R02944010 李柏緯

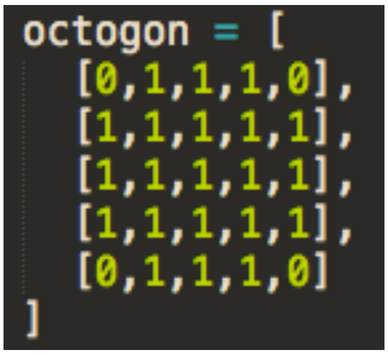
Computer Vision Homework 4

How to run it

After Install Python Image Library (Just use for I/O) :

```
python image.py lena.bmp
```

The Kernel I used:



```
octogon = [
    [0,1,1,1,0],
    [1,1,1,1,1],
    [1,1,1,1,1],
    [1,1,1,1,1],
    [0,1,1,1,0]
]
```

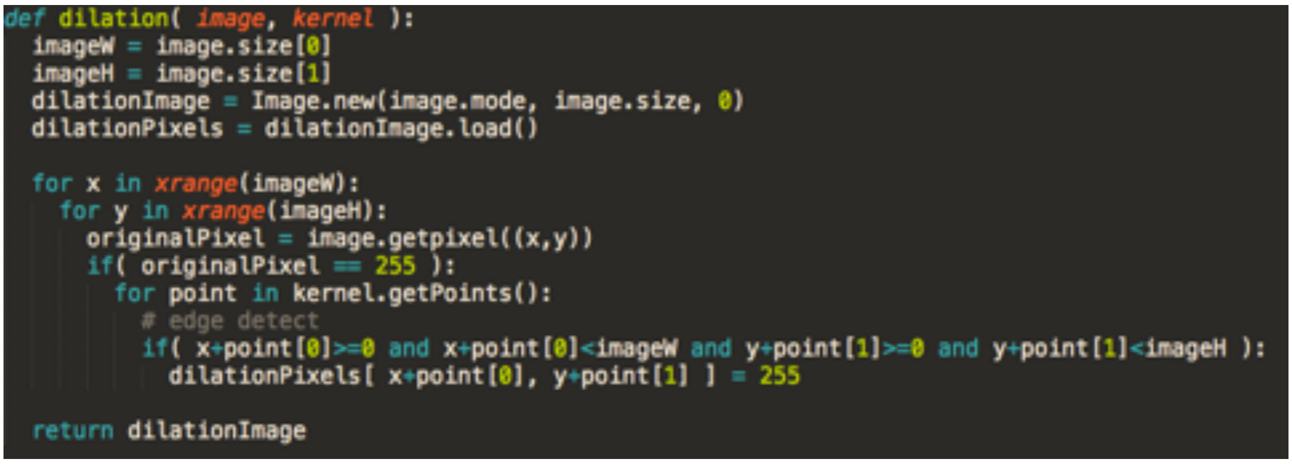
the origin is [2,2]

Dilation

Description

if one pixel of original image is 255, paste kernel pattern on it.

Principal code fragment

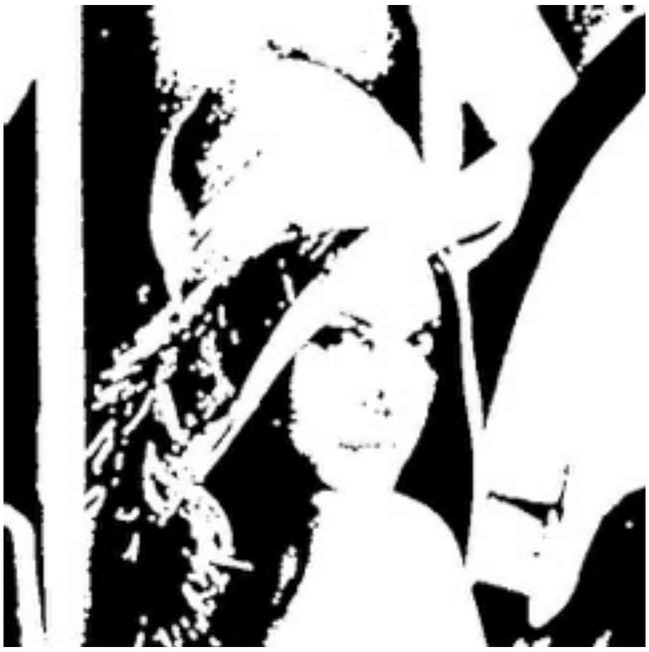


```
def dilation( image, kernel ):
  imageW = image.size[0]
  imageH = image.size[1]
  dilationImage = Image.new(image.mode, image.size, 0)
  dilationPixels = dilationImage.load()

  for x in xrange(imageW):
    for y in xrange(imageH):
      originalPixel = image.getpixel((x,y))
      if( originalPixel == 255 ):
        for point in kernel.getPoints():
          # edge detect
          if( x+point[0]>=0 and x+point[0]<imageW and y+point[1]>=0 and y+point[1]<imageH ):
            dilationPixels[ x+point[0], y+point[1] ] = 255

  return dilationImage
```

