

長安大學

## 程序开发文档

题目：\_\_\_\_\_车辆的车牌分割\_\_\_\_\_

学院：\_\_\_\_\_信息工程学院\_\_\_\_\_

专业：\_\_\_\_\_计算机科学与技术\_\_\_\_\_

班级：\_\_\_\_\_2015240204\_\_\_\_\_

成员：\_\_\_\_\_张伟 李毅博\_\_\_\_\_

# 目录

|                                   |    |
|-----------------------------------|----|
| 一 环境配置 .....                      | 3  |
| 1.1 Visual Studio 版本修改 .....      | 3  |
| 1.2 Qt5.9.1 及 Qt VS Tool 安装 ..... | 5  |
| 二 项目实施 .....                      | 9  |
| 2.1 车牌定位分割 .....                  | 9  |
| 2.2 字符分割 .....                    | 12 |
| 2.3 字符识别 .....                    | 13 |

# 一 环境配置

本程序需要安装 Visual Studio2015 以及 OpenCV3.2.0

## 1.1 Visual Studio 版本修改

由于 VS 版本比较多，低版本无法直接打开高版本的工程文件，通过对工程文件进行一些修改可以解决这些问题。

通过修改 Solution 文件和 Project 文件，可以使低版本转换为高版本。例如将 VS2015 工程转换为 VS2010 工程。

### 1. 修改解决方案文件

使用记事本打开 .lsn 文件：

```
Microsoft Visual Studio Solution File, Format Version 12.00↓  
# Visual Studio 14↓  
VisualStudioVersion = 14.0.25123.0↓  
MinimumVisualStudioVersion = 10.0.40219.1↓
```

- 将 Format Version 12.00 改为 11.00;
- 将# Visual Studio 14 改为 2010;
- 将 VisualStudioVersion = 14.0.25123.0 改为 10。

Format Version(解决方案文件版本)和 VS 版本的对应关系：

```
-Visual Studio 2005 - Format Version 9.00  
  
-Visual Studio 2008 - Format Version 10.00  
  
-Visual Studio 2010 - Format Version 11.00  
  
-Visual Studio 2012 - Format Version 12.00  
  
-Visual Studio 2013 - Format Version 12.00  
  
-Visual Studio 2015 - Format Version 12.00
```

### 2. 修改项目文件

使用记事本打开.vcxproj 文件：

```
<?xml version="1.0" encoding="utf-8"?>↓  
<Project DefaultTargets="Build" ToolsVersion="14.0"
```

将 ToolsVersion="14.0" 改为 4.0。

ToolsVersion(.Net Framework 版本)与 VS 版本的对应关系：

- Visual Studio 2008 - ToolsVersion 3.5
- Visual Studio 2010 - ToolsVersion 4.0
- Visual Studio 2012 - ToolsVersion 4.0
- Visual Studio 2013 - ToolsVersion 12.00
- Visual Studio 2015 - ToolsVersion 14.00

将 v140 改为 v110，按照如图更改，一般来说都是 4 个。（只需要搜索“PlatformToolset”关键字即可，并不一定有确定的修改个数）

工具集版本号与 VS 版本的对应关系：

- Visual Studio 2015: V140
- Visual Studio 2013: V120
- Visual Studio 2012: V110
- Visual Studio 2010: V100
- Visual Studio 2008: V90
- Visual Studio 2005: V80

