「彩色影像處理」期末作業

一、 題目

自行選用十二幅以上交通號誌的影像，撰寫能夠根據輸入影像(query image)自動辨識的程式。

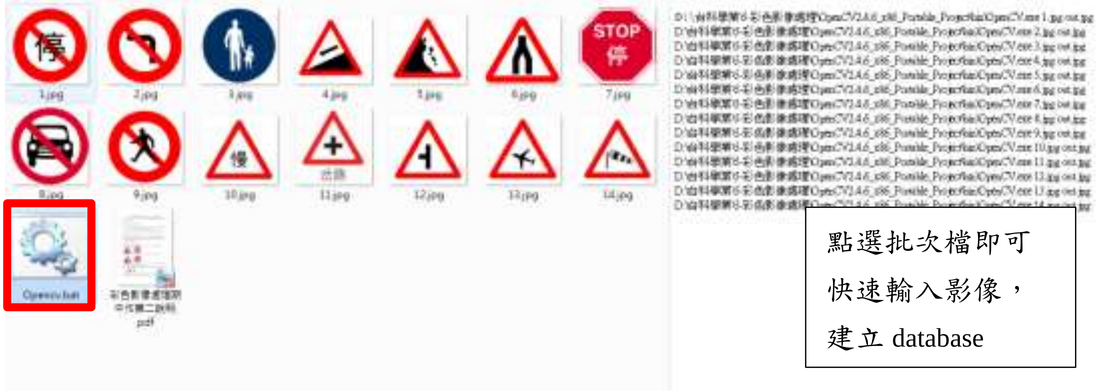
二、 程式註解

建database程式:

```
#include <cv.h>
#include <highgui.h>
#include <stdio.h>
#include <math.h>
#include <iostream>
#include <stdlib.h>
#include <fstream>
using namespace cv;
using namespace std;

int main(int argc, char *argv[])
{
        char *input = argv[1]; //讀入輸入檔名
        char *output = argv[2]; //讀入輸出檔名
        if (argc != 3)
        {
                cerr << "<exe> <input-file> <out-file>" << endl;
                return EXIT_FAILURE;
        }
        Mat image = imread(input, CV_LOAD_IMAGE_COLOR);//讀入照片
        ofstream PupartionInfo;
        PupartionInfo.open("image_data.txt", ios::app); //開啟一個txt檔，便於將參數書出建data base
        int cols = image.cols;//width
        int rows = image.rows;//height
        double meanR, meanG, meanB;
        int i, j, m, l;
        int part = 8; //將影像切割列跟行分別成8等份
```

