**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。

1. **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。

1. **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。