機器視覺

HW 5

資工三 109590049 林敬翰

1. Sobel Edge Detection

```
void SobelEdgeDetection(Mat img, string str) {
    int row = img.rows;
    int col = img.cols;
    Mat gray = ConvertToGray(img);
    Mat Sobel_Vertical(row, col, CV_8UC1);
    Mat Sobel_Horizontal(row, col, CV_8UC1);
    Mat Sobel(row, col, CV_8UC1);
    int kernel_vertical[3][3] = {
        {-1, -2, -1},
        { 0,  0,  0},
        { 1,  2,  1}
    };
    int kernel_Horizontal[3][3] = {
        {-1,  0,  1},
        {-2,  0,  2},
        {-1,  0,  1}
    };
    //calculate
    for (int i = 1; i < row - 1; i++) {
        for (int j = 1; j < col - 1; j++) {
            int sum = 0;
            int sum2 = 0;
            for (int k = -1; k <= 1; k++) {
                for (int l = -1; l <= 1; l++) {
                    sum += kernel_vertical[k + 1][l + 1] * gray.at<uchar>(i + k, j + l);
                    sum2 += kernel_Horizontal[k + 1][l + 1] * gray.at<uchar>(i + k, j + l);
                }
            }
            Sobel_Vertical.at<uchar>(i, j) = saturate_cast<uchar>(sum);
            Sobel_Horizontal.at<uchar>(i, j) = saturate_cast<uchar>(sum2);
        }
    }
    add(abs(Sobel_Vertical), abs(Sobel_Horizontal), Sobel);
    imwrite("output/" + str + "/SobelEdgeDetection/" + str + "_SobelVertical.png",
Sobel_Vertical);
    imwrite("output/" + str + "/SobelEdgeDetection/" + str + "_SobelHorizontal.png",
```

```
Sobel_Horizontal);
    imwrite("output/" + str + "/SobelEdgeDetection/" + str + "_Sobel.png", Sobel);
}
```

先將圖片轉成灰階後，然後設定好 vertical 和 horizontal 的 kernel，之後跑灰階圖的每個 pixel 將 pixel 鄰近 9 點的值乘以 kernel 後相加，最後加上 saturate_cast 來防止數值溢出，最後得出 vertical 和 horizontal 後用 add 函式將兩個 mat 相加起來得出整張圖。

2. Prewitt Edge Detection

```
void PrewittEdgeDetection(Mat img, string str) {
    int row = img.rows;
    int col = img.cols;
    Mat gray = ConvertToGray(img);
    int kernel_vertical[3][3] = {
        {-1, -1, -1},
        { 0,  0,  0},
        { 1,  1,  1}
    };
    int kernel_Horizontal[3][3] = {
        {-1,  0,  1},
        {-1,  0,  1},
        {-1,  0,  1}
    };
    Mat Prewitt_Vertical(row, col, CV_8UC1);
    Mat Prewitt_Horizontal(row, col, CV_8UC1);
    Mat Prewitt(row, col, CV_8UC1);
    //calculate
    for (int i = 1; i < row - 1; i++) {
        for (int j = 1; j < col - 1; j++) {
            int sum = 0;
            int sum2 = 0;
            for (int k = -1; k <= 1; k++) {
                for (int l = -1; l <= 1; l++) {
                    sum += kernel_vertical[k + 1][l + 1] * gray.at<uchar>(i + k, j + l);
```

```
                    sum2 += kernel_Horizontal[k + 1][l + 1] * gray.at<uchar>(i + k, j + l);
                }
            }
            Prewitt_Vertical.at<uchar>(i, j) = saturate_cast<uchar>(sum);
            Prewitt_Horizontal.at<uchar>(i, j) = saturate_cast<uchar>(sum2);
        }
    }
    add(abs(Prewitt_Vertical), abs(Prewitt_Horizontal), Prewitt);
    imwrite("output/" + str + "/PrewittEdgeDetection/" + str + "_PrewittVertical.png",
Prewitt_Vertical);
    imwrite("output/" + str + "/PrewittEdgeDetection/" + str + "_PrewittHorizontal.png",
Prewitt_Horizontal);
    imwrite("output/" + str + "/PrewittEdgeDetection/" + str + "_Prewitt.png", Prewitt);
}
```