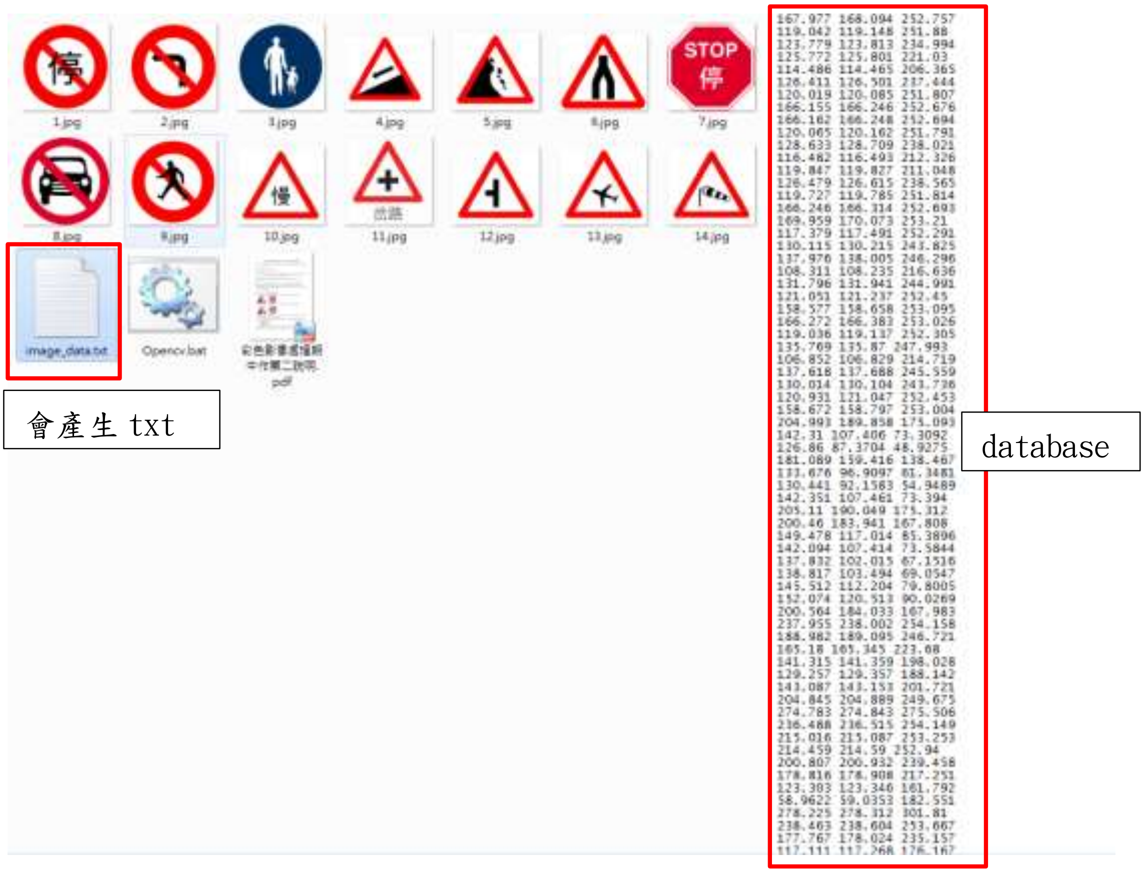
```
    for (j = 0; j < cols; j = j + cols / part)//垂直切
    {
        //for (i = 0; i < rows; i++)
        {
            meanR = 0;
            meanG = 0;
            meanB = 0;
            for (m = 0; m < cols / part; m++)
            {
                for (l = 0; l < rows; l++)
                {
                    if ((j + m) < cols)
                    {
                        meanR += (double)image.at<Vec3b>(l, j + m)[0] / (cols / part) /
rows;
                        meanG += (double)image.at<Vec3b>(l, j + m)[1] / (cols / part) /
rows;
                        meanB += (double)image.at<Vec3b>(l, j + m)[2] / (cols / part) /
rows;
                    }
                }
            }
            PupartionInfo << (double)meanR << " " << (double)meanG << " " <<
(double)meanB << " ";//輸出平均值到 txt 檔
            PupartionInfo << "\n";
        }
    }


    //for (j = 0; j < cols; j++)//水平切
    {
        for (i = 0; i < rows; i = i + rows / part)
        {
            meanR = 0;
            meanG = 0;
            meanB = 0;
            for (m = 0; m < cols; m++)
            {
                for (l = 0; l < rows / part; l++)
                {
                    if ((i + l) < rows)
                    {
                        meanR += (double)image.at<Vec3b>(l + i, m)[0] / (cols / part) / rows;
                        meanG += (double)image.at<Vec3b>(l, m)[1] / (cols / part) / rows;
                        meanB += (double)image.at<Vec3b>(l, m)[2] / (cols / part) / rows;
                    }
                }
            }
            PupartionInfo << (double)meanR << " " << (double)meanG << " " << (double)meanB << " ";//輸出平均值到 txt 檔
            PupartionInfo << "\n";
        }
    }

    PupartionInfo.close();

    system("pause");
    return 0;
```

