

《机器视觉》实验指导书

V 1.0

实验四 光学字符识别（OCR）实验



哈尔滨工业大学（深圳）
实验与创新实践教育中心

目录

4.1 实验目的	3
4.2 实验原理	3
4.3 实验设备	3
4.4 实验内容	4
4.5 实验步骤——Halcon 实现车牌号识别	4
4.5.1 Blob 分析+mlp 分类器实现车牌识别	4
4.5.2 模板匹配+mlp 分类器实现车牌定位、识别	7
4.6 实验任务要求	12

4.1 实验目的

- 1) 了解 OCR 检测的基本原理;
- 2) 掌握基于 Halcon 的 OCR 检测过程, 能够使用相关算子实现车牌字符识别;

4.2 实验原理

OCR (光学字符识别, Optical Character Recognition), 是通过图像处理和模式识别技术对光学字符进行识别, OCR 是自动识别技术研究和应用领域中的一个重要方面。通过检测暗、亮的模式确定形状, 然后用字符识别方法将形状翻译成计算机文字, 如何除错或利用辅助信息提高识别正确率, 是 OCR 最重要的课题。衡量一个 OCR 系统性能好坏的主要指标有: 拒识率、误识率、识别速度、用户界面的友好性, 产品的稳定性, 易用性及可行性等。

本实验是针对车牌字符识别进行研究, 程序流程如图 4-1 所示。

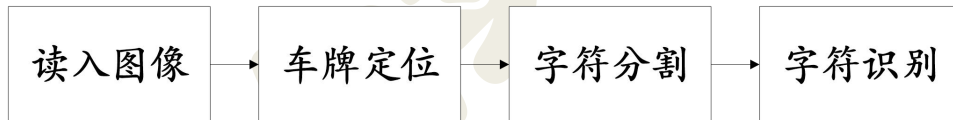


图 4-1 OCR 程序流程

分类器的作用是将目标对象指定给多个类别中的一个, OCR 是一种进一步的分类方法, 识别的第一步是将独立的字符区域从图像中提取出来, 然后将其指定给某个字符的种类。MLP、SVM 和 k-NN 分类器都可以用于 OCR。本实验采用 MLP 分类器进行字符识别。

4.3 实验设备

Basler A1300-60gm 黑白相机 (包含 12mm 镜头及安装支架)、塑料小车牌、低角度环形光。

4.4 实验内容

基于 Halcon 编程实现车牌号识别，显示识别结果，并在 VS 平台成功执行。

4.5 实验步骤——Halcon 实现车牌号识别

4.5.1 Blob 分析+mlp 分类器实现车牌识别

Halcon 中实现 OCR 识别，其内部给我们提供了 OCR 助手，但局限于汉字不能识别，因此要进行汉字识别需要通过创建训练文件、训练 OCR 分类器使 halcon 来识别汉字字符。在本次实验中，**仅识别非中文的、标准的字符**例如数字和英文字母。直接调用 halcon 内置的分类器，建议选“Industrial_0-9A-Z_NoRej.omc”。

