Computer vision homework 4

B04902028 資工三 洪浩翔

Original Lena image

1. binary morphological dilation:

The result after dilation

Main code about dilation:

```
#dilation
map_record = [[0 for i in range(h+4)] for j in range(w+4)]

for i in range(h):
  for j in range(w):
    if image.getpixel((i , j)) == 255:
      map_record[i+1][j] += 1 ; map_record[i+3][j] += 1 ; map_record[i+2][j] += 1
      map_record[i][j+1] += 1 ; map_record[i+1][j+1] += 1 ; map_record[i+2][j+1] += 1 ; map_record[i+3][j+1] += 1 ; map_record[i+4][j+1] += 1 ;
      map_record[i][j+2] += 1 ; map_record[i+1][j+2] += 1 ; map_record[i+2][j+2] += 1 ; map_record[i+3][j+2] += 1 ; map_record[i+4][j+2] += 1 ;
      map_record[i][j+3] += 1 ; map_record[i+1][j+3] += 1 ; map_record[i+2][j+3] += 1 ; map_record[i+3][j+3] += 1 ; map_record[i+4][j+3] += 1 ;
      map_record[i+1][j+4] += 1 ; map_record[i+3][j+4] += 1 ; map_record[i+2][j+4] += 1



for i in range(h):
  for j in range(w):
    if map_record[i+2][j+2] != 0:
      dilation.putpixel((i , j) , 255)
    else:
      dilation.putpixel((i , j) , 0)

dilation.show()
dilation.save('dilation.bmp')
```

2. binary morphological erosion:

The result after erosion

Main code about eorsion:

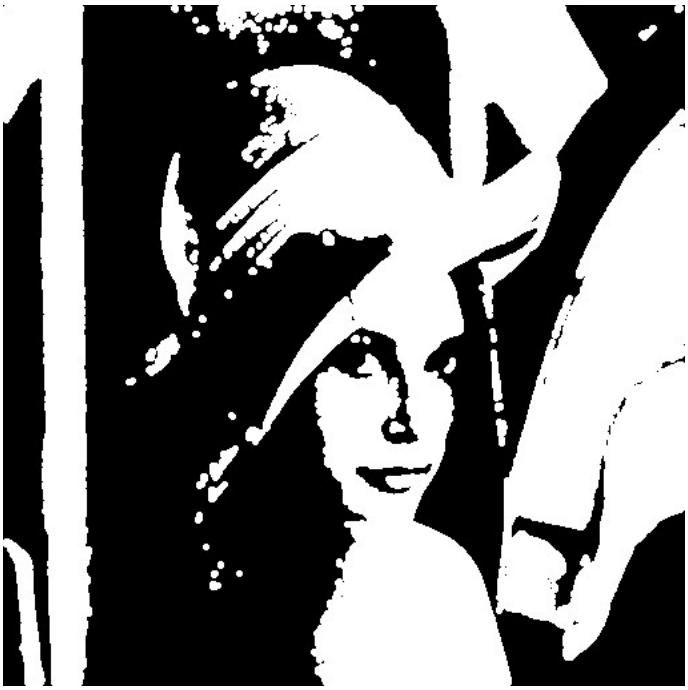
```
#erosion
map_record = [[0 for i in range(h+4)] for j in range(w+4)]

for i in range(2 , h-2):
  for j in range(2 , w-2):
    if (image.getpixel((i-1 , j-2)) == 255 and image.getpixel((i , j-2)) == 255 and image.getpixel((i+1 , j-2)) == 255 and
        image.getpixel((i-2 , j-1)) == 255 and image.getpixel((i-1 , j-1)) == 255 and image.getpixel((i , j-1)) == 255 and
        image.getpixel((i+1 , j-1)) == 255 and image.getpixel((i+2 , j-1)) == 255 and image.getpixel((i-2 , j)) == 255 and
        image.getpixel((i-1 , j)) == 255 and image.getpixel((i , j)) == 255 and image.getpixel((i+1 , j)) == 255 and
        image.getpixel((i+2 , j)) == 255 and image.getpixel((i-2 , j+1)) == 255 and image.getpixel((i-1 , j+1)) == 255 and
        image.getpixel((i , j+1)) == 255 and image.getpixel((i+1 , j+1)) == 255 and image.getpixel((i+2 , j+1)) == 255 and
        image.getpixel((i-1 , j+2)) == 255 and image.getpixel((i , j+2)) == 255 and image.getpixel((i+1 , j+2)) == 255):
      map_record[i+2][j+2] += 1


for i in range(h):
  for j in range(w):
    if map_record[i+2][j+2] != 0:
      erosion.putpixel((i , j) , 255)
    else:
      erosion.putpixel((i , j) , 0)

erosion.show()
erosion.save('erosion.bmp')
```

