



车牌智能识别

目录

| | |
|---|--------|
| 1 | 案例背景 |
| 2 | 车牌定位 |
| 3 | 字符分割 |
| 4 | 车牌识别模型 |
| 5 | 模型评估 |
| 6 | 拓展思考 |

案例背景

- 高速公路的堵点基本集中在收费站。为了实现不停车快捷收费，减少拥堵，从2018年5月起，国务院就推出了取消高速公路省界收费站的决策部署，并将在两年内基本取消全国高速公路省界收费站，加快推进ETC实施，由人工收费逐步转向电子收费。
- 传统高速公路收费站主要采取人工收费，每辆车至少要半分钟的时间缴费，且收费员需要24小时在岗，既耗时又消耗巨大的人力成本。
- 从人工收费到无感收费时代。



案例背景

无感收费

- 两种方式：ETC与基于车牌识别的无感支付
- ETC采用无线通信信号及相关设备相感应的方式自动识别车牌信息，（高速公路领域）
 - 优点：ETC在车牌识别率方面更高、更准确，车牌识别率可高达99.99%；不受车辆行驶速度、周围环境、天气等因素影响，识别快速、准确、稳定
 - 缺点：所需要的成本相对更高；ETC还存在邻道干扰以及清分结算系统开放性和个性化服务问题；ETC停车申请与提现问题；ETC需要办理信用卡
- 车牌识别则采用高清摄像头高清拍照识别车牌，（停车场景领域）
 - 优点：安装方便，在停车场景中运营机制和生态的成熟开放程度高，可以直接采用支付宝与微信接口，且提现方便
 - 缺点：识别率相对较低，为95%左右，且容易受雨雪、大雾、风暴等天气、环境、车速等因素影响，同时存在套牌等风险。

案例背景

车牌识别

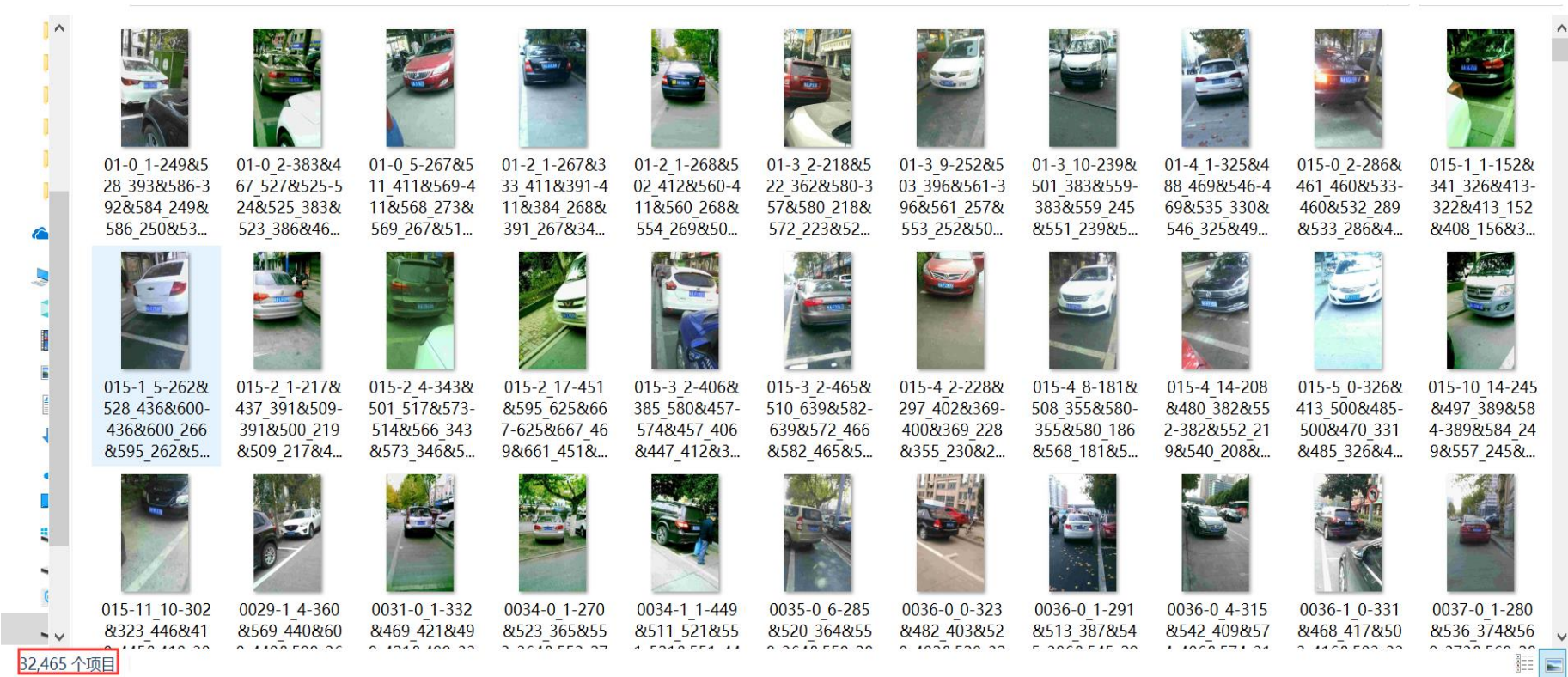
- 汽车车牌识别技术是车辆检测系统中的一个重要环节，它在交通监视和控制中占有很重要的地位，有着多种应用，例如自动收费系统、不停车缴费、失窃车辆的查寻、停车场车辆管理、特殊车辆的出入控制等等。
- 同时，汽车车牌识别的方法还可应用到其他检测和识别领域，所以汽车车牌识别问题已经成为现代交通工程领域中研究的重点和热门问题之一。