三、執行成果影像範例



```
D:\由科學第6-彩色影像處理OpenCV2.4.6_x86_Portable_Project\bin\OpenCV.exe 1.jpg out.jpg
D:\由科學第6-彩色影像處理OpenCV2.4.6_x86_Portable_Project\bin\OpenCV.exe 2.jpg out.jpg
D:\由科學第6-彩色影像處理OpenCV2.4.6_x86_Portable_Project\bin\OpenCV.exe 3.jpg out.jpg
D:\由科學第6-彩色影像處理OpenCV2.4.6_x86_Portable_Project\bin\OpenCV.exe 4.jpg out.jpg
D:\由科學第6-彩色影像處理OpenCV2.4.6_x86_Portable_Project\bin\OpenCV.exe 5.jpg out.jpg
D:\由科學第6-彩色影像處理OpenCV2.4.6_x86_Portable_Project\bin\OpenCV.exe 6.jpg out.jpg
D:\由科學第6-彩色影像處理OpenCV2.4.6_x86_Portable_Project\bin\OpenCV.exe 7.jpg out.jpg
D:\由科學第6-彩色影像處理OpenCV2.4.6_x86_Portable_Project\bin\OpenCV.exe 8.jpg out.jpg
D:\由科學第6-彩色影像處理OpenCV2.4.6_x86_Portable_Project\bin\OpenCV.exe 9.jpg out.jpg
D:\由科學第6-彩色影像處理OpenCV2.4.6_x86_Portable_Project\bin\OpenCV.exe 10.jpg out.jpg
D:\由科學第6-彩色影像處理OpenCV2.4.6_x86_Portable_Project\bin\OpenCV.exe 11.jpg out.jpg
D:\由科學第6-彩色影像處理OpenCV2.4.6_x86_Portable_Project\bin\OpenCV.exe 12.jpg out.jpg
D:\由科學第6-彩色影像處理OpenCV2.4.6_x86_Portable_Project\bin\OpenCV.exe 13.jpg out.jpg
D:\由科學第6-彩色影像處理OpenCV2.4.6_x86_Portable_Project\bin\OpenCV.exe 14.jpg out.jpg
```

點選批次檔即可
快速輸入影像，
建立 database



```
167.977 168.094 252.757
119.042 119.148 251.88
123.779 123.813 234.994
125.772 125.801 221.03
114.486 114.465 206.365
126.411 126.501 237.444
120.019 120.085 251.807
166.155 166.246 252.676
166.162 166.248 252.694
120.065 120.162 251.791
128.633 128.709 238.021
116.482 116.493 212.326
119.847 119.827 211.048
126.479 126.615 238.565
119.727 119.785 251.814
166.246 166.314 252.693
169.959 170.073 253.21
117.379 117.491 252.291
130.115 130.215 243.825
137.976 138.005 246.296
108.311 108.235 216.636
131.796 131.941 244.991
121.051 121.237 252.45
158.577 158.658 253.095
166.272 166.383 253.026
119.036 119.137 252.305
135.769 135.87 247.993
106.852 106.829 214.719
137.618 137.688 245.559
130.014 130.104 243.736
120.931 121.047 252.453
158.672 158.797 253.004
204.993 189.858 175.093
142.31 107.406 73.3092
126.86 87.3704 48.9275
181.089 159.416 138.467
133.676 96.9097 61.3481
130.441 92.1583 54.9489
142.351 107.461 73.394
205.11 190.049 175.312
200.46 183.941 167.808
149.478 117.014 85.3896
142.094 107.414 73.5844
137.832 102.015 67.1516
138.817 103.494 69.0547
145.512 112.204 79.8005
152.074 120.513 90.0269
200.564 184.033 167.983
237.955 238.002 254.158
188.982 189.095 246.721
165.18 165.345 223.68
141.315 141.359 198.028
129.257 129.357 188.142
143.087 143.153 201.721
204.845 204.889 249.675
274.783 274.843 275.506
236.488 236.515 254.149
215.016 215.087 253.253
214.459 214.59 252.94
200.807 200.932 239.458
178.816 178.908 217.251
123.303 123.346 161.792
58.9622 59.0353 182.551
278.225 278.312 301.81
238.463 238.604 253.667
177.767 178.024 235.157
117.111 117.268 176.167
```