- ① read_image、rgb1_to_gray



图 4-2 读入图像

- ② 创建 ROI，图片剪裁，根据灰度直方图进行阈值分割，获取连通域，打开特征直方图根据面积特征剔除字符“A”与“Z”之间圆点。

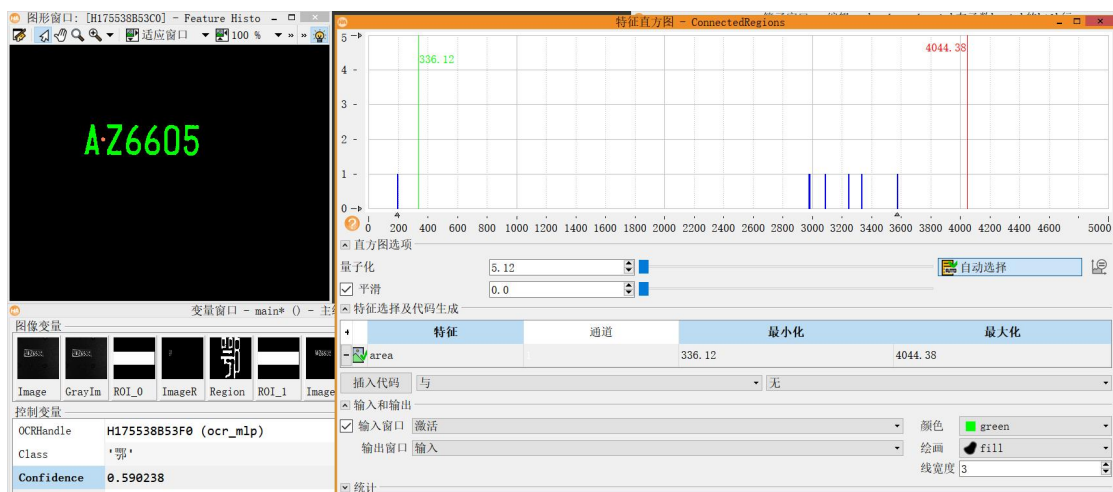


图 4-3 特征提取

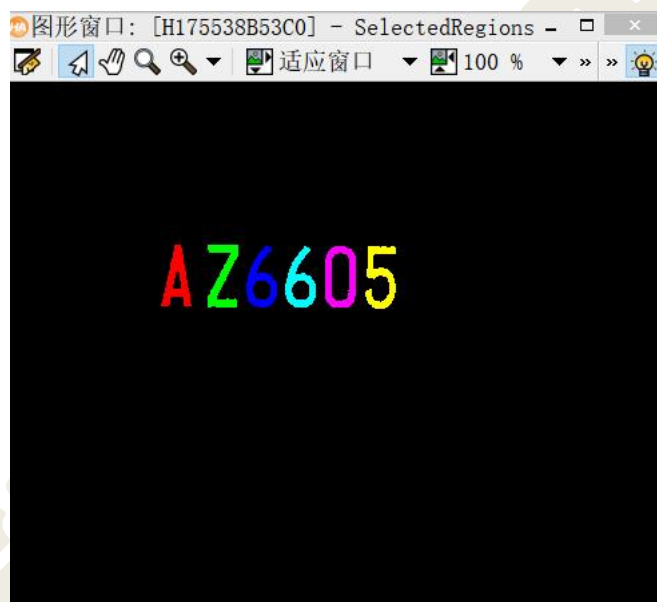


图 4-4 特征提取结果

③ 排列字符（直接做字符识别顺序是乱的）`sort_region`。由于 `halcon` 的分类器做训练时用的都是白底黑字，所以部分图像做识别之前需要执行 `invert_image` 对图像取反，如下图。

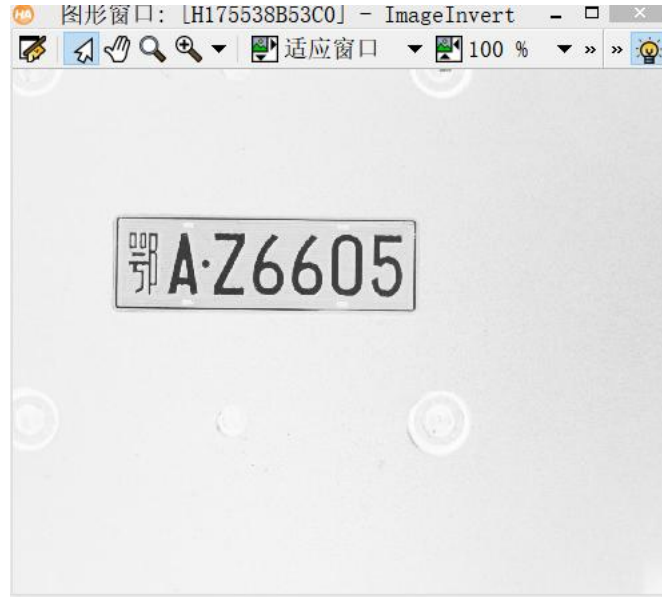


图 4-5 图像取反

④ 读取字符分类器 `read_ocr_class_mlp`，分类器选择“Industrial_0-9A-Z_NoRej.omc”（一定要选这个），然后做字符识别 `do_ocr_multi_class_mlp`；最后显示结果。实验中，部分车牌识别会出现将字符“Q”误识别为“0”的现象，实际解决此问题需要重新训练对应的分类器，本次实验对此处不做要求。