案例背景

数据说明

- 提供的案例数据集是在合肥市的停车场采集到的32,465张图片（2.46 GB），采集时间从早上7:30到晚上10:00，每种图片大小720x1160x3。



案例背景

数据说明

- 采集到的每张车牌图像的名称包含数据标注，如名称为025-95_113-154&383_386&473-386&473_177&454_154&383_363&402-0_0_22_27_27_33_16-37-15.jpg的图片，可以由分隔符'-'分为以下7个部分：
 - 025为区域，
 - 95_113 对应两个角度, 水平95°, 竖直113°
 - 154&383_386&473对应边界框坐标:左上(154, 383), 右下(386, 473)
 - 386&473_177&454_154&383_363&402对应四个角点坐标
 - 0_0_22_27_27_33_16为车牌号码，第一个为省份, 后面的为字母和文字（顺序如下：）
 - provinces = ["皖", "沪", "津", "渝", "冀", "晋", "蒙", "辽", "吉", "黑", "苏", "浙", "京", "闽", "赣", "鲁", "豫", "鄂", "湘", "粤", "桂", "琼", "川", "贵", "云", "藏", "陕", "甘", "青", "宁", "新", "警", "学", "O"] ads = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'J', 'K', 'L', 'M', 'N', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', '0', '1', '2', '3', '4', '5', '6', '7', '8', '9']
 - 37亮度
 - 15模糊度