會產生 txt

database

找出最相似的五張號誌圖片程式碼

```
#include <stdio.h>
#include <math.h>
#include <iostream>
#include <stdlib.h>
#include <fstream>
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
using namespace cv;
using namespace std;
char filename[100];
char windowname[100];

int main(int argc, char *argv[])
{

    ifstream in("image_data.txt");//讀取 database
    double s;
    double Control_Group[1000];
    int i = 0;
    if (!in.is_open())
    {
        cout << "Error opening file"; exit(1);
    }
    while (!in.eof())
    {
        while (in >> s)
        {
            Control_Group[i] = s;
            i++;
        }
    }
    Mat image = imread("13.jpg", CV_LOAD_IMAGE_COLOR);//讀取比較圖片
    cvNamedWindow("原圖", 0);
    imshow("原圖", image);
    int cols = image.cols;//width
    int rows = image.rows;//height
    //printf("%d,%d", cols, rows);
    double meanR, meanG, meanB;
    int j, m, l;
    int part;
    part = 8;
    double data[100];

    int p = 0;

    for (j = 0; j < cols; j = j + cols / part)//垂直切-1
    {
        //for (i = 0; i < rows; i++)
        {
            meanR = 0;
            meanG = 0;
            meanB = 0;
            for (m = 0; m < cols / part; m++)
            {
                for (l = 0; l < rows; l++)
                {
                    if ((j + m) < cols)
                    {
                        meanR += (double)image.at<Vec3b>(l, j + m)[0] /
(cols / part) / rows;
                        meanG += (double)image.at<Vec3b>(l, j + m)[1] /
(cols / part) / rows;
                        meanB += (double)image.at<Vec3b>(l, j + m)[2] /
(cols / part) / rows;
                    }
                }
            }
            data[p] = meanR;
            data[p+1] = meanG;
            data[p+2] = meanB;
            p = p + 3;
        }
    }


    //for (j = 0; j < cols; j++)//水平切-1
    {
        for (i = 0; i < rows; i = i + rows / part)
        {
            meanR = 0;
            meanG = 0;
            meanB = 0;
            for (m = 0; m < cols; m++)
            {
                for (l = 0; l < rows / part; l++)
                {
                    if ((i + l) < rows)
                    {
                        meanR += (double)image.at<Vec3b>(l+i, m)[0] / (cols / part) / rows;
                        meanG += (double)image.at<Vec3b>(l, m)[1] / (cols / part) / rows;
                        meanB += (double)image.at<Vec3b>(l, m)[2] / (cols / part) / rows;
                    }
                }
            }
            data[p] = meanR;
            data[p + 1] = meanG;
            data[p + 2] = meanB;
            p = p + 3;
        }
    }

    double min = 100000;
    double difference;
    double d[12];
    int num[12];
    int n = 0;
    for (int s = 0; s < 12; s++)//算圖片與database的差異
    {
        difference = 0;
        for (i = 0; i < 48; i++)
        {
            difference = double(difference + fabs(data[i] -
Control_Group[n]));
            /*cout << difference << " " << data[i] << " " <<
Control_Group[n] << " ";*/
            n++;
        }
        d[s] = difference;
        num[s] = s+1;
    }

    for (int n = 11; n > 0; n--)//泡沫排序法，由小到大
    {
        for (int m = 0; m < n; m++)
        {
            if (d[m] > d[m + 1])
            {
                int h;
                h = d[m];
                d[m] = d[m + 1];
                d[m + 1] = h;

                int h0;
                h0 = num[m];
                num[m] = num[m + 1];
                num[m + 1] = h0;
            }
        }
    }



    for (int i = 0; i < 5; i++)//展現差異最小前五名的圖片
    {
        int a = num[i];
        sprintf(filename, "%d.jpg", a);
        sprintf(windowname, "圖片%d排名%d", a, i+1);
        IplImage* pScr = cvLoadImage(filename, 1);
        cvNamedWindow(windowname,0);
        cvResizeWindow(windowname, 300, 300);
        cvShowImage(windowname, pScr);

    }

    waitKey(0);

    return 0;
}
```

四、執行成果影像範例

資料庫圖片如下:

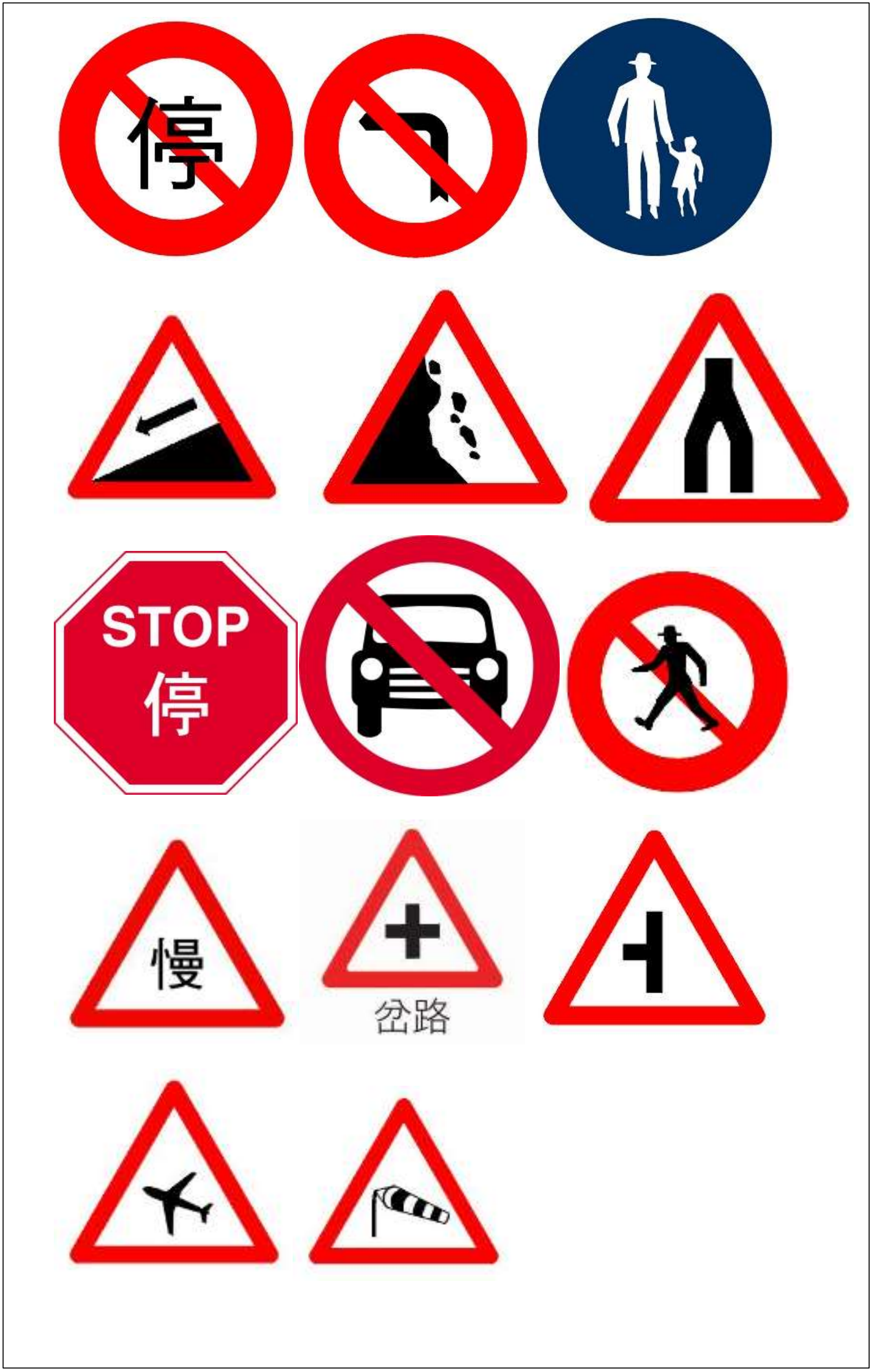
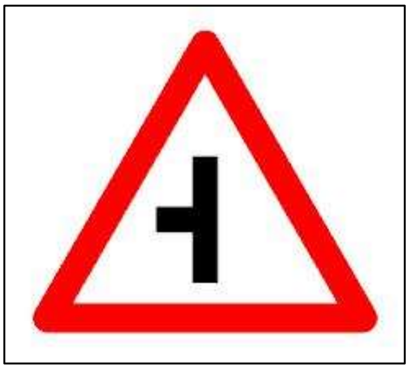


