

## Q. 71 - 80

---

### Q.71. マスキング

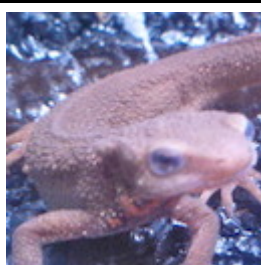
*imori.jpg*に対してHSVを用いて青色の箇所のみが黒くなるようにマスキングせよ。

このように白黒のバイナリ画像を用いて黒部分に対応する元画像の画素を黒に変更する操作をマスキングという。

青色箇所の抽出はHSVで $180 \leq H \leq 260$ となる位置が1となるような二値画像を作成し、その0と1を反転したものと元画像との積をとればよい。

これによりある程度のイモリの部分の抽出ができる。

入力 ( <i>imori.jpg</i> )	マスク ( <i>answers/answer_70.png</i> )	出力 ( <i>answers/answer_71.jpg</i> )
-------------------------	--------------------------------------	-------------------------------------



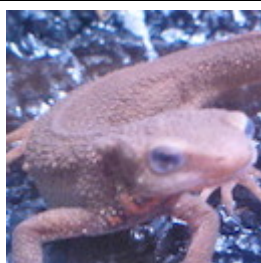
答え >> [answers/answer\\_71.py](#)

### Q.72. マスキング(カラートラッキング+モルフォロジー)

Q.71ではマスクが雑になってしまっていたので、イモリの目の部分が削除されていたり、背景がところどころ残ってしまった。

よってマスク画像にN=5のクロージング処理(Q.50)とオープニング処理(Q.49)を施してマスク画像を正確にして、マスキングを行え。

入力 ( <i>imori.jpg</i> )	マスク ( <i>answers/answer_72_mask.png</i> )	出力 ( <i>answers/answer_72.jpg</i> )
-------------------------	---	-------------------------------------




答え >> [answers/answer\\_72.py](#)

### Q.73. 縮小と拡大

*imori.jpg*をグレースケールにしたものを0.5倍に縮小した後に2倍に拡大した画像を求めよ。この処理を行うと、ぼやけた画像ができる。

拡大縮小にはbi-linear補間を用いよ。bi-linear補間をメソッド（関数）化すると、プログラムが簡潔にできる。



入力 (imori.jpg)	出力(answers/answer_73.jpg)
	

答え >> [answers/answer\\_73.py](#)

Q.74. ピラミッド差分による高周波成分の抽出

Q.73で求めた画像と元画像の差分を求め、[0,255]に正規化せよ。

ここで求めた画像はエッジとなっている。つまり、画像中の高周波成分をとったことになる。

入力 (imori.jpg)	出力(answers/answer_74.jpg)
	








答え >> [answers/answer\\_74.py](#)

Q.75. ガウシアンピラミッド

ここでは、元画像を1/2, 1/4, 1/8, 1/16, 1/32にリサイズした画像を求めよ。

このように元画像を小さくリサイズして重ねたものを**ガウシアンピラミッド**と呼ぶ。

このガウシアンピラミッドの概念は現在でも有効であり、画像をきれいにする超解像を行うディープラーニングの手法でもガウシアンピラミッドの概念が用いられる。

入力 (imori.jpg)	1/1(answers/answer_75_1.jpg)	1/2	1/4	1/8	1/16	1/32
						

答え >> [answers/answer\\_75.py](#)

Q.76. 顕著性マップ

ここではガウシアンピラミッドを用いた簡単な顕著性マップを作成する。

顕著性マップとは画像の中で人間の目を引きやすい領域を表した画像である。

現在ではディープラーニングによる顕著性マップがよく用いられるが、本来は画像のRGB成分やHSV成分などのガウシアンピラミッドを作成し、それらの差分から求める手法がよく用いられた(例えばIttiらの手法などがある)。