3. binary morphological opening:



The result after opening

Main code about opening:

```
#opening

opening = image.copy()
map_record = [[0 for i in range(h+4)] for j in range(w+4)]

for i in range(2 , h-2):
  for j in range(2 , w-2):
    if (image.getpixel((i-1 , j-2)) == 255 and image.getpixel((i , j-2)) == 255 and image.getpixel((i+1 , j-2)) == 255 and
        image.getpixel((i-2 , j-1)) == 255 and image.getpixel((i-1 , j-1)) == 255 and image.getpixel((i , j-1)) == 255 and
        image.getpixel((i+1 , j-1)) == 255 and image.getpixel((i+2 , j-1)) == 255 and image.getpixel((i-2 , j)) == 255 and
        image.getpixel((i-1 , j)) == 255 and image.getpixel((i , j)) == 255 and image.getpixel((i+1 , j)) == 255 and
        image.getpixel((i+2 , j)) == 255 and image.getpixel((i-2 , j+1)) == 255 and image.getpixel((i-1 , j+1)) == 255 and
        image.getpixel((i , j+1)) == 255 and image.getpixel((i+1 , j+1)) == 255 and image.getpixel((i+2 , j+1)) == 255 and
        image.getpixel((i-1 , j+2)) == 255 and image.getpixel((i , j+2)) == 255 and image.getpixel((i+1 , j+2)) == 255):
      map_record[i+2][j+2] += 1


for i in range(h):
  for j in range(w):
    if map_record[i+2][j+2] != 0:
      erosion.putpixel((i , j) , 255)
    else:
      erosion.putpixel((i , j) , 0)
```

```
map_record = [[0 for i in range(h+4)] for j in range(w+4)]

for i in range(h):
  for j in range(w):
    if erosion.getpixel((i , j)) == 255:
      map_record[i+1][j] += 1 ; map_record[i+3][j] += 1 ; map_record[i+2][j] += 1
      map_record[i][j+1] += 1 ; map_record[i+1][j+1] += 1 ; map_record[i+2][j+1] += 1 ; map_record[i+3][j+1] += 1 ; map_record[i+4][j+1] += 1 ;
      map_record[i][j+2] += 1 ; map_record[i+1][j+2] += 1 ; map_record[i+2][j+2] += 1 ; map_record[i+3][j+2] += 1 ; map_record[i+4][j+2] += 1 ;
      map_record[i][j+3] += 1 ; map_record[i+1][j+3] += 1 ; map_record[i+2][j+3] += 1 ; map_record[i+3][j+3] += 1 ; map_record[i+4][j+3] += 1 ;
      map_record[i+1][j+4] += 1 ; map_record[i+3][j+4] += 1 ; map_record[i+2][j+4] += 1



for i in range(h):
  for j in range(w):
    if map_record[i+2][j+2] != 0:
      opening.putpixel((i , j) , 255)
    else:
      opening.putpixel((i , j) , 0)

opening.show()
opening.save('opening.bmp')
```

4. binary morphological closing:

The result after closing

Main code about closing:

```
#closing
map_record = [[0 for i in range(h+4)] for j in range(w+4)]

for i in range(h):
  for j in range(w):
    if image.getpixel((i , j)) == 255:
      map_record[i+1][j] += 1 ; map_record[i+3][j] += 1 ; map_record[i+2][j] += 1
      map_record[i][j+1] += 1 ; map_record[i+1][j+1] += 1 ; map_record[i+2][j+1] += 1 ; map_record[i+3][j+1] += 1 ; map_record[i+4][j+1] += 1 ;
      map_record[i][j+2] += 1 ; map_record[i+1][j+2] += 1 ; map_record[i+2][j+2] += 1 ; map_record[i+3][j+2] += 1 ; map_record[i+4][j+2] += 1 ;
      map_record[i][j+3] += 1 ; map_record[i+1][j+3] += 1 ; map_record[i+2][j+3] += 1 ; map_record[i+3][j+3] += 1 ; map_record[i+4][j+3] += 1 ;
      map_record[i+1][j+4] += 1 ; map_record[i+3][j+4] += 1 ; map_record[i+2][j+4] += 1

for i in range(h):
  for j in range(w):
    if map_record[i+2][j+2] != 0:
      dilation.putpixel((i , j) , 255)
    else:
      dilation.putpixel((i , j) , 0)
```

```
map_record = [[0 for i in range(h+4)] for j in range(w+4)]

for i in range(2 , h-2):
  for j in range(2 , w-2):
    if (dilation.getpixel((i-1 , j-2)) == 255 and dilation.getpixel((i , j-2)) == 255 and dilation.getpixel((i+1 , j-2)) == 255 and
        dilation.getpixel((i-2 , j-1)) == 255 and dilation.getpixel((i-1 , j-1)) == 255 and dilation.getpixel((i , j-1)) == 255 and
        dilation.getpixel((i+1 , j-1)) == 255 and dilation.getpixel((i+2 , j-1)) == 255 and dilation.getpixel((i-2 , j)) == 255 and
        dilation.getpixel((i-1 , j)) == 255 and dilation.getpixel((i , j)) == 255 and dilation.getpixel((i+1 , j)) == 255 and
        dilation.getpixel((i+2 , j)) == 255 and dilation.getpixel((i-2 , j+1)) == 255 and dilation.getpixel((i-1 , j+1)) == 255 and
        dilation.getpixel((i , j+1)) == 255 and dilation.getpixel((i+1 , j+1)) == 255 and dilation.getpixel((i+2 , j+1)) == 255 and
        dilation.getpixel((i-1 , j+2)) == 255 and dilation.getpixel((i , j+2)) == 255 and dilation.getpixel((i+1 , j+2)) == 255):
      map_record[i+2][j+2] += 1


for i in range(h):
  for j in range(w):
    if map_record[i+2][j+2] != 0:
      closing.putpixel((i , j) , 255)
    else:
      closing.putpixel((i , j) , 0)

closing.show()
closing.save('closing.bmp')
```

5. binary morphological hit and miss:

The result after hit and miss

Main code about hit and miss:

```
#hit and miss
for i in range(h):
  for j in range(w):
    if image.getpixel((i , j)) == 255:
      inverse.putpixel((i , j) , 0)
    else:
      inverse.putpixel((i , j) , 255)

map_record = [[0 for i in range(h+2)] for j in range(w+2)]

for i in range(1 , h):
  for j in range(0 , w-1):
    if image.getpixel((i , j)) == 255 and image.getpixel((i-1 , j)) == 255 and image.getpixel((i , j+1)) == 255:
      map_record[i+1][j+1] += 1

for i in range(h):
  for j in range(w):
    if map_record[i+1][j+1] != 0:
      erosion.putpixel((i , j) , 255)
    else:
      erosion.putpixel((i , j) , 0)
```

```
map_record = [[0 for i in range(h+2)] for j in range(w+2)]

for i in range(0 , h-1):
  for j in range(1 , w):
    if inverse.getpixel((i+1 , j)) == 255 and inverse.getpixel((i , j-1)) == 255 and inverse.getpixel((i+1 , j-1)) == 255:
      map_record[i+1][j+1] += 1

for i in range(h):
  for j in range(w):
    if map_record[i+1][j+1] != 0:
      inverse.putpixel((i , j) , 255)
    else:
      inverse.putpixel((i , j) , 0)

for i in range(h):
  for j in range(w):
    if inverse.getpixel((i , j)) == erosion.getpixel((i , j)):
      hit_and_miss.putpixel((i , j) , erosion.getpixel((i , j)))
    else:
      hit_and_miss.putpixel((i , j) , 0)
hit_and_miss.save('hit_and_miss.bmp')
hit_and_miss.show()
```