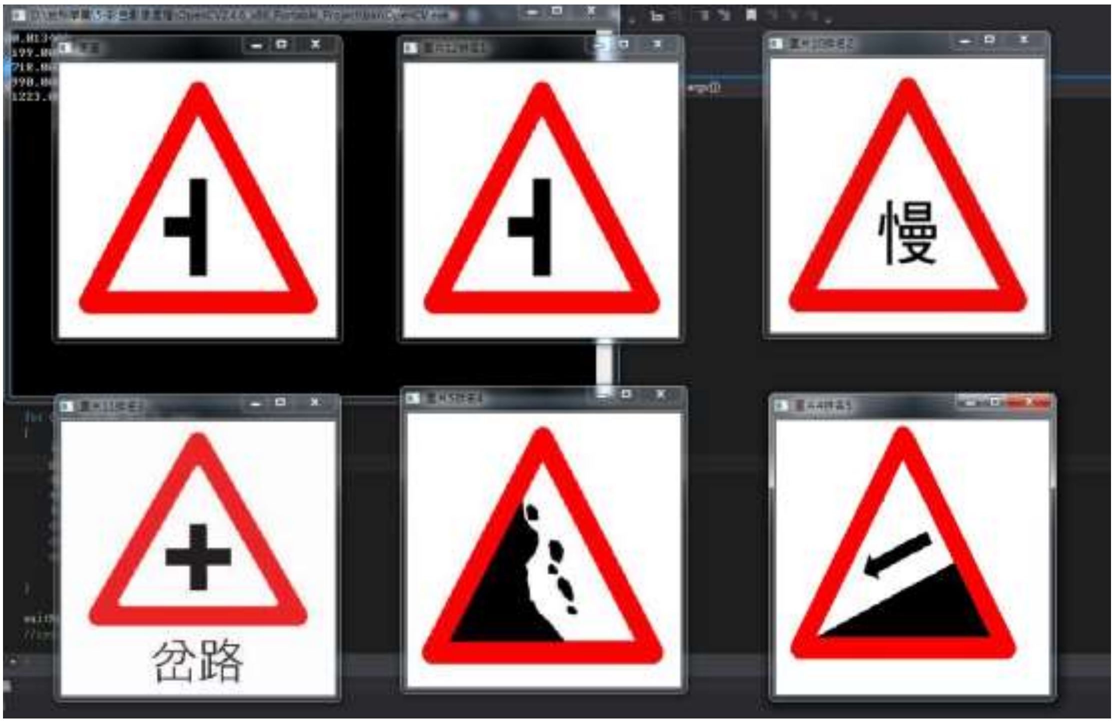
示意圖:

程式執行結果:左圖為輸入影像，下圖為前五名圖片。



<table>
<tr><td colspan="2"> </td><td colspan="2"> </td><td colspan="2"> </td><td colspan="2"> </td><td colspan="2"> </td></tr>
<tr><td>差異度</td><td>0.013</td><td>差異度</td><td>199</td><td>差異度</td><td>718</td><td>差異度</td><td>990</td><td>差異度</td><td>1233</td></tr>
</table>

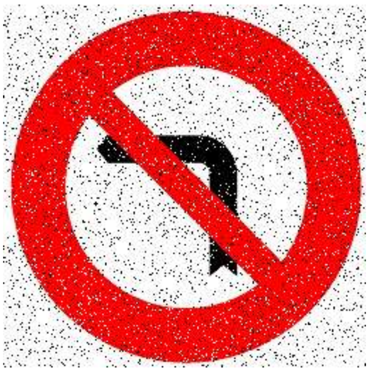
實際執行程式畫面



```
0.0134
199.00
718.00
990.00
1223.0
```