图 4-6 识别结果（不要求识别汉字）

⑤ 参考代码如下：

```
1 *关闭默认窗口
2 dev_close_window()
3 read_image(Image, 'C:/Users/30863/Desktop/IMAGES/OCR/OCR/Image__2021-05-07__17-40-58.jpg')
4 *新建窗口
5 dev_open_window_fit_image(Image, 0, 0, -1, -1, WindowHandle)
6 dev_display(Image)
7 rgb1_to_gray(Image, GrayImage)
8 *获取ROI
9 gen_rectangle1 (ROI_0, 208.567, 319.5, 351.927, 804.193)
10 *图片剪裁
11 reduce_domain(GrayImage, ROI_0, ImageReduced)
12 *阈值分割
13 threshold (ImageReduced, Regions, 127, 255)
14 *断开连通域
15 connection(Regions, ConnectedRegions)
16 *特征选择
17 select_shape (ConnectedRegions, SelectedRegions, 'area', 'and', 890.68, 5000)
18 *连通域排序
19 sort_region(SelectedRegions, SortedRegions, 'first_point', 'true', 'column')
20 *读分类器
21 read_ocr_class_mlp('D:/halcon18/ocr/Industrial_NoRej.omc', OCRHandle)
22 *取反图像（只有黑底图像需要）
23 invert_image(ImageReduced, ImageInvert)
24 *字符识别
25 do_ocr_multi_class_mlp(SortedRegions, ImageInvert, OCRHandle, Class, Confidence)
26 *显示
27 dev_display(Image)
28 count_obj(SelectedRegions, Number)
29 for index:=0 to Number-1 by 1
30     disp_message(WindowHandle, Class[index], 'window', 12, 12+index*15, 'black', 'true')
31 endfor
```

