 課題1の実行結果

### 3.3 検討・考察

im1の一部を反転させるのは、容易であったが (upside-down\_Mandrill)、im1の一部をグレースケールにするには、一点一点を変更していく (プログラム 11 行目から 13 行目) ので、高速に実行するのは難しいと考える。

## 4 課題 2

### 4.1 実験方法

自分自身で撮影した画像  $f$  に対して、以下を実行する。

1. ガンマ値  $\gamma$  を変化させながら、画像  $f$  にガンマ補正を施した結果を求める。
2. ガンマ補正によるコントラスト向上の効果を確認する。そして、見た目が良くなるガンマ値を選択する。
3. 2つ以上のガンマ値に対して補正結果を図示する。

### 4.2 実験結果

ガンマ値  $\gamma$  を変化させながら、画像  $f$  にガンマ補正を施した結果を下記に示す。

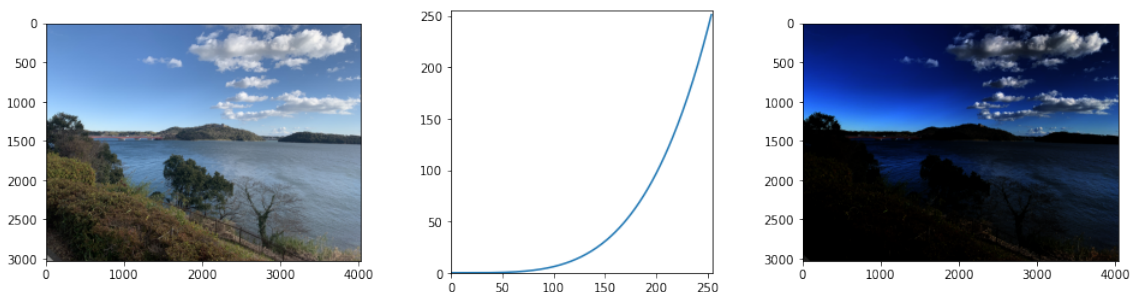


図 2:  $\gamma = \frac{1}{4}$  の場合

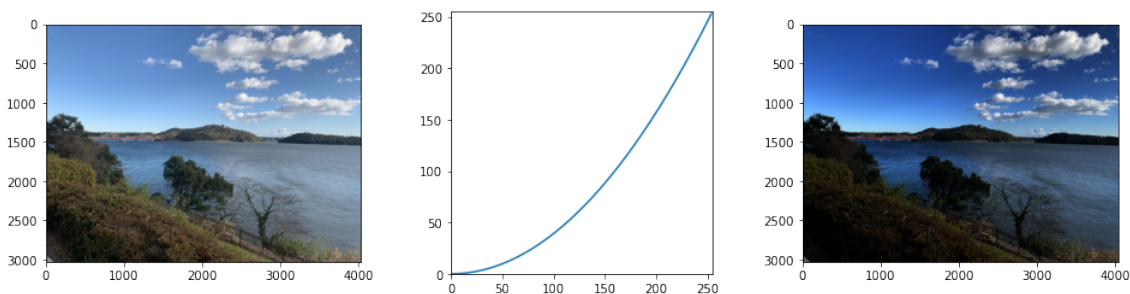


図 3:  $\gamma = \frac{1}{2}$  の場合

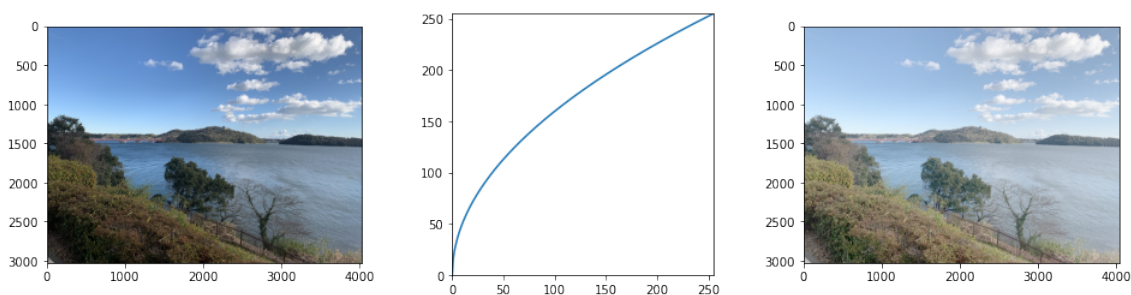


図 4:  $\gamma = 2$  の場合

また、作成するために用いたプログラムは付録のソースコード 2 に載せた。

### 4.3 検討・考察

実験の結果より、 $\gamma$  の値を 1 より小さくすることで全体が暗くなり、 $\gamma$  の値を 1 より大きくすることで全体が明るい画像となった。

