R02944010 李柏緯

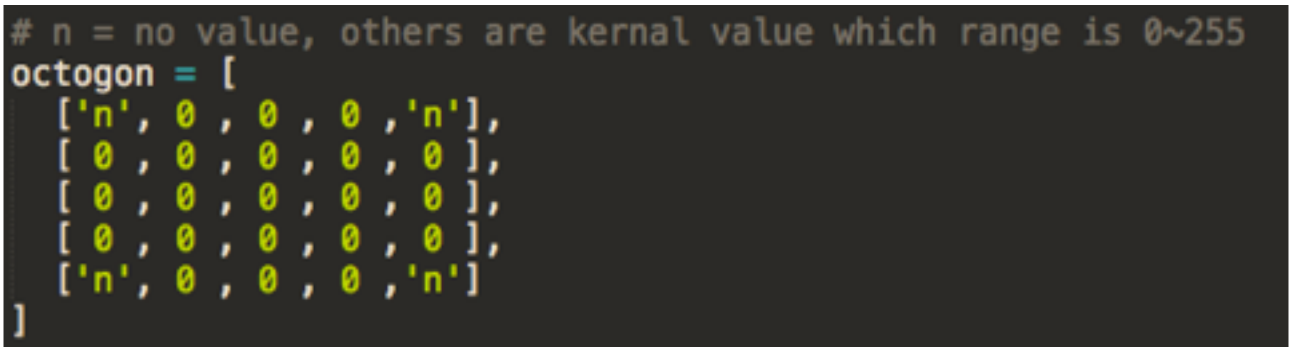
Computer Vision Homework 5

How to run it

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

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

The Kernel I used:



```
# n = no value, others are kernal value which range is 0~255
octogon = [
  ['n', 0 , 0 , 0 ,'n'],
  [ 0 , 0 , 0 , 0 , 0 ],
  [ 0 , 0 , 0 , 0 , 0 ],
  [ 0 , 0 , 0 , 0 , 0 ],
  ['n', 0 , 0 , 0 ,'n']
]
```

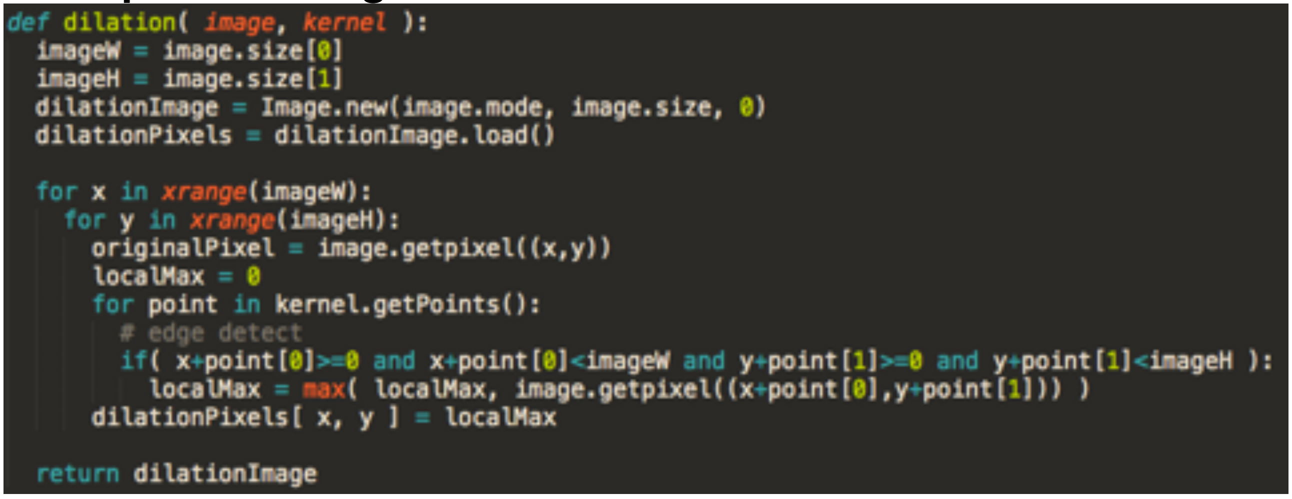
the origin is [2,2]

Dilation

Description

Find local maximum of all points in kernel area for each point in image. Then, set the point value as local maximum.

