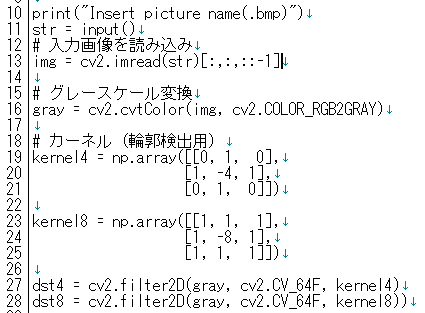
**景山研究室 画像処理プログラムに関する課題**

7018903　堀井萌希

1. 4 方向，8 方向のラプラシアンフィルタ

* ソースコード



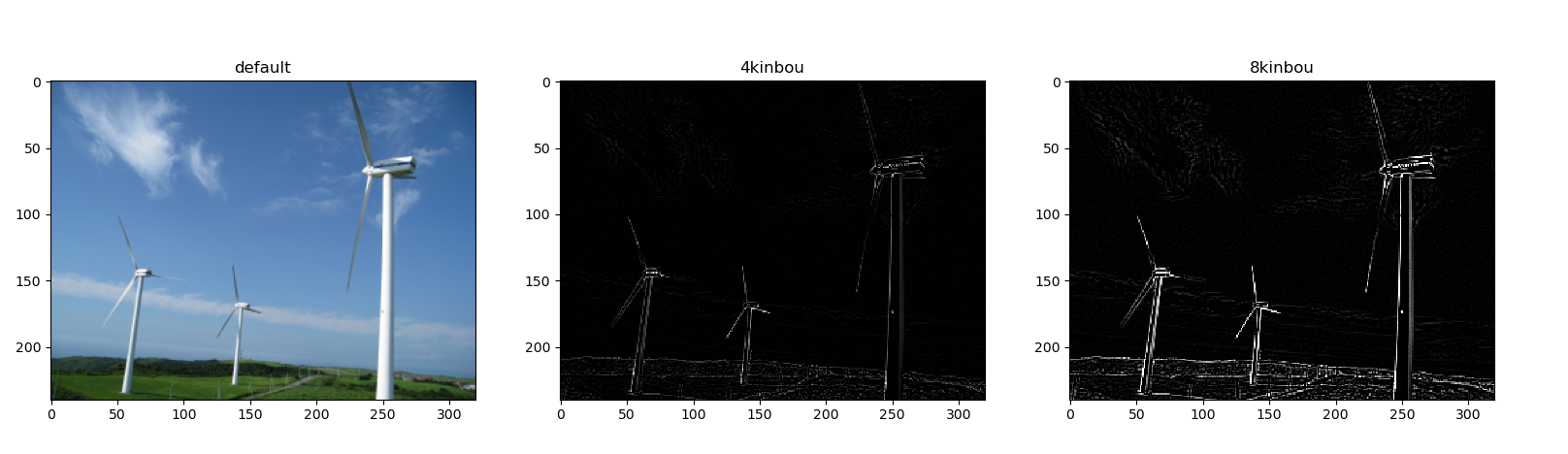
* 処理内容

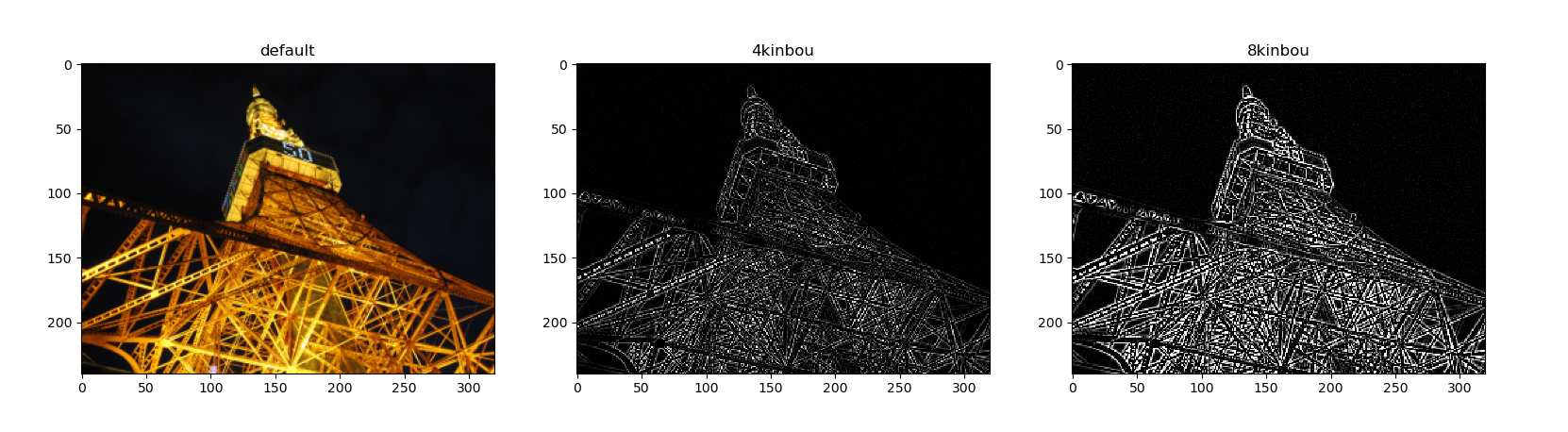
13行目に画像を読み込み、16行目でグレースケールに変換。19行目で4近傍のカーネルを行列として、23行目で8近傍のカーネルを行列として設定している。