通过以上设置后就可以在低版本 VS 中正常使用了。

## 1.2 Qt5.9.1 及 Qt VS Tool 安装

### 1. QT5. 9. 1

安装时，记得把“msvc2015”和“msvc2015\_64”安装上（图 1-1 中圈红部分）。

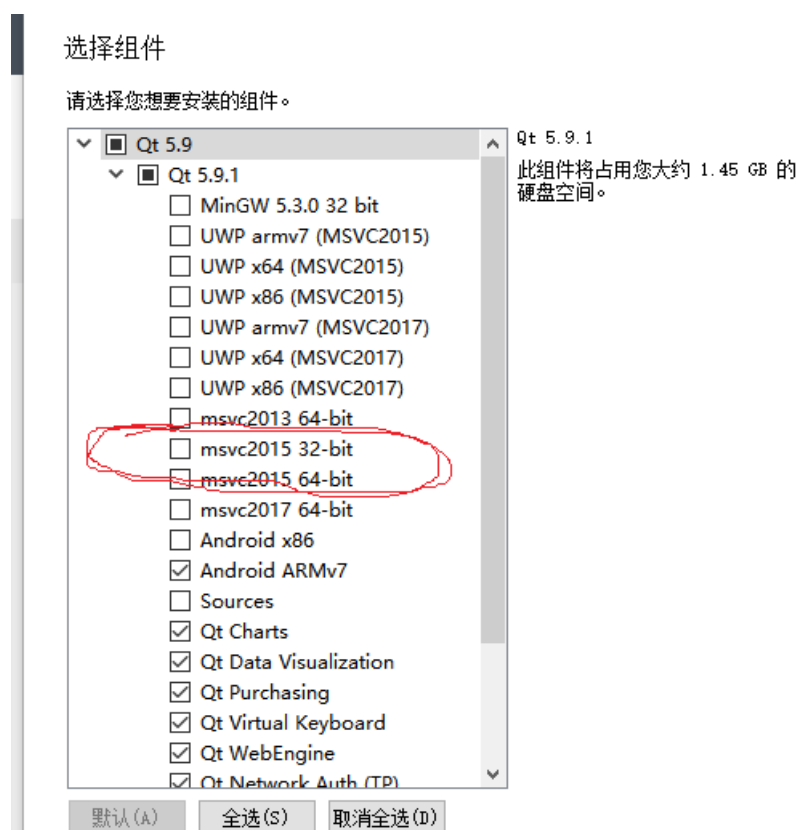


图 1-1 Qt5. 9. 1 安装界面

### 2. VS2015

安装时，一定要选择“visual c++”选项（图 1-2 中圈红部分）。

### 3. Qt VS Tools

（1）当安装完 VS2015 后，接下来安装 Vs 集成 Qt 的工具“Qt VS Tools”。

打开 VS2015 后，在工具栏找到“工具(T)”，在“工具”子菜单栏找到“扩展和更新(U)…”并打开。点击“联机”，在右边搜索框中输入“Qt”，

会出现图 1-3 所示的搜索结果。可以安装第一个“Qt VS Tools”或第三个“Qt Package”。