4.5.2 模板匹配+mlp 分类器实现车牌定位、识别

① 读图，以车牌上的汉字区域绘出 ROI，创建基于 shape 的模板（步骤参考实验二）。

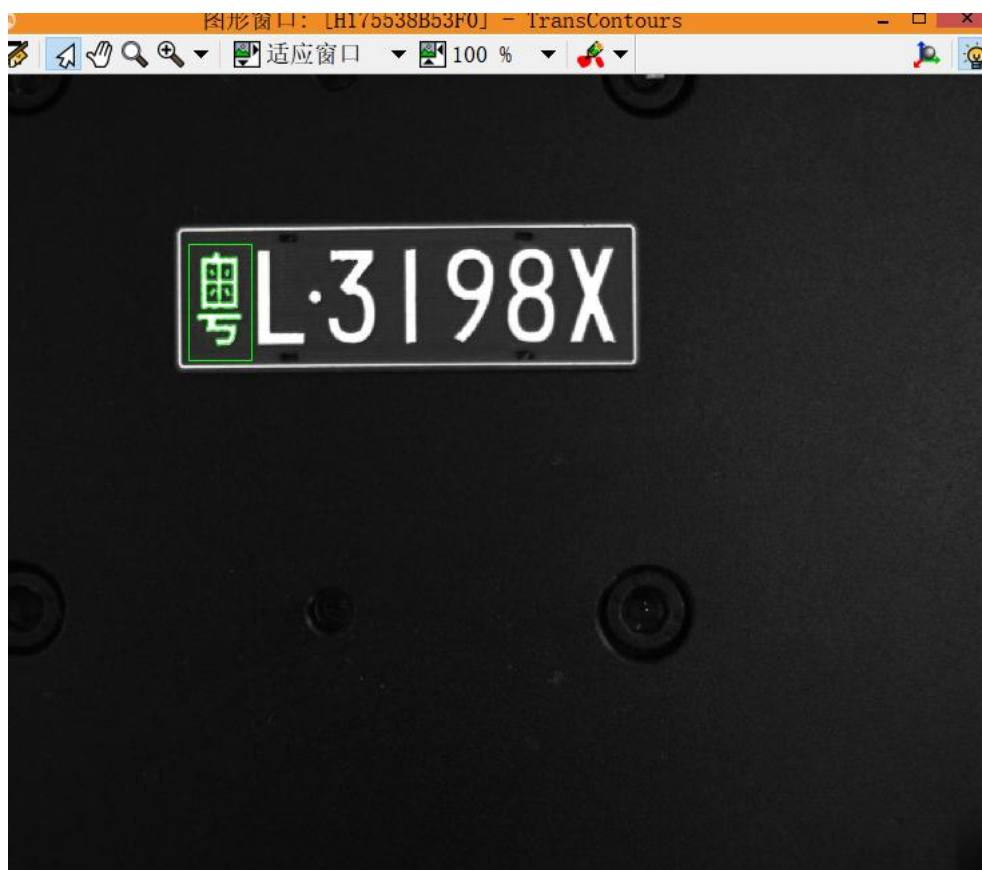


图 4-7 创建模板

② 通过模板匹配找到待匹配的车牌，并得到仿射矩阵（匹配助手——代码生成选项卡——生成校正代码（勾选））。然后创建字符区域 ROI，blob 分析得到字符连通域。



图 4-8 得到仿射矩阵

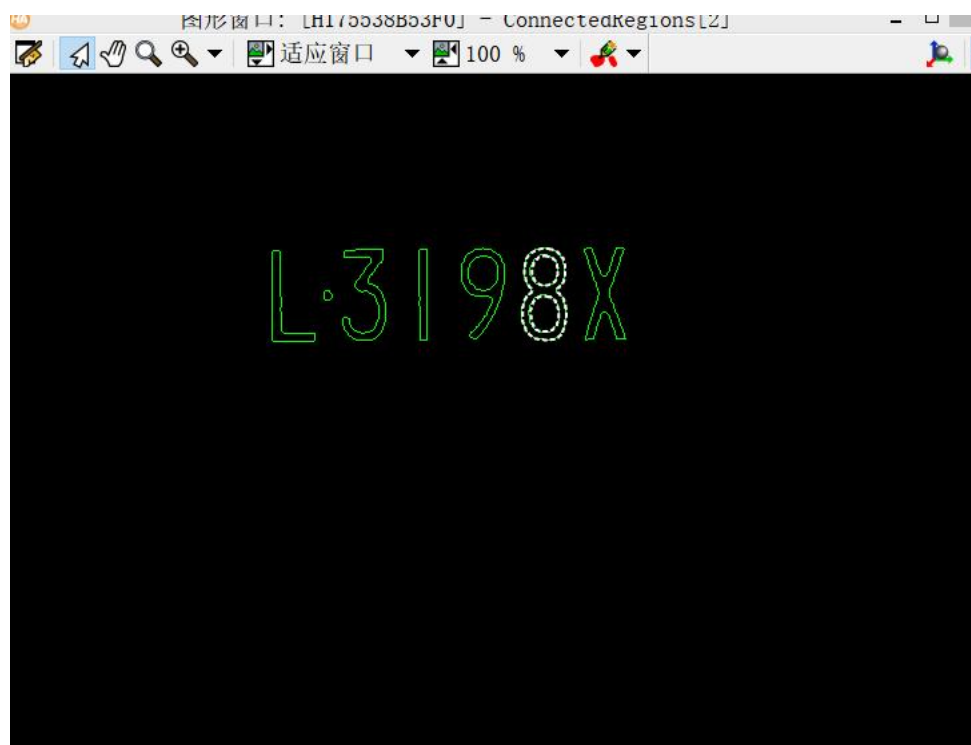


图 4-9 字符连通域

- ③ 根据面积特征，剔除非字符部分。