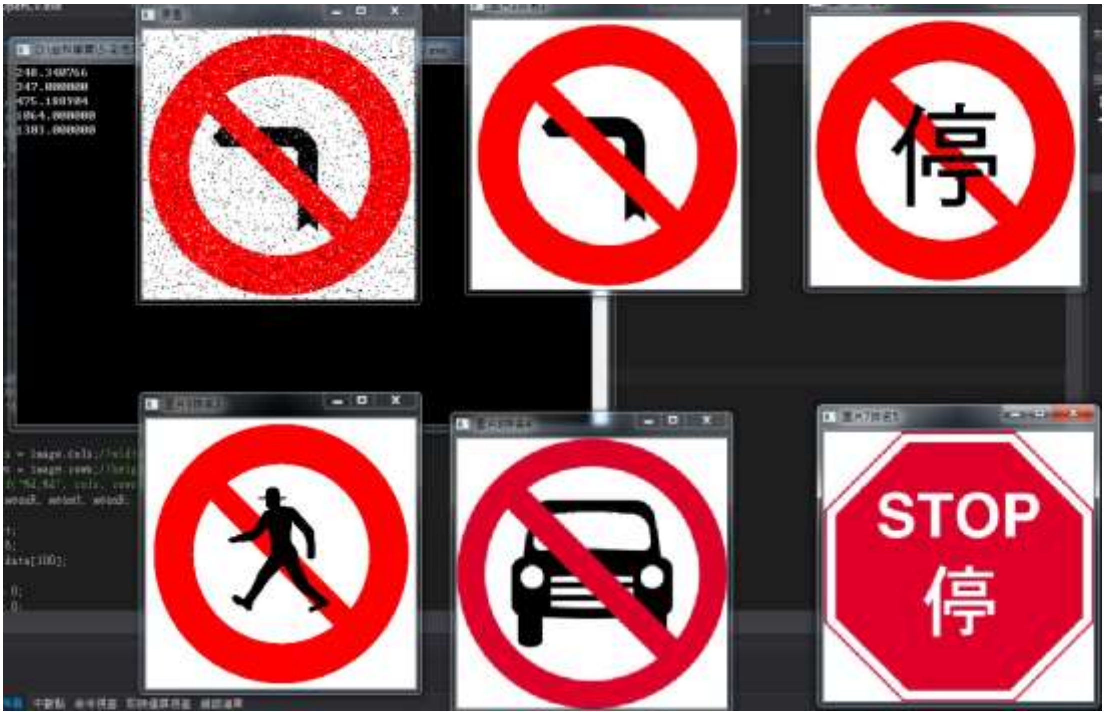
示意圖

程式執行結果:左圖為輸入影像，右圖為輸入雜訊，下圖為前五名圖片。

差 異 度	248	差 異 度	347	差 異 度	475	差 異 度	1064	差 異 度	1303

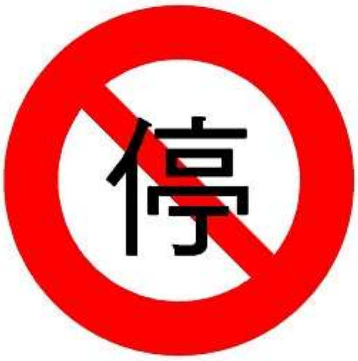
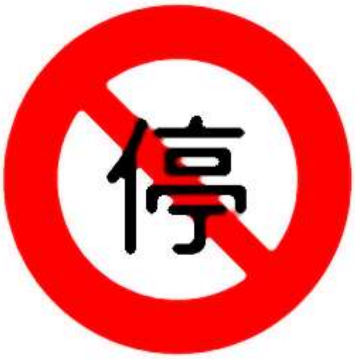
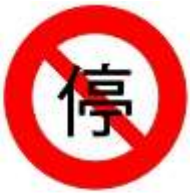
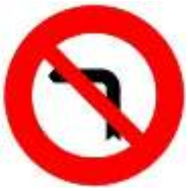
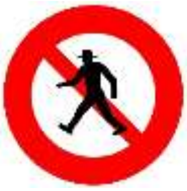
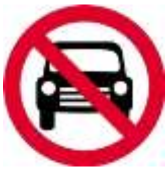
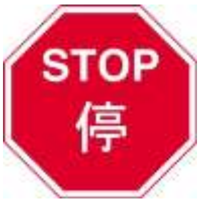
實際執行程式畫面



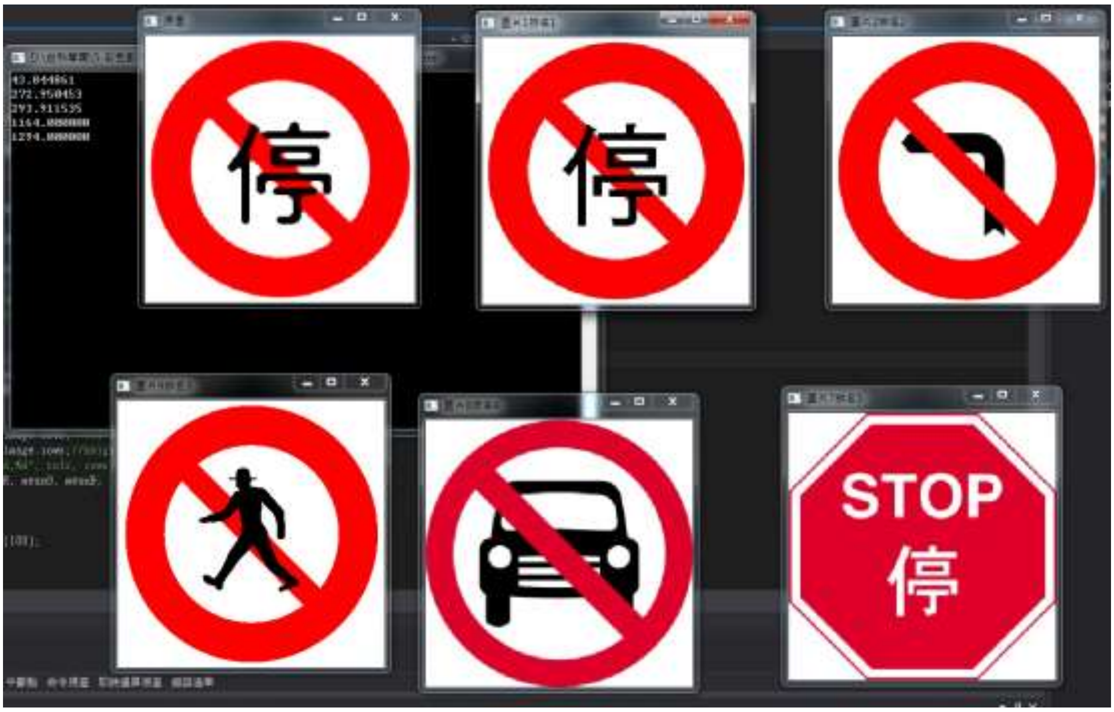
```
248.340766
347.000000
475.180904
1064.000000
1303.000000
```