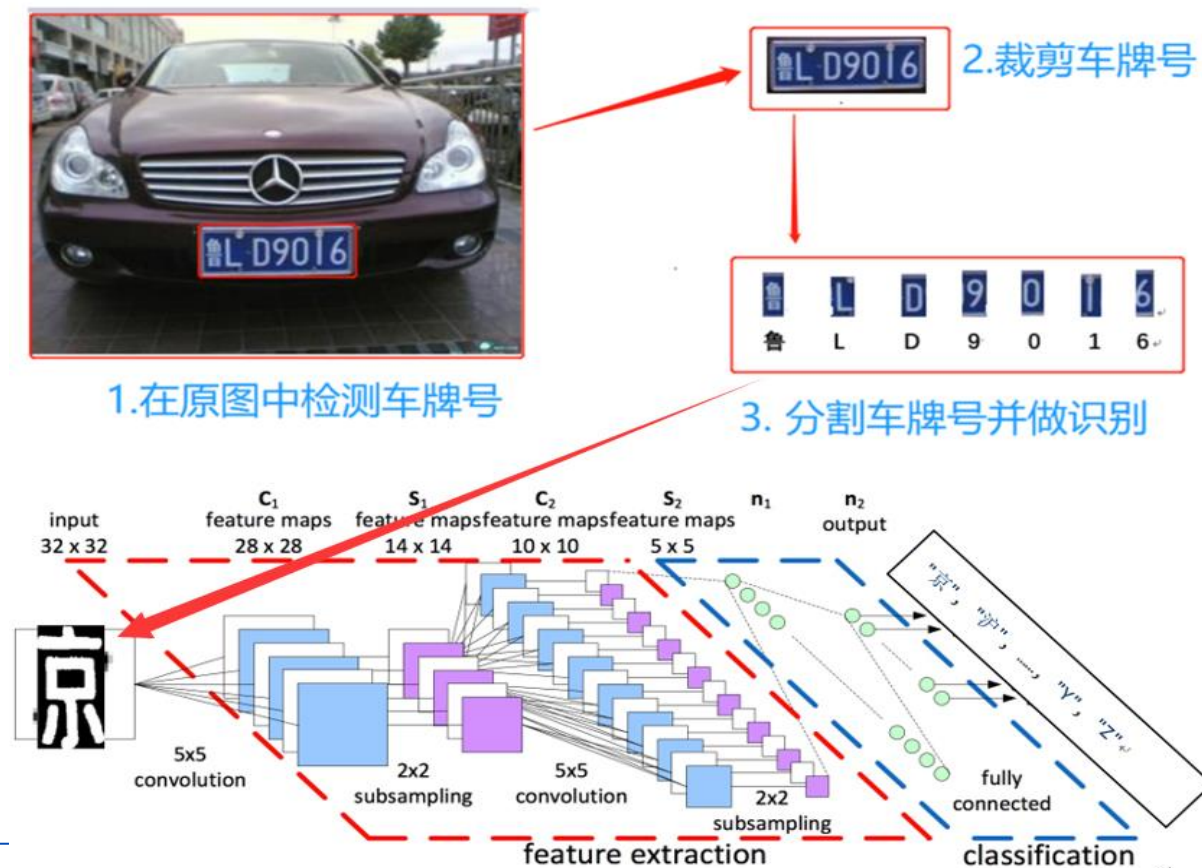
案例背景

挖掘目标

1. 通过采集到的图片进行车牌定位，获取车牌所在位置并进行截取。
2. 对截取的车牌位置图片进行字符分割。
3. 构建车牌识别模型。

案例背景

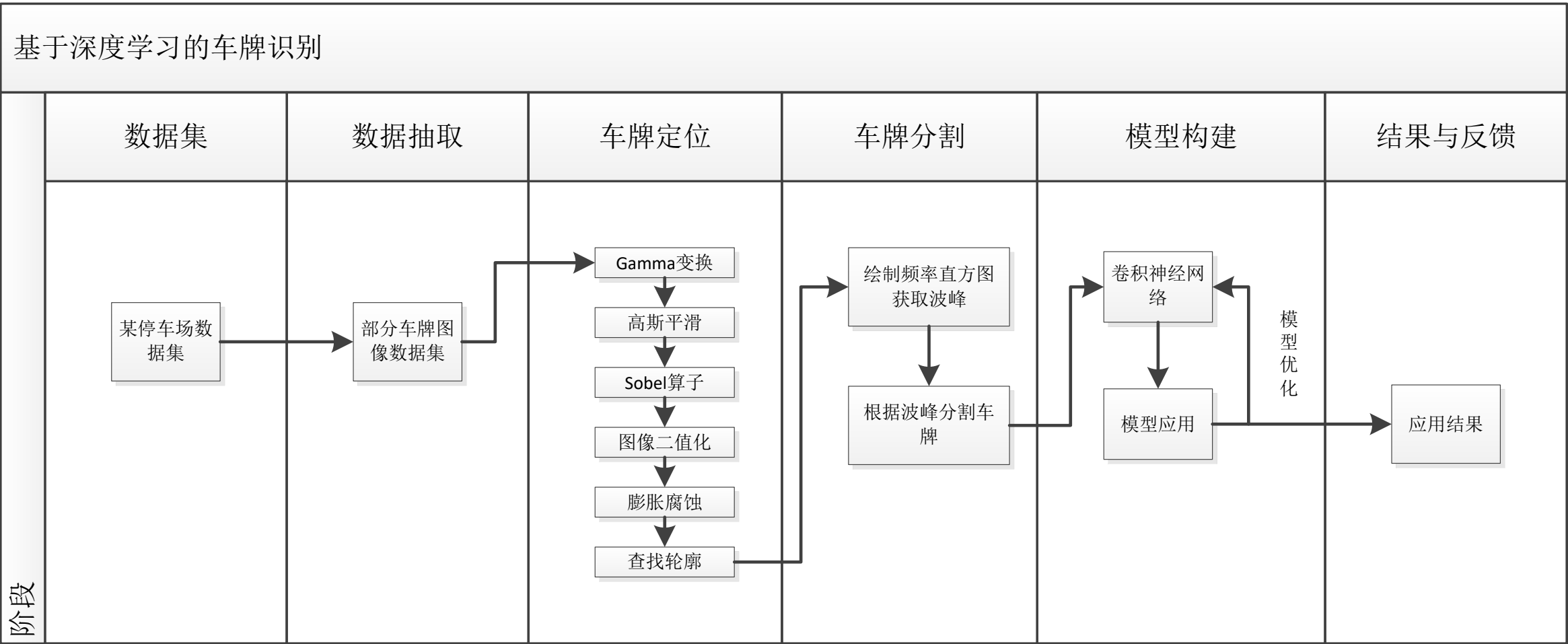