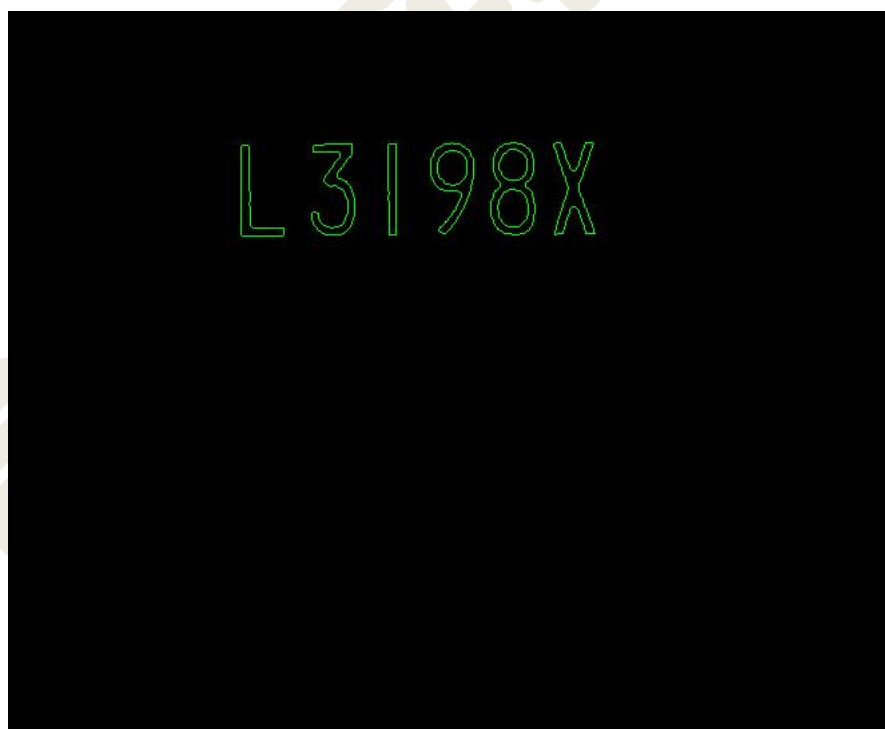


图 4-10 仅保留字符

- ④ 变化车牌位置、方向，通过仿射变换即可实现位置变化的车牌定位和车牌号识别。

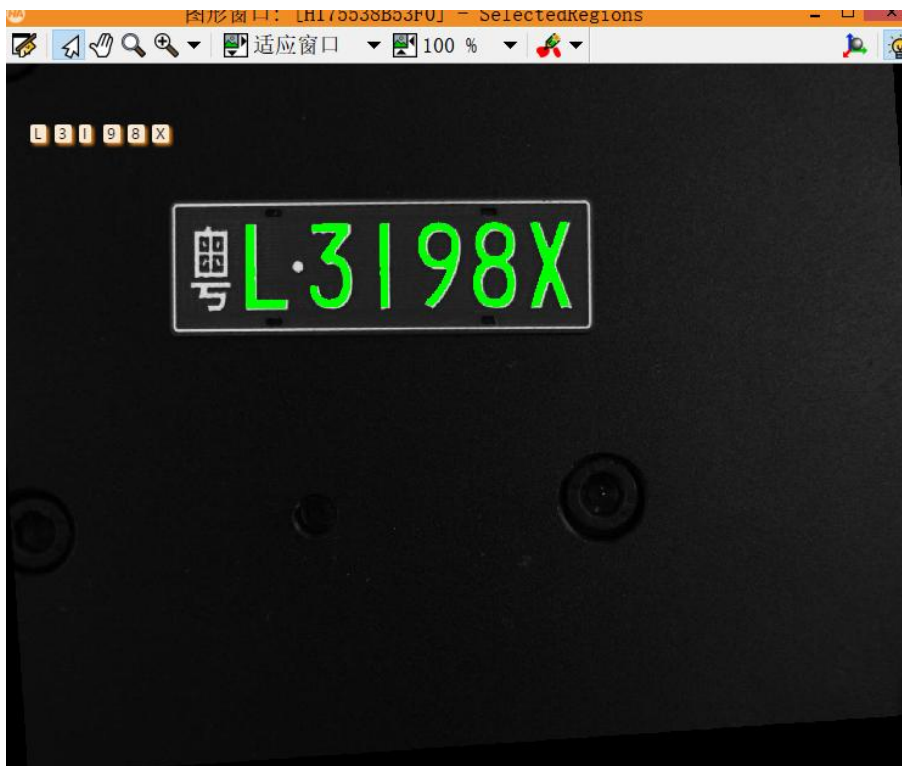


图 4-11 字符识别结果

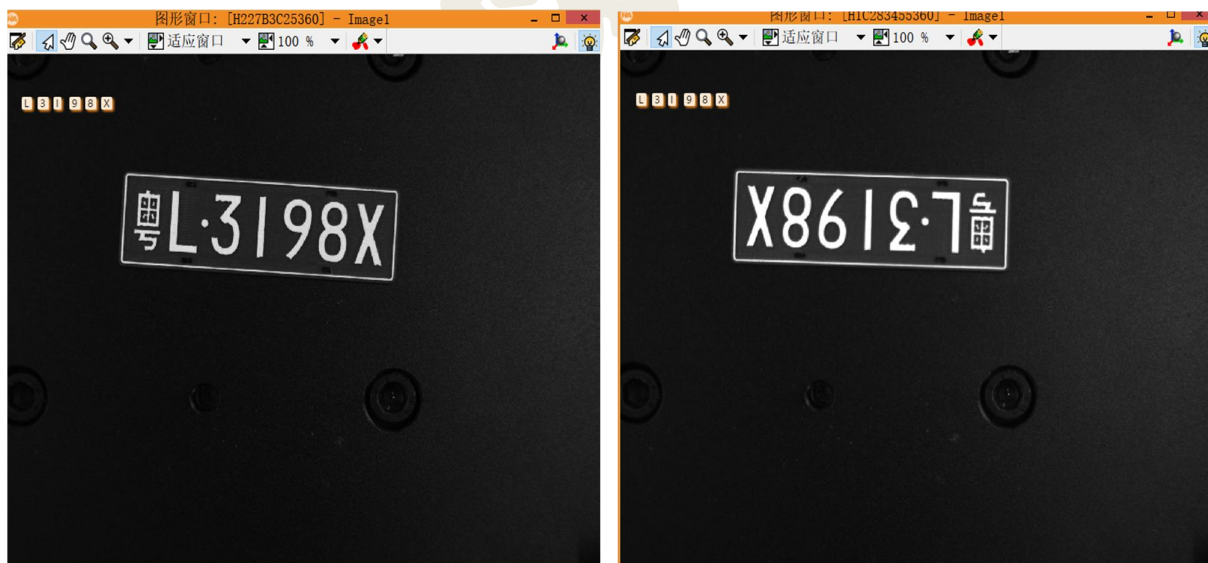


图 4-12 位置方向变化后识别结果 (a)



图 4-12 位置方向变化后识别结果(b)

⑤ 参考代码:

...

读图创建模板 `create_shape_model`读不同位置、方向车牌图片 `TestImages := [...]`查找模板 `find_shape_model`仿射变换 `hom_mat2d_identity`、`hom_mat2d_translate`、`hom_mat2d_rotate`、`affine_trans_image`生成检测区域 `gen_rectangle`图片剪裁 `reduce_domain`获取连通域 `connection`提取字符连通域 `select_shape`排列字符连通域 `sort_region`读分类器 `read_ocr_class_mlp`字符识别 `do_ocr_multi_class_mlp`显示 `disp_message`