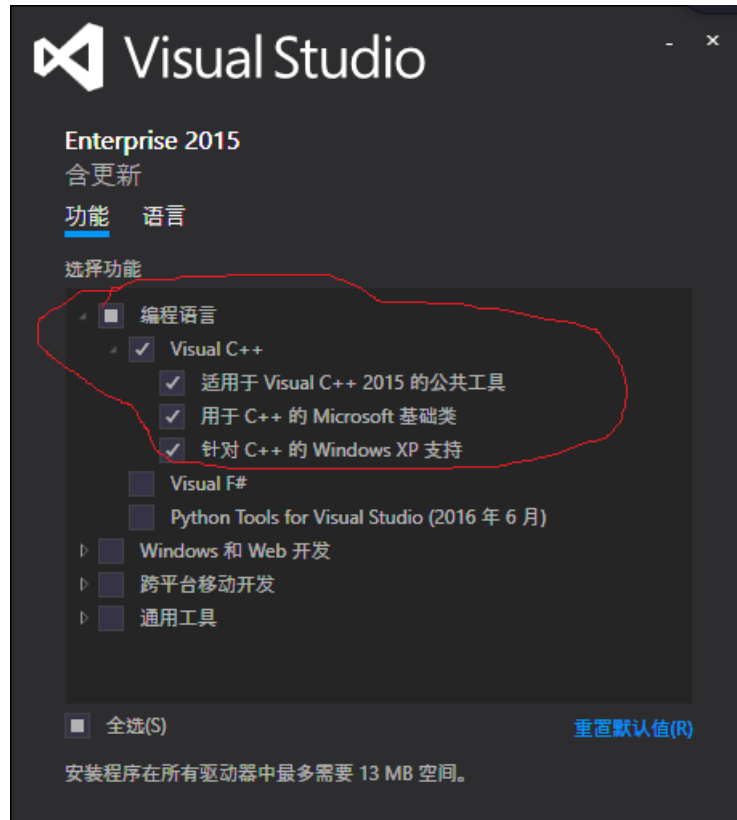


图 1-2 VS2015 安装界面

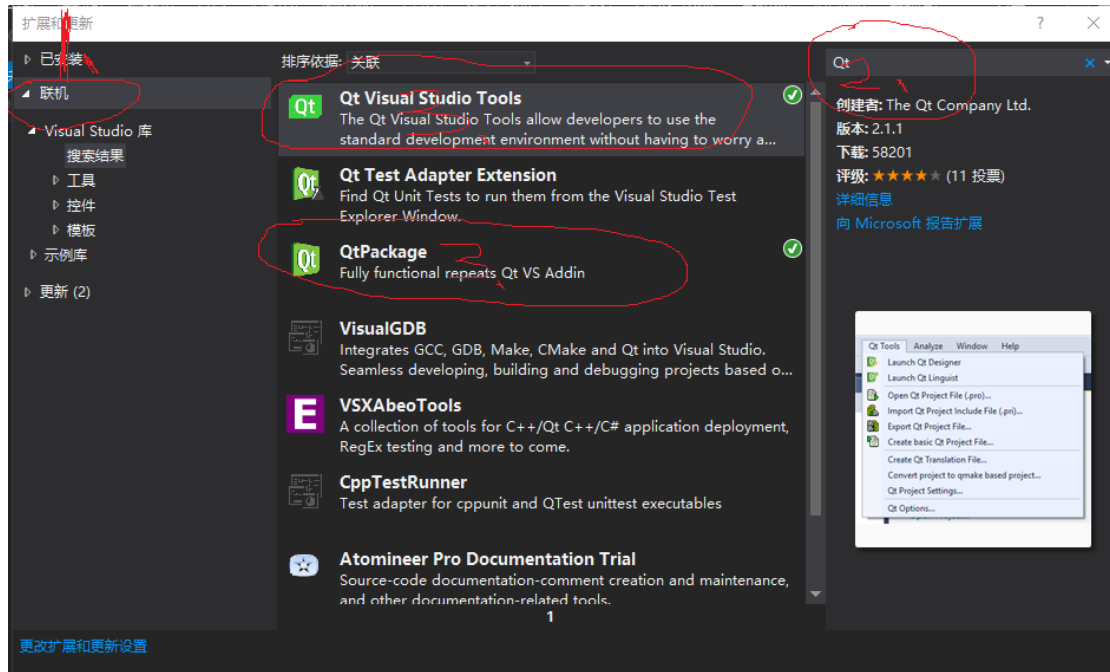


图 1-3 Qt VS Tools 安装界面

## (2) Qt VS Tools 配置

当安装完“Qt VS Tools”后，重启 VS2015 后，会发现在工具栏多了一个“Qt VS Tools”（安装了“Qt VS Tools”）或“QT5”（安装了 QtPackage）工具。这两个工具接下来的操作基本相同。下面以“Qt VS Tools”为例进行说明。点击“Qt VS Tools”中的“Qt Options”选项，在“Qt Versions”中，点击“Add”，在弹出的对话框中（如图 1-4 所示），分别添加 Qt5.9.1 中“msvc2015”和“msvc2015\_64”的安装位置，保存即可。