Result

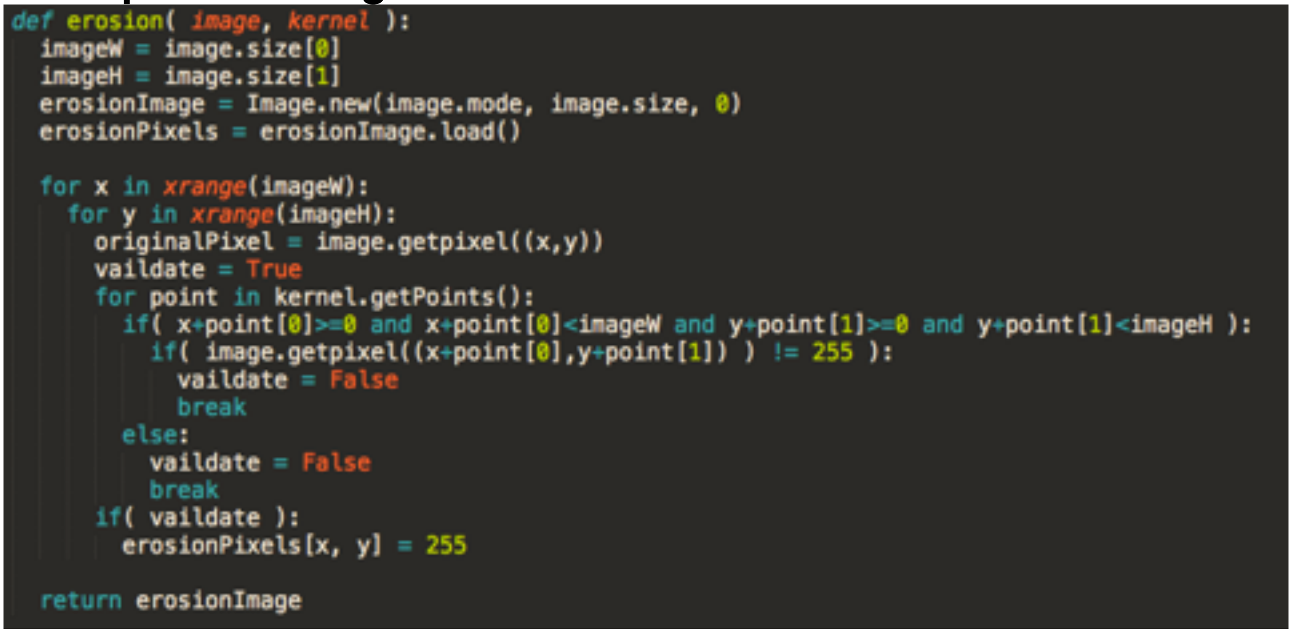


Erosion

Description

if kernel pattern fit on original image, set origin of kernel as 255

Principal code fragment

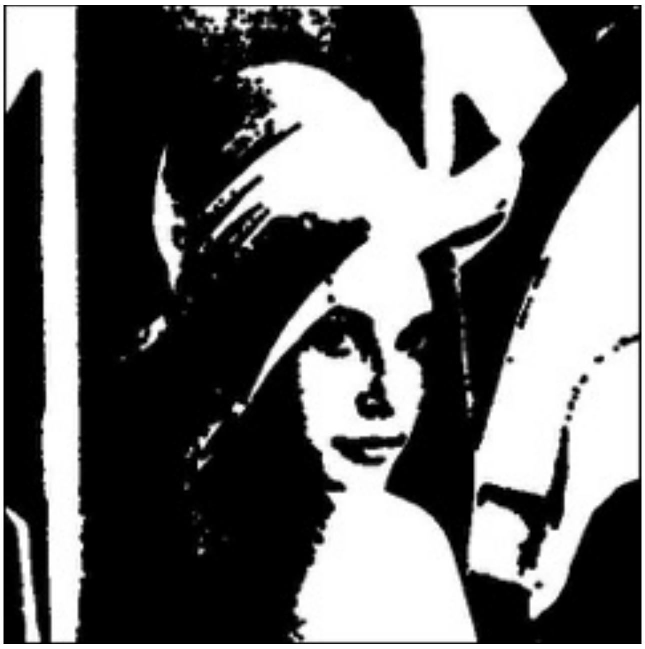


```
def erosion( image, kernel ):
  imageW = image.size[0]
  imageH = image.size[1]
  erosionImage = Image.new(image.mode, image.size, 0)
  erosionPixels = erosionImage.load()

  for x in xrange(imageW):
    for y in xrange(imageH):
      originalPixel = image.getpixel((x,y))
      vaildate = True
      for point in kernel.getPoints():
        if( x+point[0]>=0 and x+point[0]<imageW and y+point[1]>=0 and y+point[1]<imageH ):
          if( image.getpixel((x+point[0],y+point[1]) ) != 255 ):
            vaildate = False
            break
        else:
          vaildate = False
          break
      if( vaildate ):
        erosionPixels[x, y] = 255

  return erosionImage
```