ここではQ.75で作成したガウシアンピラミッドから簡単な顕著性マップを作成する。 アルゴリズムは、

1. ガウシアンピラミッドをそれぞれ、128, 64, 32, ...というサイズになっているが、はじめにそれらを128にリサイズせよ。リサイズはbi-linear補間を用いよ。
2. 作成したピラミッド（それぞれ0, 1, 2, 3, 4, 5と番号をふる）の2つを選び差分を求める。
3. 2で求めた差分を全て足し合わせ、[0, 255]に正規化せよ。

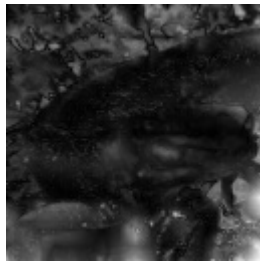
以上で顕著性マップが求められる。2で選ぶ2つの画像は特に指定はしないが、いいものを選べば解答例のように顕著性マップが作成できる。

イモリの目の部分や色が周辺と極端に違う部分など、人の目に止まりやすい領域が白くなっているのが分かる。

解答例( (0,1), (0,3), (0,5), (1,4), (2,3), (3,5) を使用)

入力 (imori.jpg)

出力(answers/answer\_76.jpg)



答え >> [answers/answer\\_76.py](#)

## Q.77. ガボールフィルタ

ガボールフィルタを実装せよ。

ガボールフィルタとは

ガウス分布と周波数変換を合わせたフィルタであり、画像の特定方向のみのエッジを抽出する時に使われる。

フィルタは次式で定義される。

$$G(y, x) = \exp(-(x'^2 + \gamma^2 y'^2) / 2 s^2) * \cos(2 \pi x' / l + p)$$

$$x' = \cos A * x + \sin A * y$$

$$y' = -\sin A * x + \cos A * y$$

$y, x$  はフィルタの位置 フィルタサイズが $K$ とすると、 $y, x$  は  $[-K/2, k/2]$  の値を取る。

$\gamma$  ... gamma ガボールフィルタの楕円率  
 $s$  ... sigma ガウス分布の標準偏差  
 $l$  ... lambda 周波数の波長

p ... 位相オフセット  
A ... フィルタの回転 抽出したい角度を指定する。

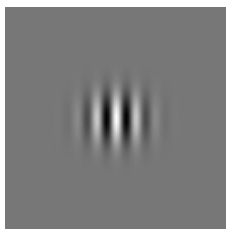
ここでは、 $K=111$ ,  $s=10$ ,  $g = 1.2$ ,  $l=10$ ,  $p=0$ ,  $A=0$ としてガボールフィルタを可視化せよ。

ガボールフィルタを実際に使う時は、フィルタ値の絶対値の和が1になるように正規化すると使いやすくなる。

答えでは可視化のためにフィルタの値を $[0,255]$ に正規化している。

**出力(answers/answer\_77.jpg)**

---



答え >> [answers/answer\\_77.py](#)

## Q.78. ガボールフィルタの回転

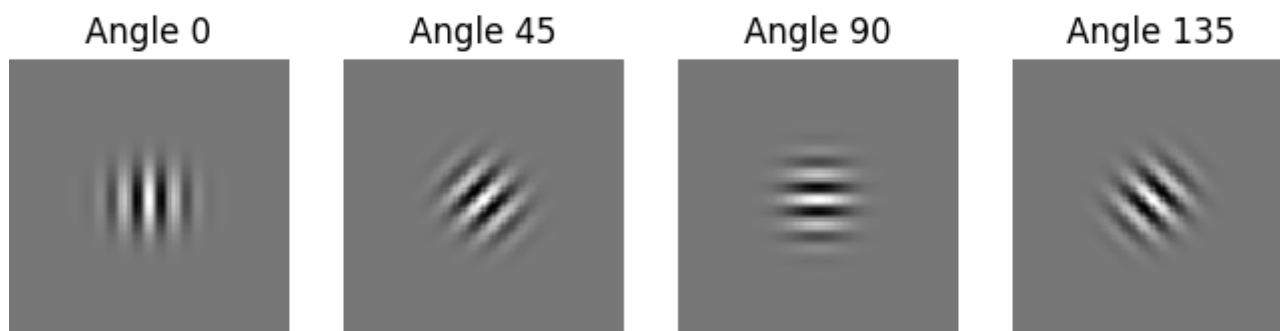
ここでは  $A=0, 45, 90, 135$ として回転方向のガボールフィルタを求めよ。その他のパラメータはQ.77同様、 $K=111$ ,  $s=10$ ,  $g = 1.2$ ,  $l=10$ ,  $p=0$ とせよ。