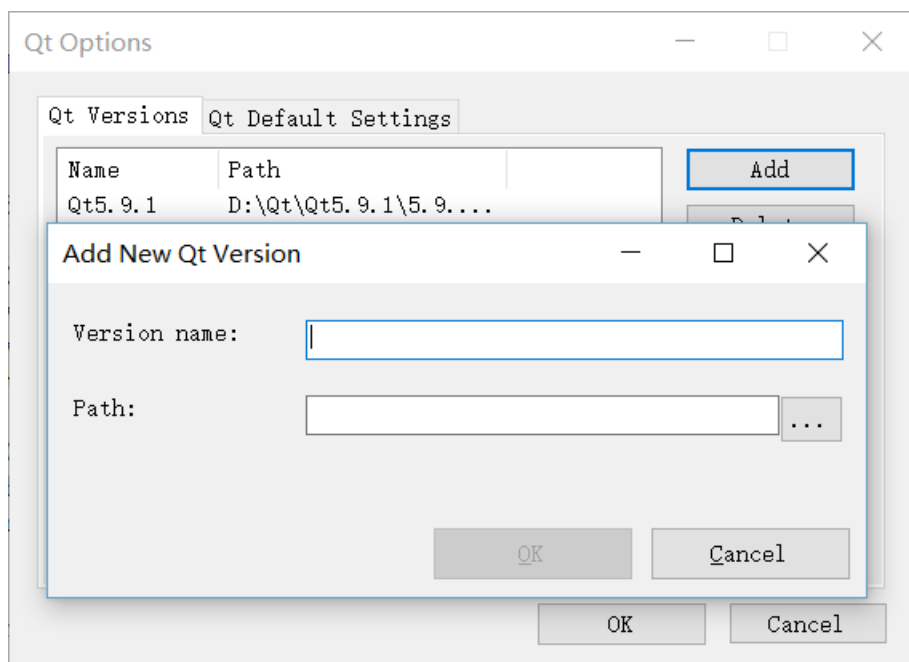


图 1-4 添加 Qt 到 Qt VS Tools

### (3) 添加环境变量

在完成上述操作后，还需要将 Qt5.9.1 和 VS2015 的编译器位置添加到系统变量的“Path”中。打开“控制面板”->“系统和安全”->“系统”中的“高级系统设置”，点击“环境变量”，在“系统变量”中找到“Path”并打开。分别将 VS2015 安装目录下的“VS2015 安装根目录\VC\bin”和 Qt5.9.1 安装目录下的“Qt5.9.1 安装根目录\msvc2015\bin”（32 位）及“Qt5.9.1 安装根目录\msvc2015\_64\bin”（64 位）添加到 Path 中。



## 二 项目实施

本项目实现了一个针对静态图片中车牌检测分割, 分割字符, 识别字符的方法。提出一种基于车牌特征信息分析的车牌识别方法, 它充分利用车牌定位和字符分割过程中得到的信息对车牌识别过程进行反馈, 将二值化、车牌定位和字符分割紧密结合, 注重车牌与车辆背景图像分离特征, 以连通域分析为车牌分割特点, 结合局部二值化算法, 提高正确率。使用投影法作为字符分割的基础进行车牌字符分割。使用模板匹配的方法进行字符识别。

### 2.1 车牌定位分割

#### 2.1.1 图像灰度化

图像灰度化是将彩色图像中的彩色信息剔除, 只包含亮度信息。计算机中表示灰度图是把亮度值进行量化等分成 到 255 个级别, 最暗 (全黑), 255 最亮 (全白), 而在 RGB 模型中, 如果  $R=G=B$ , 则颜色  $(R, G, B)$  就表示灰度色 [4]。图像灰度化方法很多, 比如平均值法、加权平均化、最大值化。在 OpenCV 函数库中, 灰度化是通过 `cvCvtColor (const CvArr* src, CvArr* dst, int code)` 函数实现的, 其中 `src` 为原彩色图像, `dst` 为处理后图像, `code` 为色彩空间转换方式, 在这里 `code` 定义为 `CV_RGB2GRAY`。

#### 2.1.2 边缘检测

边缘信息是重要的图像特征信息。因此, 边缘检测的方法研究是图像分析和识别领域中一个十分引入关注的课题。传统的边缘检测算子如 Sobel、Prewitt、Roberts 和 Kirch 等算子, 大部分为局部窗口梯度算子。由于它们对噪声很敏感, 所以对处理实际图像不太实用。Canny 算子是一类具有优良性能的边缘检测算子, 它在许多图像处理领域得到广泛应用, 本文采用 Canny 算子, 其基本思路是: 先对处理的图像选择一定的高斯滤波器进行平滑滤波, 然后采用一阶偏导的有限差分计算梯度的幅值和方向, 对梯度幅值进行非极大值抑制, 然后用双阈值算法检测和连接边缘, 最后采用高斯平滑后, 得到所需边缘图像。在 OpenCV 函数库中, `cvCanny` 采用 Canny 算法寻找输入图像的边缘并在输出图像中标记这些边缘。`void cvCanny (const CvArr* image, CvArr* edges, double threshold1, double threshold2, int aperture_size=3);`

image: 输入图像, edges: 输出的边缘图像, thresholds1: 第一个阈值, thresholds2: 第二个阈值, aperture\_\_size: Sobel 算子内核大小。thresholds1 和 thresholds2 中的小阈值用来控制边缘连接, 大阈值用来控制强边缘的初始分割。

