電腦視覺 HW9

R10922176 陳冠穎

Roberts operator (threshold: 30)

說明:

```python
def Roberts(img, threshold=30):
    new_img = np.zeros((img.shape[0], img.shape[1]))
    eximg = cv2.copyMakeBorder(img, 0, 1, 0, 1, cv2.BORDER_REPLICATE)

    for r in range(img.shape[0]):
        for c in range(img.shape[1]):
            r1 = -1 * eximg[r][c] + eximg[r + 1][c + 1]
            r2 = -1 * eximg[r][c + 1] + eximg[r + 1][c]
            gradient = math.sqrt(r1 ** 2 + r2 ** 2)
            if (gradient >= threshold):
                new_img[r][c] = 0
            else:
                new_img[r][c] = 255

    return new_img
```

Prewitt edge detector (threshold: 24)

說明:

```python
def GetGradient(img, r, c, mask1, mask2):
    offset = int(len(mask1) / 2)
    p1 = 0
    p2 = 0
    for i in range(-offset, offset + 1):
        for j in range(-offset, offset + 1):
            p1 += mask1[i + offset][j + offset] * img[r + i][c + j]
            p2 += mask2[i + offset][j + offset] * img[r + i][c + j]

    return math.sqrt(p1 ** 2 + p2 ** 2)
```

```python
def Prewitt(img, threshold):
    new_img = np.zeros((img.shape[0], img.shape[1]))
    eximg = cv2.copyMakeBorder(img, 1, 1, 1, 1, cv2.BORDER_REPLICATE)
    m1 = [[-1, -1, -1],
          [0, 0, 0],
          [1, 1, 1]]
    m2 = [[-1, 0, 1],
          [-1, 0, 1],
          [-1, 0, 1]]

    for r in range(1, img.shape[0] + 1):
        for c in range(1, img.shape[1] + 1):
            gradient = GetGradient(eximg, r, c, m1, m2)
            if (gradient >= threshold):
                new_img[r - 1][c - 1] = 0
            else:
                new_img[r - 1][c - 1] = 255

    return new_img
```

Sobel edge detector (threshold: 38)

說明:

```python
def Soble(img, threshold):
    new_img = np.zeros((img.shape[0], img.shape[1]))
    eximg = cv2.copyMakeBorder(img, 1, 1, 1, 1, cv2.BORDER_REPLICATE)
    m1 = [[-1, -2, -1],
          [0, 0, 0],
          [1, 2, 1]]
    m2 = [[-1, 0, 1],
          [-2, 0, 2],
          [-1, 0, 1]]

    for r in range(1, img.shape[0] + 1):
        for c in range(1, img.shape[1] + 1):
            gradient = GetGradient(eximg, r, c, m1, m2)
            if (gradient >= threshold):
                new_img[r - 1][c - 1] = 0
            else:
                new_img[r - 1][c - 1] = 255

    return new_img
```

Frei and Chen gradient operator (threshold: 30)

說明:

```
def FreiAndChen(img, threshold):
    new_img = np.zeros((img.shape[0], img.shape[1]))
    eximg = cv2.copyMakeBorder(img, 1, 1, 1, 1, cv2.BORDER_REPLICATE)
    m1 = [[-1, -math.sqrt(2), -1],
          [0, 0, 0],
          [1, math.sqrt(2), 1]]
    m2 = [[-1, 0, 1],
          [-math.sqrt(2), 0, math.sqrt(2)],
          [-1, 0, 1]]

    for r in range(1, img.shape[0] + 1):
        for c in range(1, img.shape[1] + 1):
            gradient = GetGradient(eximg, r, c, m1, m2)
            if (gradient >= threshold):
                new_img[r - 1][c - 1] = 0
            else:
                new_img[r - 1][c - 1] = 255

    return new_img
```

Kirsch compass operator (threshold: 135)

說明:

GetMaxK 為搜尋所有要判斷的 K 並找出最大值。

```
def Kirsch(img, threshold):
    new_img = np.zeros((img.shape[0], img.shape[1]))
    eximg = cv2.copyMakeBorder(img, 1, 1, 1, 1, cv2.BORDER_REPLICATE)

    for r in range(1, img.shape[0] + 1):
        for c in range(1, img.shape[1] + 1):
            gradient = GetMaxK(eximg, r, c)
            if (gradient >= threshold):
                new_img[r - 1][c - 1] = 0
            else:
                new_img[r - 1][c - 1] = 255

    return new_img
```

Robinson compass operator (threshold: 43)

說明:

GetMaxR 為搜尋所有要判斷的 R 並找出最大值。

```
def Robinsin(img, threshold):
    new_img = np.zeros((img.shape[0], img.shape[1]))
    eximg = cv2.copyMakeBorder(img, 1, 1, 1, 1, cv2.BORDER_REPLICATE)

    for r in range(1, img.shape[0] + 1):
        for c in range(1, img.shape[1] + 1):
            gradient = GetMaxR(eximg, r, c)
            if (gradient >= threshold):
                new_img[r - 1][c - 1] = 0
            else:
                new_img[r - 1][c - 1] = 255

    return new_img
```

Nevatia Babu 5 × 5 operator (threshold: 12500)

說明:

GetMaxN 搜尋所有要判斷的 N 找出最大值。

```
def Nevatia(img, threshold):
    new_img = np.zeros((img.shape[0], img.shape[1]))
    eximg = cv2.copyMakeBorder(img, 2, 2, 2, 2, cv2.BORDER_REPLICATE)

    for r in range(2, img.shape[0] + 2):
        for c in range(2, img.shape[1] + 2):
            gradient = GetMaxN(eximg, r, c)
            if (gradient >= threshold):
                new_img[r - 2][c - 2] = 0
            else:
                new_img[r - 2][c - 2] = 255

    return new_img
```