Principal code fragment

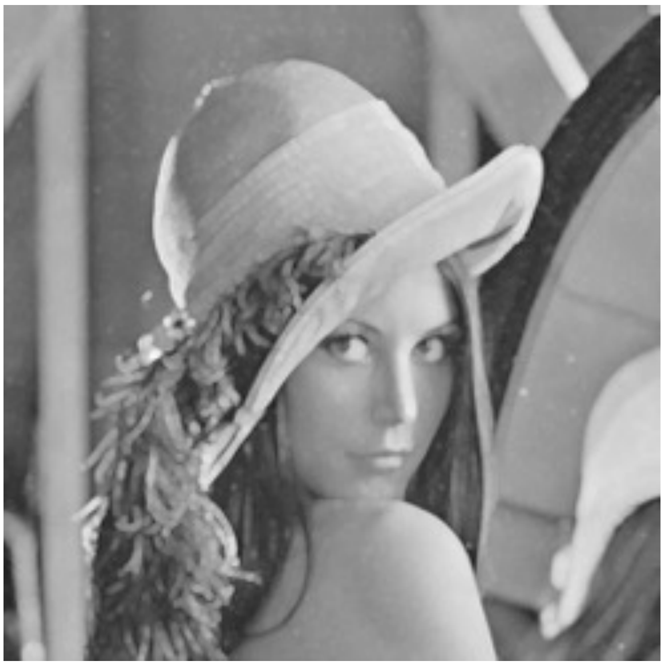


```
def dilation( image, kernel ):
  imageW = image.size[0]
  imageH = image.size[1]
  dilationImage = Image.new(image.mode, image.size, 0)
  dilationPixels = dilationImage.load()

  for x in xrange(imageW):
    for y in xrange(imageH):
      originalPixel = image.getpixel((x,y))
      localMax = 0
      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 ):
          localMax = max( localMax, image.getpixel((x+point[0],y+point[1])) )
      dilationPixels[ x, y ] = localMax

  return dilationImage
```