示意圖

程式執行結果:左圖為輸入影像，右圖為平均值濾波，下圖為前五名圖片。

<table>
<tr><td colspan="6"></td><td colspan="4"></td></tr>
<tr><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td>差異度</td><td>43</td><td>差異度</td><td>272</td><td>差異度</td><td>293</td><td>差異度</td><td>1164</td><td>差異度</td><td>1294</td></tr>
</table>

實際執行程式畫面



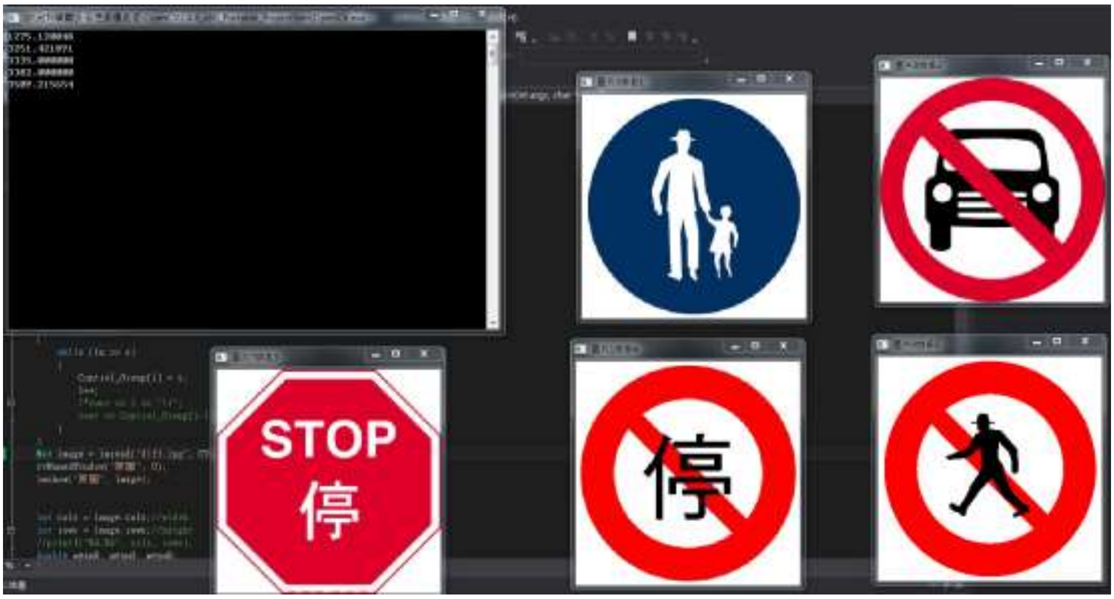
```
43.044861
272.950453
293.911535
1164.000000
1294.000000
```

示意圖

程式執行結果:左圖為輸入影像，右圖為偏色影像，下圖為前五名圖片。

<table>
<tr><td colspan="4"> </td><td colspan="6"> </td></tr>
<tr><td colspan="2"> </td><td colspan="2"> </td><td colspan="2"> </td><td colspan="2"> </td><td colspan="2"> </td></tr>
<tr><td>差異度</td><td>1275</td><td>差異度</td><td>3251</td><td>差異度</td><td>3335</td><td>差異度</td><td>3382</td><td>差異度</td><td>3509</td></tr>
</table>

實際執行程式畫面



```
1275.120048
3251.421971
3335.000000
3382.000000
3509.215654
```