Computer Vision Homework #5

資工四 b05902115 陳建丞

1. Dilation

- **Result**

- **Implementation**
  - $(f \oplus k)(x,\ y) = max\{f(x-i,\ y-j) + k(i,\ j) \mid (i,\ j) \in K,\ (x-i,\ y-j) \in F\}$
  - **Kernel: Octagonal 3-5-5-5-3 kernel**

  According to the formula above, traverse every pixel of the input image. Find the local maximum with the kernel applied on each pixel, and the local maximum is the value of the corresponding pixel values for the output image.

```
def Dilation(img, kernel):

  r = int((kernel.shape[0]-1)/2)

  new_img = img.copy()
  height, width = img.shape[:2]

  for i in range(height):
```

```
    for j in range(width):
      local_max = 0
      for h in range(-r, r+1):
        for w in range(-r, r+1):
          if (i-h) < 0 or (i-h) >= height or (j-w) < 0 or (j-w) >= width:
            continue
          if kernel[h+r][w+r] == 0:
            continue

          if img[i-h][j-w] > local_max:
            local_max = img[i-h][j-w]
      new_img[i][j] = local_max

  return new_img
```

2. Erosion

- **Result**

- **Implementation**
  - $(f \ominus k)(x,\ y) = min\{f(x+i,\ y+j) + k(i,\ j) \mid (i,\ j) \in K,\ (x+i,\ y+j) \in F\}$
  - **Kernel: Octagonal 3-5-5-5-3 kernel**

  According to the formula above, traverse every pixel of the input image. Find the local minimum with the kernel applied on each pixel, and the local minimum is the value of the corresponding pixel values for the output image.

```
def Erosion(img, kernel):

  r = int((kernel.shape[0]-1)/2)

  new_img = img.copy()
  height, width = img.shape[:2]

  for i in range(height):
    for j in range(width):
      local_min = 255
      for h in range(-r, r+1):
        for w in range(-r, r+1):
          if (i+h) < 0 or (i+h) >= height or (j+w) < 0 or (j+w) >= width:
            continue
          if kernel[h+r][w+r] == 0:
            continue

          if img[i+h][j+w] < local_min:
            local_min = img[i+h][j+w]
      new_img[i][j] = local_min

        return new_img
```

3. Opening

- **Result**

- **Implementation**
  - $B \circ K = (B \ominus K) \oplus K$

  Simply apply the formula with the erosion and dilation function above.

```
def Opening(img, kernel):
  return(Dilation(Erosion(img, kernel), kernel))
```

4. Closing

- **Result**

- **Implementation**
  - $B \bullet K = (B \oplus K) \ominus K$

  Simply apply the formula with the erosion and dilation function above.

```
def Closing(img, kernel):
  return(Erosion(Dilation(img, kernel), kernel))
```

Note

- **Python package**

- **skimage** : read and write image
- **numpy** : array manipulation