ここではガボールフィルタをメソッド化すれば簡単に実装できる。

**出力(answers/answer\_78.png)**

---

---

出力(answers/answer\_78.png)

答え >> [answers/answer\\_78.py](#)

## Q.79. ガボールフィルタによるエッジ抽出

*imori.jpg*をグレースケール化し、 $A=0, 45, 90, 135$  のガボールフィルタでフィルタリングせよ。

パラメータは $K=11, s=1.5, g=1.2, l=3, p=0$ とする。

解答例の様に、ガボールフィルタで指定した方向のエッジが抽出できていることが分かる。このようにガボールフィルタはエッジの特徴抽出に優れている。

ガボールフィルタは生物の視神経における脳内の一次視覚野(V1)での働きに近いとされていて、つまり生物が見ている時の目の前の画像の特徴抽出を再現しているともいわれる。

**ディープラーニング**のConvolutional層はガボールフィルタの働きに近いとも考えられている。しかし、ディープラーニングではフィルタの係数が機械学習によって自動的に決定される。機械学習の結果、ガボールフィルタに近い働きが生じると言われる。

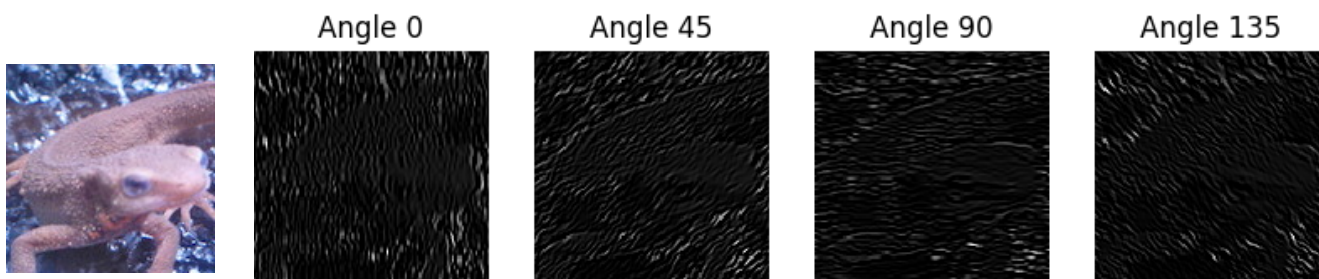
---

入力 (imori.jpg)出力(answers/answer\_79.png)

---

入力 (imori.jpg)

出力(answers/answer\_79.png)



答え >> [answers/answer\\_79.py](#)

## Q.80. ガボールフィルタによる特徴抽出

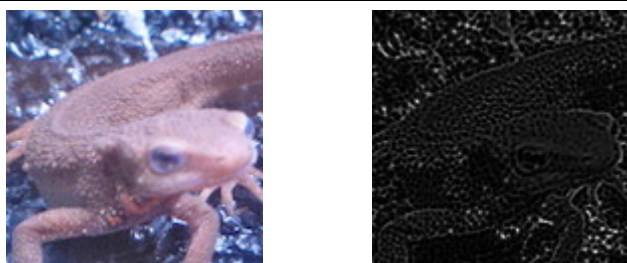
Q.79で求めた4枚の画像を足し合わせることで、画像の特徴を抽出せよ。

結果を見ると、画像の輪郭部分が白くなっていることからエッジ検出のような出力を得たように見える。

ディープラーニングのCNN(Convolutional Neural Network)では、最初に画像の特徴を抽出する働きが備わっているが、その特徴抽出の計算はこの間で行ったような操作を延々と繰り返している。ディープラーニングではこのようにして画像の特徴を自動的に抽出している。

入力 (imori.jpg)

出力(answers/answer\_80.jpg)



答え >> [answers/answer\\_80.py](#)