- 本案例详细描述了基于深度学习的车牌识别模块的主要实现流程，涉及“图像预处理——车牌定位——字符分割——字符识别——结果输出”等一系列流程，复现了车牌识别项目的详细过程，其中的技术关键在于“车牌定位，字符分割和字符识别”环节。



目录

| | |
|---|--------|
| 1 | 案例背景 |
| 2 | 车牌定位 |
| 3 | 字符分割 |
| 4 | 车牌识别模型 |
| 5 | 模型评估 |
| 6 | 拓展思考 |

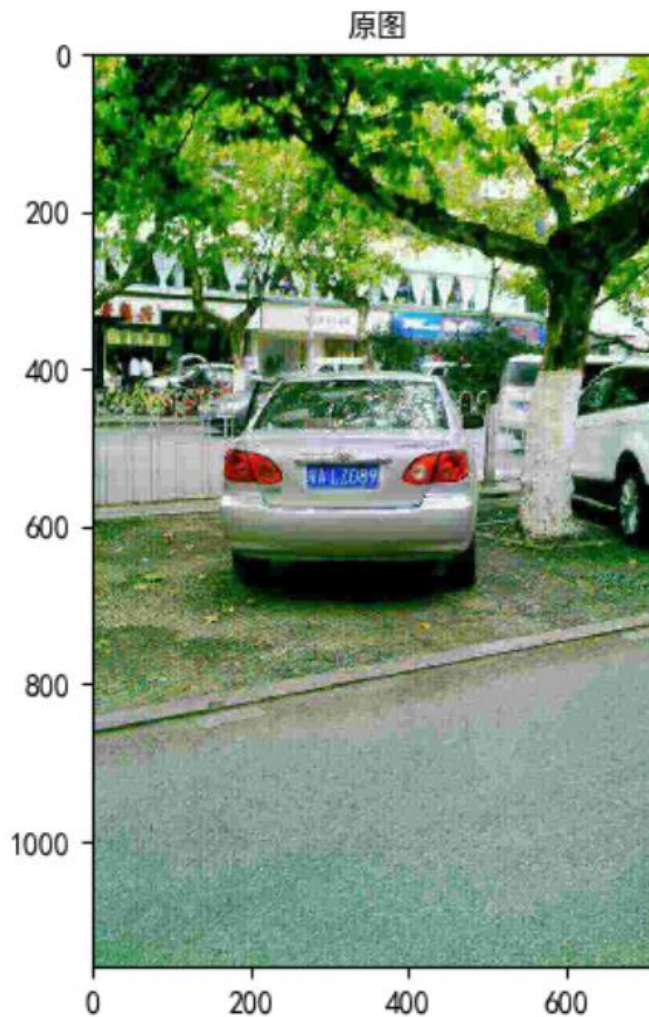
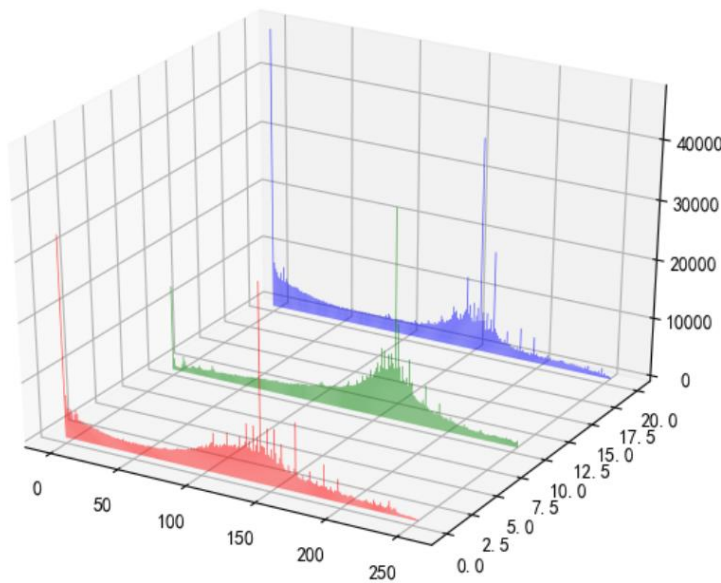
车牌定位