Result

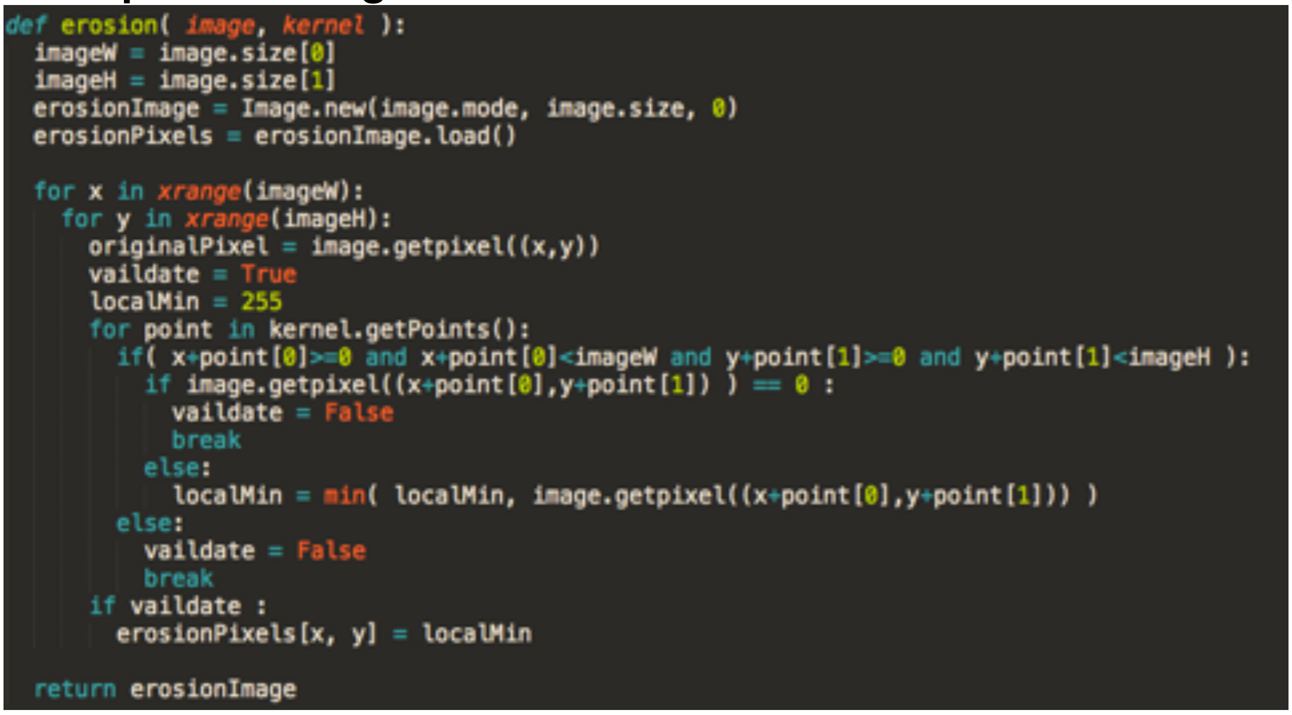


Erosion

Description

If kernel pattern fit on original image, set origin of kernel as local minimum of all points in kernel.

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
      localMin = 255
      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]) ) == 0 :
            vaildate = False
            break
          else:
            localMin = min( localMin, image.getpixel((x+point[0],y+point[1])) )
        else:
          vaildate = False
          break
      if vaildate :
        erosionPixels[x, y] = localMin

  return erosionImage
```

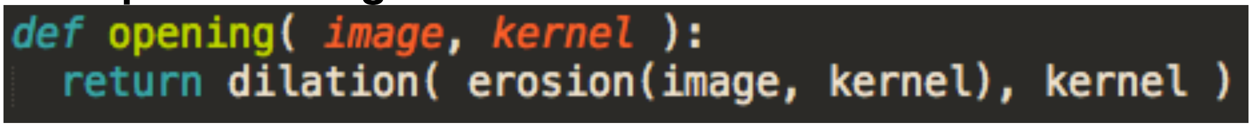
Result

Opening

Description

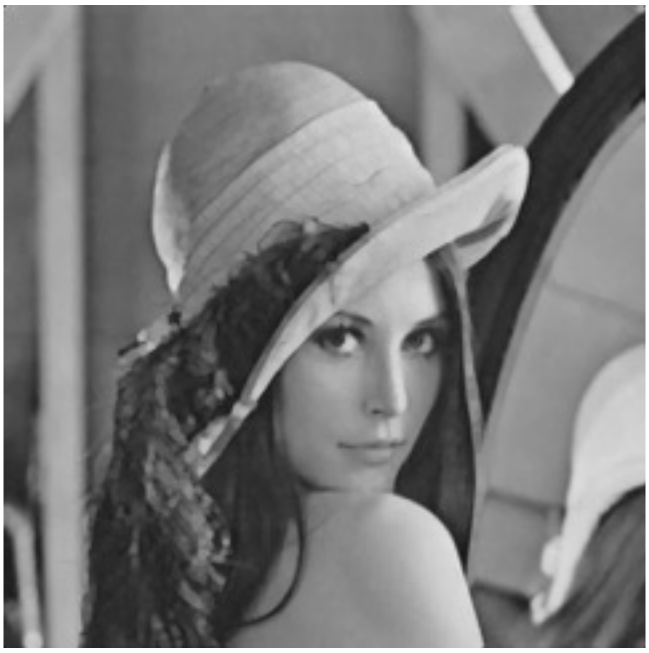
$$A \circ B = (A \ominus B) \oplus B$$

Principal code fragment



```
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