27、28行目では、OpenCVで使えるcv2.filter2D()関数を使って空間フィルタリングを行っている。

表示はグレースケールで行っている

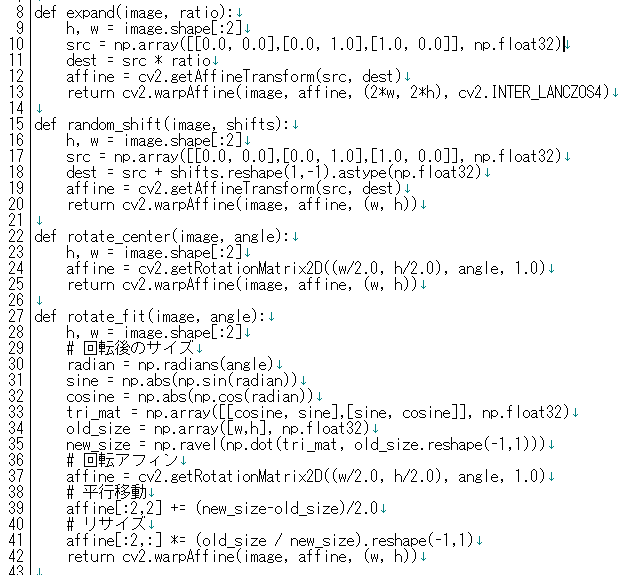
* 処理結果





1. アフィン変換

* ソースコード



* 処理内容

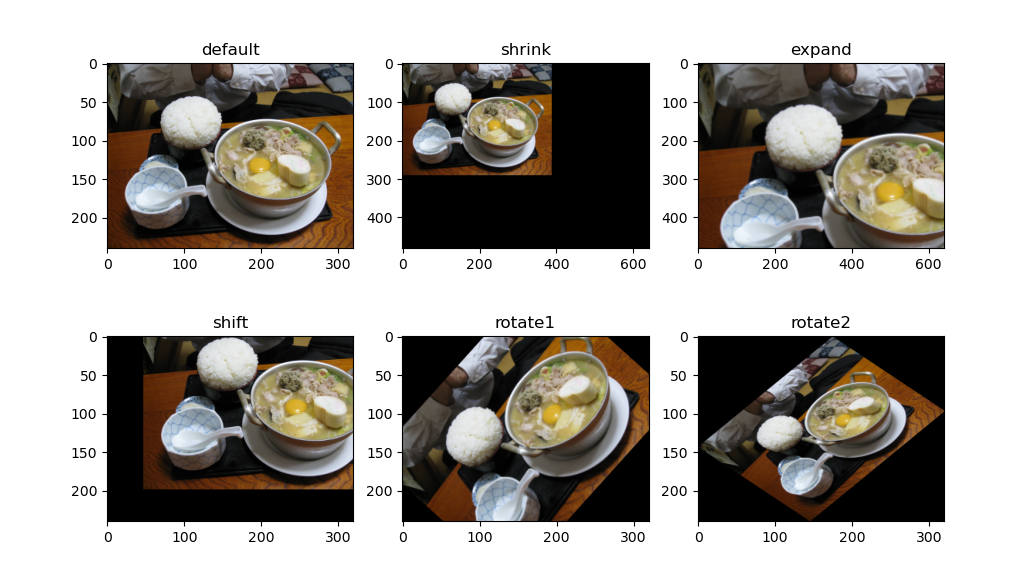
8行目から13行目は、画像の拡大・縮小を行っている関数であり、読み込んだ画像の原点と対角線上にある2つの点の3点を引数であるratio(画像サイズの倍率)倍してそれをdestとし、変換後の画像の座標を返している。