图片预处理

读取原始图片

- 为展示对图片的处理效果，以 “0034-0_1-270&523_365&553-364&553_270&553_271&523_365&523-0_0_10_23_24_32_33-90-17.jpg” 为例。
- 例图每个通道的直方图如右所示，如果直方图的成分接近0或者255，可能出现暗部细节不足或亮部细节丢失的情况。



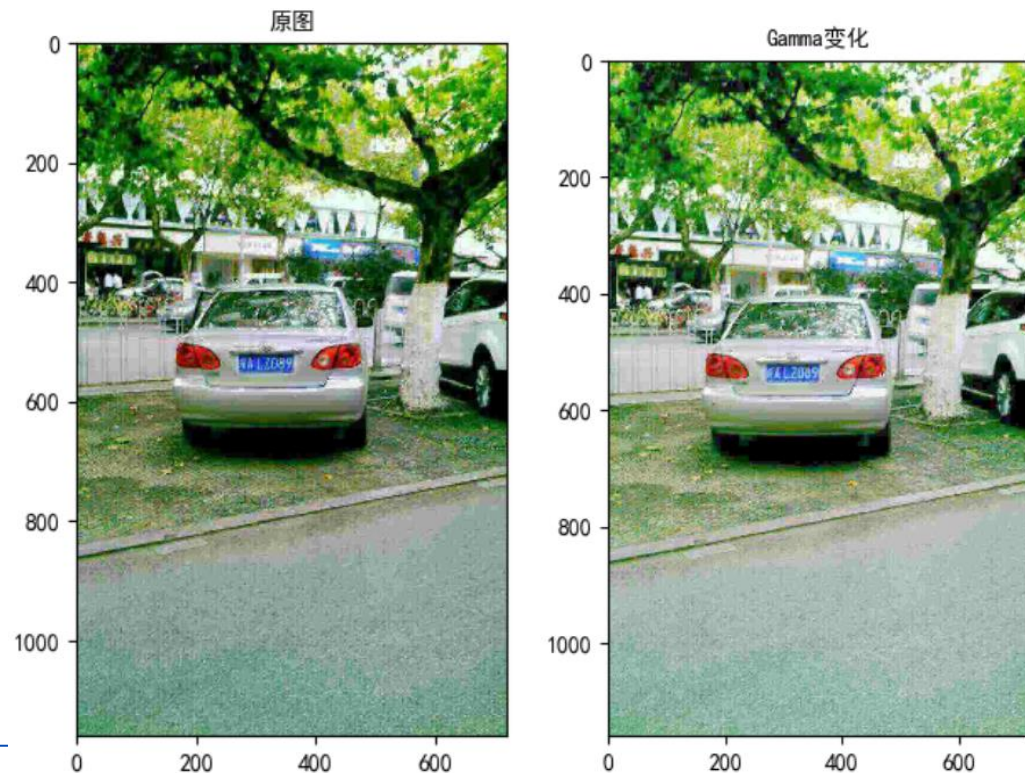
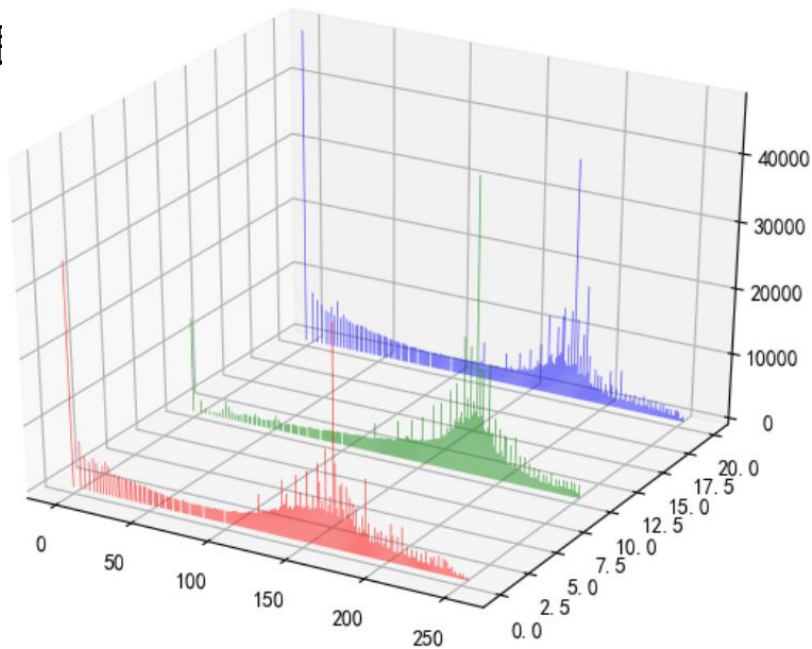
图片预处理

