Computer Vision Hw 5

R07921010 余奇安

Input image: grayscale Lena.jpg

1.Dilation

Use 35553 kernel. Every pixel value is the max value in kernel.

Image:

Code:
Define dilation function and dil_kernel
Create new np array with another 4 dimension in row and column respectively to deal with boundary detection.
Eventually, reduce the dimension to original image (res) .

```
def dilation(img):
    row=img.shape[0]
    col=img.shape[1]
    new=np.zeros((row+4,col+4),dtype=np.int)
    res=np.zeros((row,col),dtype=np.int)

    for i in range(row):
        for j in range(col):
            new[i+2][j+2]=img[i][j]

    for i in range(row):
        for j in range(col):
            res[i][j]=dil_kernal(new,i+2,j+2)
    return res
```

```
def dil_kernal(img,row,col):
    value=[]
    for i in range(-2,3):
        for j in range(-2,3):
            if (i==-2 and j==-2) or (i==-2 and j==2)or(i==2 and j==-2)or(i==2 and j==2):
                pass
            else:
                value.append(img[row+i][col+j])
    return max(value)
```

dil_kernel function is design to find the max value in kernel.

2.Erosion

Use 35553 kernel. Every pixel value is the min value in kernel.

Code:

```
def erosion(img):
    row=img.shape[0]
    col=img.shape[1]
    new=np.zeros((row+4,col+4),dtype=np.int)
    res=np.zeros((row,col),dtype=np.int)
    for i in range(row):
        for j in range(col):
            new[i+2][j+2]=img[i][j]

    for i in range(row):
        for j in range(col):
                res[i][j]=ero_kernal(new,i+2,j+2)
    return res
```

```
def ero_kernal(img,row,col):
    value=[]
    for i in range(-2,3):
        for j in range(-2,3):
            if (i==-2 and j==-2) or (i==-2 and j==2)or(i==2 and j==-2)or(i==2 and j==2):
                pass
            else:
                value.append(img[row+i][col+j])

    return min(value)
```

use ero_kernel to find the min value in kernel.
The meaning of variable is the same as dilation function.

3.Opening

Opening operation means that do the erosion first and do the dilation.
After gray scale opening, the image becomes more smooth and vague, the dark pixel is reinforced.

Code:

```
def opening(img):
    ero_im=erosion(img)
    open_im=dilation(ero_im)
    return open_im
```

Based on the erosion and dilation function, we can do opening easily.

4.Closing

Code:

```
def closing(img):
    dil_im=dilation(img)
    close_im=erosion(dil_im)
    return close_im
```

Closing operation means that do the dilation first and do the erosion.
After gray scale closing, the image becomes more smooth and vague, the bright pixel is reinforced.