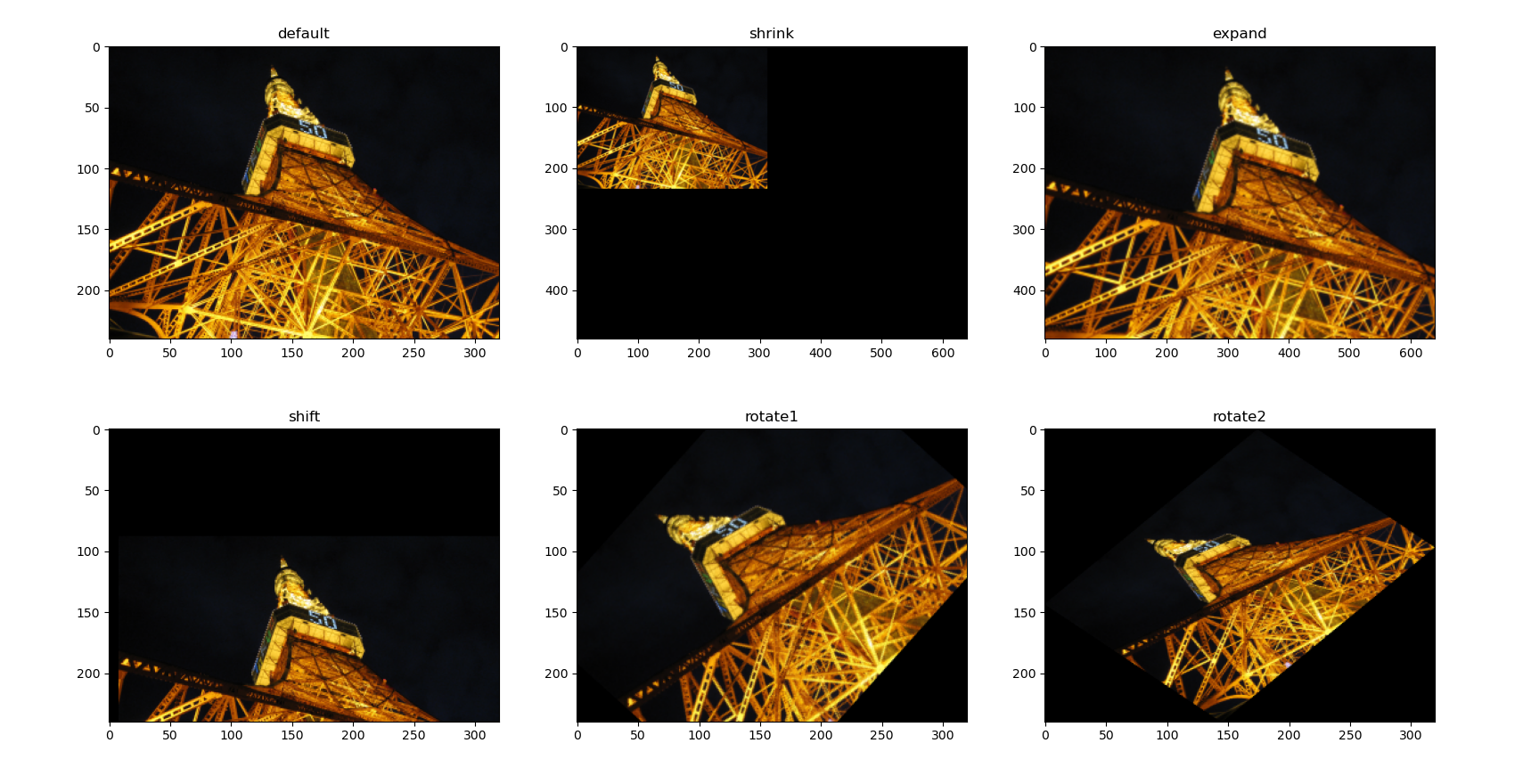
15行目から20行目は、画像の移動を行っている関数であり、拡大・縮小を行っている関数同様3・の座標を読み取って、引数であるshifts(移動するピクセル数)分移動させ、変換後の座標を返すというものになっている。

22行目から25行目は、画像の回転を行っている関数であり、同じく画像の座標を引数として読み取り、そこからcv2.getRotationMatrix2Dという関数を用いて画像の中心で回転させるような処理を行っている。

27行目から42行目は、同じく画像の回転を行っている関数であるが前の関数と違い、画像がはみ出さないような処理を行っている。30行目から35行目で回転後のサイズを取得し、37行目で回転を行い、39行目で移動の処理を行い、41行目でサイズの変更を行っている。

* 処理結果





1. 理解した点

本課題を通して、ラプラシアンフィルタとアフィン変換の求め方や仕組みをpythonに触れながら学ぶことができた。またアフィン変換というものの具体的な実装方法がわからなかったため、理解することができた。それに加え、pythonで行列式を使った際出力による数字の型の違いのエラーの対処の仕方も会得することができた。

1. 工夫した点

処理結果をわかりやすく1枚のウィンドウに変換前の画像と変換した後の画像複数枚のせてそれぞれの違いを分かりやすくした。