⑥ VS 联合 halcon 实现上述功能，读入不同位置方向的图片，输出匹配后的字符识别结果，每幅图都弹窗显示。

参考结果如下图所。

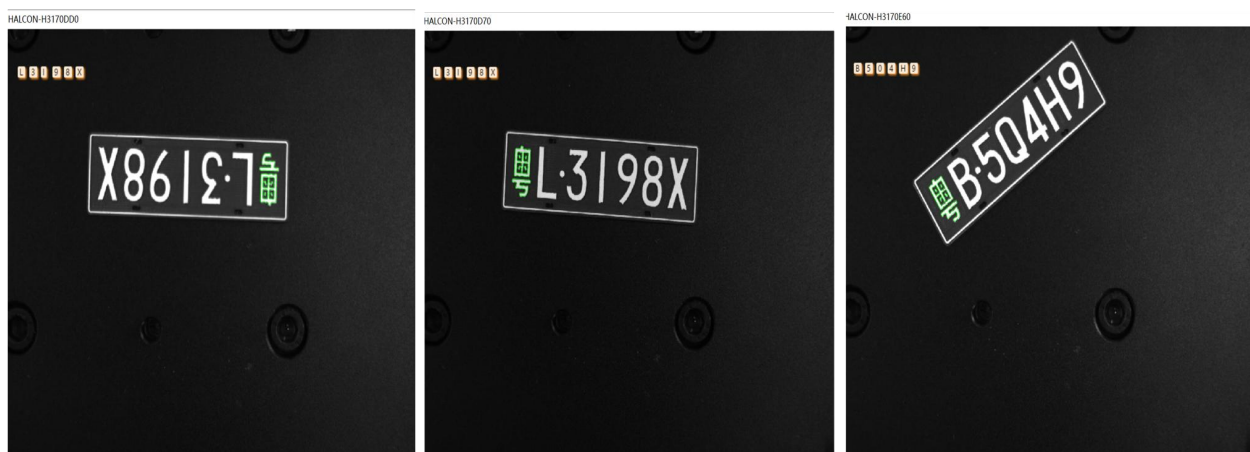


图 4-13 VS 联合 halcon 字符识别

4.6 实验任务要求

基于 VS 调用 Halcon 通过模板匹配+mlp 分类器实现位置变化的车牌定位(定位选择汉字做基于形状模板匹配)和字符识别,只识别字母和数字即可,并在窗口显示结果,要求导入不同位置方向的图片至少 2 张(也可做实时效果)。