Prewitt 的做法跟做 Sobel 時類似，只是 kernel 的值不一樣。

3. Laplacian Edge Detection

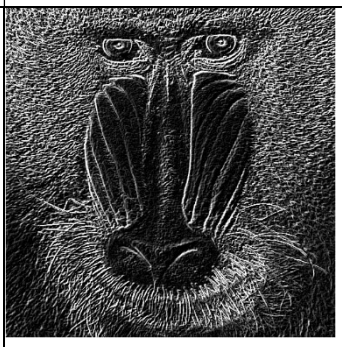
```
void LaplacianEdgeDetection(Mat img, string str) {
    int row = img.rows;
    int col = img.cols;
    Mat gray = ConvertToGray(img);
    int kernel[3][3] = {
        {0,  1,  0},
        {1, -4,  1},
        {0,  1,  0}
    };
    int kernel2[3][3] = {
        {1,  1,  1},
        {1, -8,  1},
        {1,  1,  1}
    };
    Mat laplacian4(row, col, CV_8UC1);
    Mat laplacian8(row, col, CV_8UC1);
    //calculate
    for (int i = 1; i < row - 1; i++) {
        for (int j = 1; j < col - 1; j++) {
```

```
            int sum = 0;
            int sum2 = 0;
            for (int k = -1; k <= 1; k++) {
                for (int l = -1; l <= 1; l++) {
                    sum += kernel[k + 1][l + 1] * gray.at<uchar>(i + k, j + l);
                    sum2 += kernel2[k + 1][l + 1] * gray.at<uchar>(i + k, j + l);
                }
            }
            laplacian4.at<uchar>(i, j) = saturate_cast<uchar>(sum);
            laplacian8.at<uchar>(i, j) = saturate_cast<uchar>(sum2);
        }
    }
    imwrite("output/" + str + "/LaplacianEdgeDetection/" + str + "_Laplacian4.png",
laplacian4);
    imwrite("output/" + str + "/LaplacianEdgeDetection/" + str + "_Laplacian8.png",
laplacian8);
}
```

Laplacian 的做法跟前面兩個一樣，只是 kernel 的值不一樣。

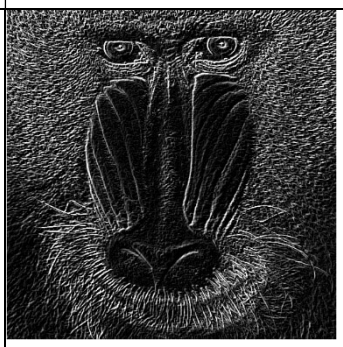
輸出

1. Sobel Edge Detection

	Vertical	Horizontal	Sobel
House512			
Lena			
Mandrill			

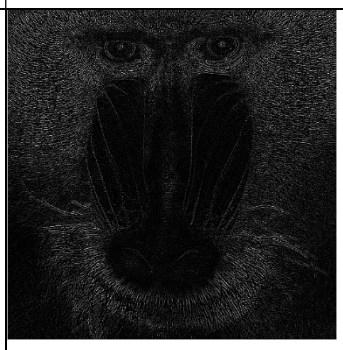
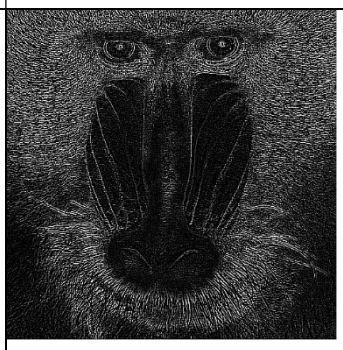
2. Prewitt Edge Detection

	Vertical	Horizontal	Sobel

House512			
Lena			
Mandrill			

3.Laplacian Edge Detection

	-4	-8

House512		
Lena		
Mandrill		

我做完之後發現 sobel 得出來 edge 的會比 prewitt 還要亮一些, 在細節上比較難看出差異，但是用 Laplacian 的話，細節感覺是更加的明顯。