车牌含有 7 个字符, 针对车牌的这种特点对边缘检测后的图像进行光栅扫描。将一行中白色的点少于 20 个, 一列中白色的点少于 2 个的行或者列均置为黑色。对图像进行邻域运算, 获取 3\*3 模板范围内白色的点的个数, 若小于某一阈值, 则将此点置为黑色。这样可以消除大部分噪音点的干扰。

### 2.1.3 车牌分割

使用种子法标定图像的单连通区域, 利用车牌特征去除干扰单连通区。车牌的宽度定义在了 100-260 之间, 并且宽高比 $\sim 3.3$ , 误差在 $-0.6\sim 0.6$ 之间, 即宽高比大约为 $3.9\sim 3.3:1$ 。

使用 HIS 颜色空间消除其余近似形状连通区域的干扰。将连通区域中蓝色区域最多的一个作为车牌区域。

在灰度图中将车牌大致区域分割出来, 使用线性插值的方法进行图像尺寸归一化。使用水平梯度和垂直梯度进行图像的精确定位。具体方法如下:

计算图像水平一阶差分累加值和垂直一阶差分累加值, 分别求出累加值的最大值。水平方向从波峰开始向两边扫描, 当搜索到累加值开始变大且小于 0.6 倍峰值大小时, 停止扫描, 此时找到波峰两边的波谷。将较小值作为标准在另一侧继续搜索, 找到比最小的值更小的值为止。这两行作为车牌区域的上下限。垂直方向从第一列开始, 找到第一个波峰且大于 0.5 倍峰值大小作为车牌区域左边界扫描起始点, 继续向右扫描, 找到最后一个满足条件的波谷作为车牌区域的右边界扫描起始点。从左右边界扫描起始点分别向左右开始扫描, 找到第一个波谷为止。

车牌大致区域图像, 一阶差分可视化图像及车牌精确定位图像如图 2-1, 图 2-2, 图 2-3, 图 2-4 所示。



图 2-1 使用连通区域分割的车牌



图 2-2 车牌水平一阶差分图像



图 2-3 车牌垂直一阶差分图像



图 2-4 精确定位后的车牌区域

#### 2.1.4 滤波及二值化

使用高斯滤波滤除部分噪声。使用大津法进行图像的二值化（如图 2-5 所示）。



图 2-5 车牌区域二值化

## 2.2 字符分割

字符分割采用垂直投影的方法。首先将图像尺寸归一化为 409\*90。

### 2.2.1 垂直投影

将二值化图像每一列目标点累加求得每一列目标点个数。垂直投影图如 2-6 所示。



图 2-6 垂直投影图

### 2.2.2 寻找分隔符

设  $m$  为分隔符宽度，初始值为 34，若给的范围太宽，逐渐减小宽度

搜索范围为 START(57)→END(181)。为容错不同的图像，稍微扩大搜索范围，设定搜索范围在第 2 个字符的起始位置到第 3 个字符的结束位置。

分隔符的模板为 (3, 3, 15, 15, ..., 15, 15, 3, 3)，长度为  $m$ 。当缩小  $m$  时，将最后两个 3 前移。

### 2.2.3 分割字符

以分割符为基准，将分隔符左边界作为第二个字符右边界，分隔符右边界作为第三个字符左边界。

分隔符左边界向左 45-57 的范围内寻找最小值，这个范围是归一化图像后第一个字符及第二个字符的间隙。在最小值处分别向左右寻找第一个字符的右边界和第二个字符的左边界。第一个字符的左边界与第二个字符左边界寻找方法一致。同理，可找到后面 5 个字符的左右边界。将字符左右边界在二值图中画出（如图 2-7 所示）。



图 2-7 字符分割图

## 2.3 字符识别

使用模板匹配算法进行字符识别。

读取模板图片并进行水平投影和垂直投影。将每一个模板与字符图像进行比较，差值最小的模板被认为是该字符的值。