今回の画像に関しては元が明るい風景なので、 $\gamma$  の値を図 4 のように明るくするのではなく、図 2,3 のように暗くするほうが見た目が良いと考える。 $\gamma$  の値を小さくしすぎると海の色が真っ黒になってあまり見えなくなるので主観的に一番見た目が良いと思うものは  $\gamma = \frac{1}{2}$  (図 3) である。

図 3 は、元の画像でぼんやりしている雲をくっきりしているので見た目が良くなっている。

## 5 課題 3

### 5.1 実験方法

### 5.2 実験結果

### 5.3 検討・考察

## 6 まとめ

### 参考文献

## A 付録

ソースコード 1: kadai1

```
1 #4619055
2 from skimage.io import imread, imsave
3 import numpy as np
4 import matplotlib.pyplot as plt
5 plt.figure(figsize=(11,7))
6
7 im1 = imread("data/Mandrill.bmp") # カラー画像として読み込み
8 im2 = imread("data/Mandrill.bmp")
9 im3 = imread("data/Mandrill.bmp")
10
11 for x in range(64,192):
12     for y in range(25,220):
13         im2[y,x] = np.dot(im2[y,x],[0.299, 0.587, 0.114])
14
15 im3[25:220,64:192]= np.flipud(im3[25:220,64:192])
16
17 plt.subplot(1, 3, 1)
18 plt.imshow(im1)
19 plt.title("original_Mandrill")
20 plt.subplot(1, 3, 2)
21 plt.imshow(im2)
22 plt.title("gray_Mandrill")
23 plt.subplot(1, 3, 3)
24 plt.imshow(im3)
25 plt.title("upside-down_Mandrill")
26 plt.show()
```

ソースコード 2: kadai2

```
1 #4619055
2 import numpy as np
3 from skimage.io import imread, imsave
4 from skimage.color import rgb2gray, gray2rgb
5 import matplotlib.pyplot as plt
6 %matplotlib inline
7
8 def show_transform(im1, im2, h):
9     """ 濃淡変換の入力画像, トーンカーブ, 出力画像を描画
10         im1 : 入力画像
11         im2 : 出力画像
12         h : 階調変換関数
13     """
14     plt.figure(figsize=(16, 4))
15     plt.subplot(1, 3, 1)
16     plt.imshow(im1.astype(np.uint8), cmap="gray")
17
18     plt.subplot(1, 3, 2)
19     I = np.arange(0, 255)
20     plt.xlim(0, 255); plt.ylim(0, 255); plt.gca().set_aspect(1.0)
```

```

21     try:
22         y = h(I)
23         plt.plot(I, y)
24     except:
25         x = np.vstack([I, I, I])
26         y = h(x.T)
27         plt.plot(I, y[0,:,0], 'r', label='R')
28         plt.plot(I, y[0,:,1], 'g', label='G')
29         plt.plot(I, y[0,:,2], 'b', label='B')
30         plt.legend()
31
32     plt.subplot(1, 3, 3)
33     plt.imshow(im2.astype(np.uint8), cmap="gray")
34
35     def gamma(gam, I): return 255 * (I / 255) ** (1/gam)
36     def gamma14(I): return gamma(1/4, I)
37     def gamma12(I): return gamma(1/2, I)
38     def gamma2(I): return gamma(2, I)
39
40     f = imread("data/kadai2.jpg")
41
42     show_transform(f, gamma14(f), gamma14)
43     show_transform(f, gamma12(f), gamma12)
44     show_transform(f, gamma2(f), gamma2)

```