Result

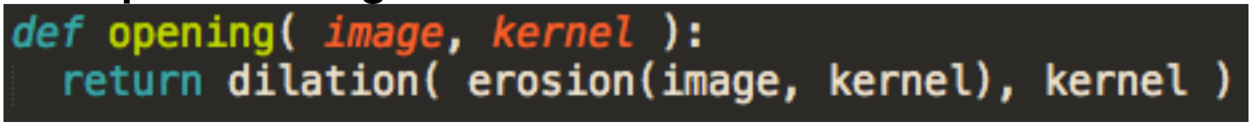


Opening

Description

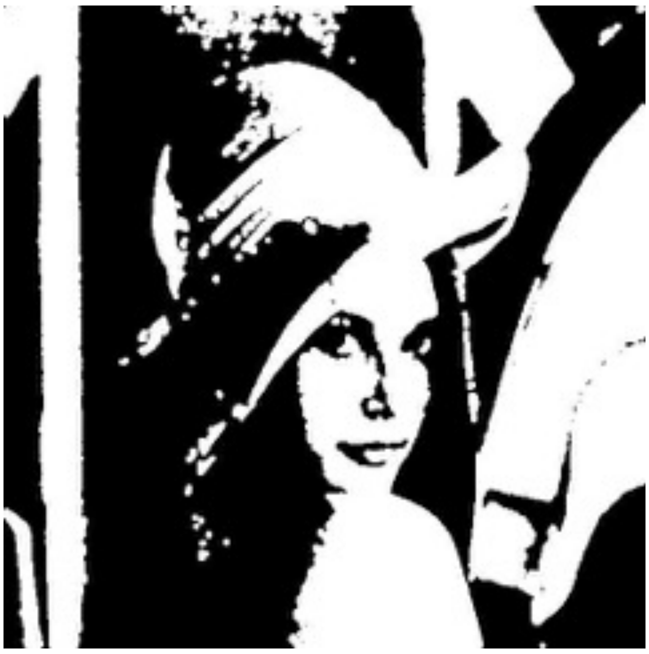
$$A \circ B = (A \ominus B) \oplus B$$

Principal code fragment



```python
def opening( image, kernel ):
    return dilation( erosion(image, kernel), kernel )
```

Result



Closing

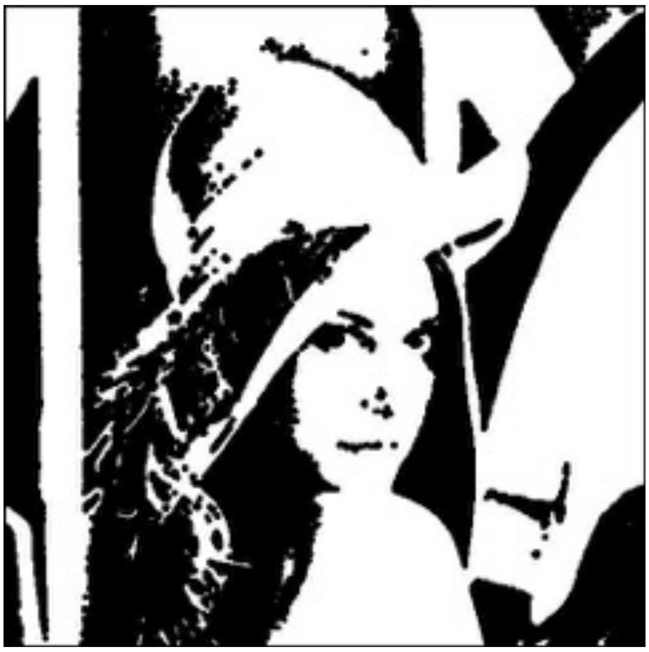
Description

$$A \bullet B = (A \oplus B) \ominus B$$

Principal code fragment

```
def closing( image, kernel ):
  return erosion( dilation(image, kernel), kernel )
```

Result



Hit and Miss

Description

$$A \odot B = (A \ominus C) \cap (A^c \ominus D)$$

Principal code fragment

```
def hitAndMiss( image, jKernel, kKernel ):
  return intersect( erosion(image, jKernel), erosion( reverse(image), kKernel ) )
```

Result

R02944010 李柏緯