

デジタルメディア処理1

担当: 井尻 敬

デジタルメディア処理1、2017（後期）

- 09/26 イントロダクション1：デジタル画像とは，量子化と標本化，Dynamic Range
- 10/03 イントロダクション2：デジタルカメラ，人間の視覚，表色系
- 10/10 フィルタ処理1：トーンカーブ，線形フィルタ
- 10/17 フィルタ処理2：非線形フィルタ，ハーフトーニング
- 10/24 フィルタ処理3：離散フーリエ変換と周波数フィルタリング
- 11/07 前半のまとめと中間試験
- 11/14 画像処理演習：python入門（演習室）
- 11/21 画像処理演習：フィルタ処理（演習室）
- 11/28 画像処理演習：フィルタ処理（演習室）
- 12/05 画像処理演習：フィルタ処理（演習室）
- 12/12 画像の幾何変換1：アフィン変換
- 12/19 画像の幾何変換2：画像の補間
- 01/16 画像復元：ConvolutionとDe-convolution（変更する可能性有り）
- 01/23 後半のまとめと期末試験

演習課題 11/21

締め切り: 12/08 23:59

提出方法: 共有フォルダに『**dm1(学籍番号)**』というフォルダを作成し，その中にソースコードの入ったファイルを置く．フォルダ名は全て半角．

フォルダ名の例: dm1AL150999

課題雛形: http://takashijiri.com/classes/dm2017_1/dm1exer.zip

入出力 : 課題では入力画像を受け取り，出力画像を保存するプログラムを作る．入力画像と出力画像のファイル名は以下の通りコマンドライン入力より与えよ

```
$python exer*.py fname_in.png fname_out.png
```

注意 : 採点は自動化されています．フォルダ名・ファイル名やプログラムの仕様は指示に厳密に従ってください．入出力の仕様を満たさないコードは評価できず0点扱いとなります．

注意 : 今回は計算速度を重視しませんが，512x512程度の画像に対して20秒以上の計算時間がかかるものは0点とします．

課題1. 色の変換 (exer1.py)

カラー画像を読み込み，画像の赤と青チャンネルを入れ替えた画像を保存するプログラムを作成せよ

- ファイル名はexer1.py
- 実行コマンドは以下の通り

```
$python exer1.py fname_in.png fname_out.png
```

- コマンドライン引数の与え方はひな形を参照

課題2. ガウシアンフィルタ (exer2.py)

画像を読み込み、グレースケール画像に変換後、ガウシアンフィルタを掛けた画像を保存するプログラムを作成せよ

- ファイル名は exer2.py とし、実行コマンドは以下の通り

```
$python exer2.py fname_in.png fname_out.png
```

- グレースケール化の方法についてはひな形を参照
- 画像の周囲1pixelは計算せず0を入れること
- 今回はプログラミング練習が目的なので、次のフィルタ関数『cv2.filter2D() / cv2.GaussianBlur() / cv2.Sobel() / np.convolve()』は利用しないこと

1/16	2/16	1/16
2/16	4/16	2/16
1/16	2/16	1/16

フィルタの係数

課題3. ソーベルフィルタ (exer3.py)

画像を読み込み、グレースケール画像に変換後、横ソーベルフィルタを掛けた画像を保存するプログラムを作成せよ

- ファイル名は exer3.py とし、実行コマンドは以下の通り

```
$python exer3.py fname_in.png fname_out.png
```

- フィルタ適用後、負値となる画素は -1 倍して正值に変換すること
- フィルタ適用後、値が255を超える画素には255を代入すること
- 画像の周囲1pixelは計算せず0を入れること
- 今回はプログラミング練習が目的なので、次のフィルタ関数『cv2.filter2D() / cv2.GaussianBlur() / cv2.Sobel() / np.convolve()』は利用しないこと

-1	0	1
-2	0	2
-1	0	1

課題4. 勾配強度画像の作成 (exer4.py)

画像を読み込み、グレースケール画像に変換後、勾配強度画像を計算し保存するプログラムを作成せよ

- ファイル名は exer4.py とし、実行コマンドは以下の通り

```
$python exer3.py fname_in.png fname_out.png
```

- 画像の周囲1pixelは計算せず0を入れること
- ある画素の勾配強度は $I = \sqrt{f_x^2 + f_y^2}$ と求まる。ただし、 f_x と f_y はそれぞれ横方向・縦方向のソーベルフィルタの応答である。
- ※今回はプログラミング練習が目的なので、次のフィルタ関数『cv2.filter2D() / cv2.GaussianBlur() / cv2.Sobel() / np.convolve()』は利用しないこと

勾配強度を計算後、画素値が255を越えていたら255を代入してください

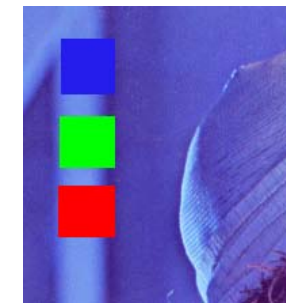
-1	0	1
-2	0	2
-1	0	1

1	2	1
0	0	0
-1	-2	-1

参考 (出力結果の例)



入力画像



課題1

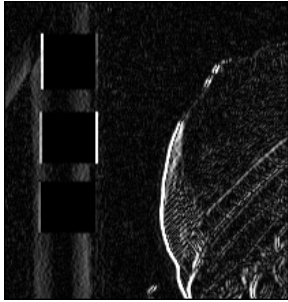


課題2

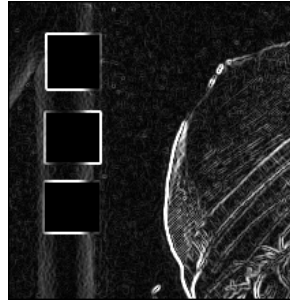
参考 (出力結果の例)



入力画像



課題3



課題4