附录——整体参考代码

```

1 *关闭默认窗口
2 dev_close_window()
3 read_image(Image, 'F:/学校实验课文件/2024年机器视觉/IMAGES/OCR/Image__2021-05-07__17-40-58.jpg')
4 *新建窗口
5 dev_open_window_fit_image(Image, 0, 0, -1, -1, WindowHandle)
6 dev_display(Image)
7 rgb1_to_gray(Image, GrayImage)
8 *获取ROI
9 gen_rectangle1 (ROI_0, 208.567, 319.5, 351.927, 804.193)
10 *图片剪裁
11 reduce_domain(GrayImage, ROI_0, ImageReduced)
12 *阈值分割
13 threshold (ImageReduced, Regions, 127, 255)
14 *断开连通域
15 connection(Regions, ConnectedRegions)
16 *特征选择
17 select_shape (ConnectedRegions, SelectedRegions, 'area', 'and', 890.68, 5000)
18 *连通域排序
19 sort_region(SelectedRegions, SortedRegions, 'first_point', 'true', 'column')
20 *读分类器
21 read_ocr_class_mlp('C:/Program Files/MVTec/HALCON-18.11-Progress/ocr/Industrial_NoRej.omec', OCRHandle)
22 *取反图像 (只有黑底图像需要)
23 invert_image(ImageReduced, ImageInvert)
24 *字符识别
25 do_ocr_multi_class_mlp(SortedRegions, ImageInvert, OCRHandle, Class, Confidence)
26
27 *停止窗口刷新
28 dev_update_window('off')
29 *设置字体
30 set_display_font (WindowHandle, 32, 'mono', 'true', 'false')
31 *显示
32 dev_display(Image)

33 count_obj(SelectedRegions, Number)
34 for index:=0 to Number-1 by 1
35     disp_message(WindowHandle, Class[index], 'window', 12, 12+index*40, 'black', 'true')
36 endfor
37 stop ()
38
39
40 *划取汉字区域ROI. 建议以汉字为模板的定位过程
41 gen_rectangle1 (ModelRegion, 211.98, 249.527, 360.46, 319.5)
42 *剪裁
43 reduce_domain (Image, ModelRegion, TemplateImage)
44 * 创建模板
45 create_shape_model (TemplateImage, 4, rad(0), rad(360), rad(1.9033), ['none', 'no_pregeneration'], 'use_polarity', [11,17,4], 4, ModelID)
46 * 获取轮廓
47 get_shape_model_contours (ModelContours, ModelID, 1)
48
49 * 计算模板位置角度信息
50 area_center (ModelRegion, ModelRegionArea, RefRow, RefColumn)
51 vector_angle_to_rigid (0, 0, 0, RefRow, RefColumn, 0, HomMat2D)
52 affine_trans_contour_xld (ModelContours, TransContours, HomMat2D)
53
54 *辅助显示
55 dev_set_color ('green')
56 dev_set_draw ('margin')
57 dev_display (ModelRegion)
58 dev_display (TransContours)
59
60 *暂停, 便于观察
61 stop ()
62

63 *准备弹出不同窗口, 赋值不同的窗口ID号
64 WindowHandle1:=3200
65 TestImages := ['F:/学校实验课文件/2024年机器视觉/IMAGES/OCR/Image__2021-05-07__17-39-50.jpg', 'F:/学校实验课文件/2024年机器视觉/IMAGES/OCR/Image__2021-05-07__17-40-44.jpg']
66 for T := 0 to 1 by 1
67     WindowHandle1:=3200+T
68     * Matching 02: Obtain the test image
69     read_image (Image, TestImages[T])
70     *新建窗口
71     dev_open_window_fit_image(Image, 0, 0, -1, -1, WindowHandle1)
72
73     * Matching 02: Find the model
74     *基于形状模板匹配找汉字
75     find_shape_model (Image, ModelID, rad(0), rad(360), 0.8, 0, 0.5, 'least_squares', [4,1], 0.9, Row, Column, Angle, Score)
76     *
77     * Matching 02: Transform the model contours into the detected positions
78
79     for I := 0 to |Score| - 1 by 1
80         hom_mat2d_identity (HomMat2D)
81         hom_mat2d_rotate (HomMat2D, Angle[I], 0, 0, HomMat2D)
82         hom_mat2d_translate (HomMat2D, Row[I], Column[I], HomMat2D)
83         affine_trans_contour_xld (ModelContours, TransContours, HomMat2D)
84         dev_set_color ('green')
85         dev_display (TransContours)
86         stop ()
87     endfor
88     *
89
90     *将车牌位置通过仿射变换放正, 学生自行解释每一行的意义
91     for I := 0 to |Score| - 1 by 1
92         hom_mat2d_identity (RectificationHomMat2D)
93         hom_mat2d_translate (RectificationHomMat2D, RefRow-Row[I], RefColumn-Column[I], RectificationHomMat2D)
94         hom_mat2d_rotate (RectificationHomMat2D, -Angle[I], RefRow, RefColumn, RectificationHomMat2D)
95         affine_trans_image (Image, RectifiedImage, RectificationHomMat2D, 'constant', 'false')

```

```
96
97 * 校正后的图片, 即可做字符识别, 流程同上。
98 rgb1_to_gray(RectifiedImage, GrayImage)
99 gen_rectangle1 (ROI_0, 208.567, 319.5, 351.927, 804.193)
100 reduce_domain(GrayImage, ROI_0, ImageReduced)
101 threshold (ImageReduced, Regions, 127, 255)
102 connection(Regions, ConnectedRegions)
103 select_shape (ConnectedRegions, SelectedRegions, 'area', 'and', 890.68, 5000)
104 sort_region(SelectedRegions, SortedRegions, 'first_point', 'true', 'column')
105 read_ocr_class_mlp('C:/Program Files/MVTec/HALCON-18.11-Progress/ocr/Industrial_NoRej.omc', OCRHandle)
106 invert_image(ImageReduced, ImageInvert)
107 do_ocr_multi_class_mlp(SortedRegions, ImageInvert, OCRHandle, Class, Confidence)
108 dev_display(Image)
109 count_obj(SelectedRegions, Number)
110 *设置字体
111 set_display_font (WindowHandle1, 32, 'mono', 'true', 'false')
112 for index:=0 to Number-1 by 1
113     disp_message(WindowHandle1, Class[index], 'window', 12, 12+index*40, 'red', 'true')
114     endfor
115 endfor
116
117 dev_display (TransContours)
118 stop()
119
120 endfor
```