

畳み込み実装1

shun sato

畳み込み(Convolution)とは

フィルタの畳み込み

$$\begin{array}{r} 2 \times 2 + 1 \times 0 + 1 \times 1 \\ + 1 \times 1 + 0 \times 4 + 1 \times 2 \\ + 2 \times 0 + 0 \times 1 + 3 \times 0 = 8 \end{array}$$

フィルタ

$\frac{1}{11}$	3	0	2
	1	0	1
	1	1	2

入力画像

1	2	0	1	3	
2	1	4	2	2	
1	0	1	0	1	
1	2	1	0	2	
2	5	3	1	2	

出力画像

	15	8			
	11	11			

- カーネルを動かしながら積和を計算
 - その様子が畳み込んでいるように見える

↓ 動いている様子を見てみよう

<https://cvml-expertguide.net/terms/cv/image-filtering/convolution-for-image-filtering/>

数式で見てみよう

機械学習において、畳み込みとは以下のような処理です。いま、入力が2次元の場合を考えることにします。畳み込みカーネルを $K \in \mathbb{R}^{n_1 \times n_2}$ とするとき、入力 $I \in \mathbb{R}^{N_1 \times N_2}$ に対するカーネル K による畳み込み $I * K \in \mathbb{R}^{N_1 \times N_2}$ は、 $1 \leq x \leq N_1, 1 \leq y \leq N_2$ に対して

$$(I * K)(x, y) = \sum_{1 \leq k \leq n_1} \sum_{1 \leq l \leq n_2} I(x + k - 1, y + l - 1) K(k, l)$$

で定義されます。

画像 I の座標 (x, y) における畳み込み処理は上記の式で表される

実装してみよう

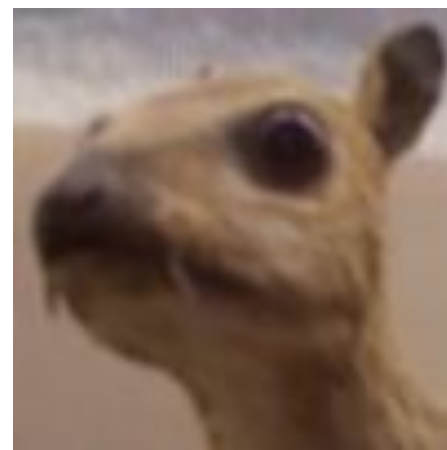
基本的なカーネル処理を行う関数を実装する

- ライブラリ
 - Numpy & OpenCV
- 画像サイズ
 - 入力=(1024x1024) -> 出力=(1022x1022)
- 画像形式
 - 1チャンネル(RGBで読み込んでグレースケール変換をしよう)
- 畳み込みカーネルサイズ
 - 3x3 (できれば任意のサイズで)

カーネルの例：ぼかしフィルタ(blur filter)

- 画像を全体的にぼかす

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



カーネルの例：微分フィルタ(sobel filter)

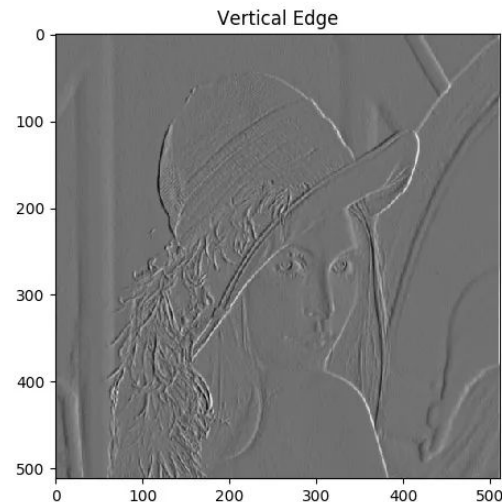
- 横方向(縦方向)のエッジを検出する

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

Vertical

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

Horizontal



カーネルの例：エッジ強調フィルタ(bilateral filter)

- 画像内のエッジが強調される

$$kernel = \frac{1}{2} \begin{pmatrix} -1 & -1 & -1 \\ -1 & 10 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

