Homework 5

R13525009 羅筠笙

1. Dilation

Description: For each pixel, it checks a local neighborhood defined by the kernel and assigns the maximum pixel value found in that neighborhood to the current pixel position.



1. def dilation(img, kernel):
2. kernel\_size = len(kernel)
3. dilation\_img = np.zeros\_like(img)
4. for i in range(img\_size0):
5. for j in range(img\_size1):
6. max\_value = 0
7. for ki in range(kernel\_size):
8. for kj in range(kernel\_size):
9. ni, nj = i + ki - kernel\_size // 2, j + kj - kernel\_size // 2
10. if 0 <= ni < img\_size0 and 0 <= nj < img\_size1:
11. if kernel[ki][kj] == 1:
12. pixel\_value = img[ni][nj]
13. if pixel\_value > max\_value:
14. max\_value = pixel\_value
15. dilation\_img[i, j] = max\_value
16. return dilation\_img.astype(np.uint8)
17. Erosion

Description: For each pixel, it searches for the minimum pixel value within the neighborhood specified by the kernel, assigning that minimum value to the current position.



1. def erosion(img, kernel):
2. kernel\_size = len(kernel)
3. erosion\_img = np.zeros\_like(img)
4. for i in range(img\_size0):
5. for j in range(img\_size1):
6. min\_value = 255
7. for ki in range(kernel\_size):
8. for kj in range(kernel\_size):
9. ni, nj = i + ki - kernel\_size // 2, j + kj - kernel\_size // 2
10. if 0 <= ni < img\_size0 and 0 <= nj < img\_size1:
11. if kernel[ki][kj] == 1:
12. pixel\_value = img[ni][nj]
13. if pixel\_value < min\_value:
14. min\_value = pixel\_value
15. erosion\_img[i, j] = min\_value
16. return erosion\_img.astype(np.uint8)
17. Opening

Description: First use erosion, then use dilation.



1. def opening(img, kernel):
2. return (dilation(erosion(img, kernel), kernel))
3. Closing

Description: First use dilation, then use erosion.



1. def closing(img, kernel):
2. return (erosion(dilation(img, kernel), kernel))