Gama变化——提高暗部细节

- Gamma变换是对输入图像灰度值进行的非线性操作，使输出图像灰度值与输入图像灰度值呈指数关系，

$$V_{out} = A V_{in}^{\gamma}$$

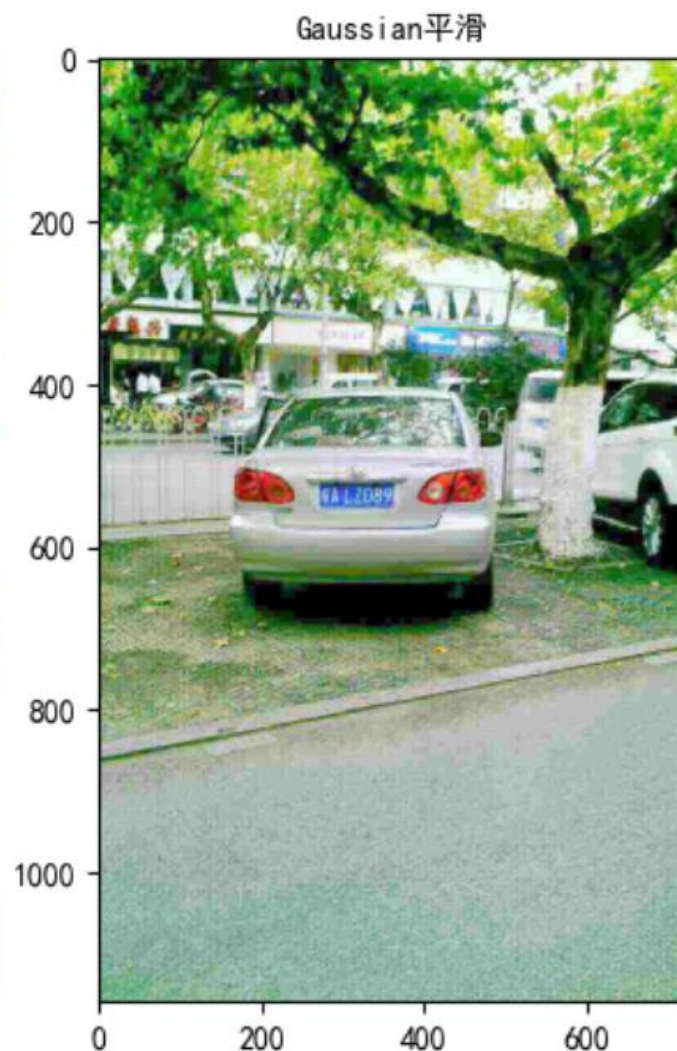
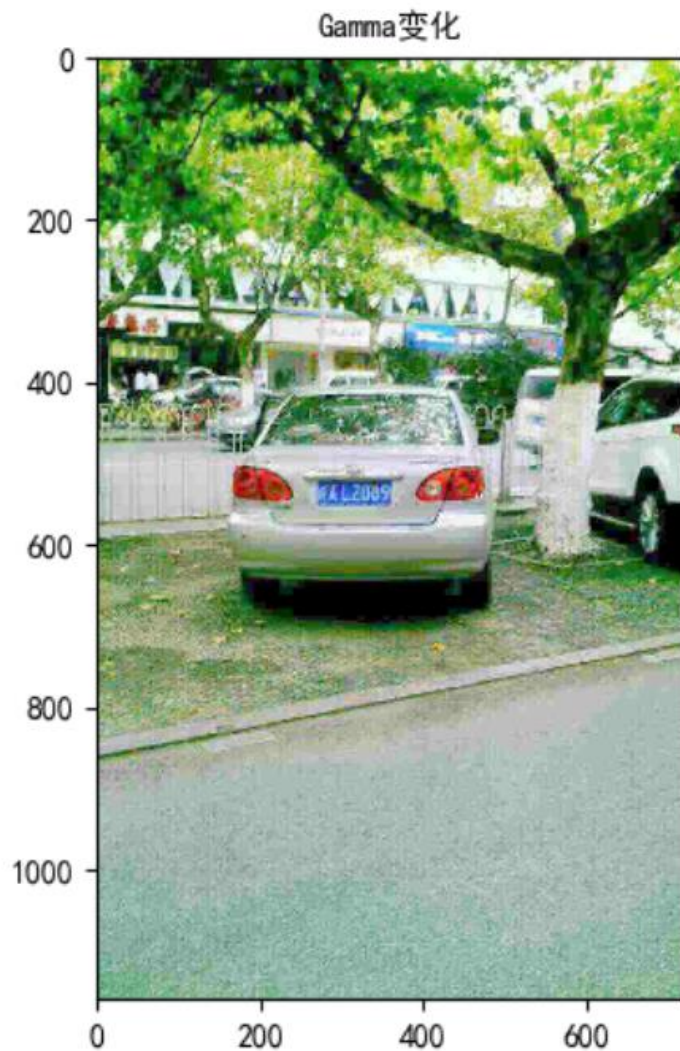
- 这个指数 γ



图片预处理

高斯平滑——降噪操作

- 在尽量保留原图信息的情况下，过滤掉图片内部的噪声，这一过程称为对图像的平滑处理，所得图像称为平滑图像。平滑处理会对图像中与周围像素点的像素值差异较大的像素点进行处理，将其值调整为周围像素点的近似值。
- 常用方法有：均值滤波、方框滤波、高斯滤波、中值滤波等。在这里，使用的是高斯滤波。



图片预处理

中值平滑——降噪操作

- 针对左图中第3行第3列的像素点，计算其中值平滑后的像素值。将其邻域设置为 3×3 大小，对于其 3×3 邻域内像素点进行排序（升序降序均可），按升序排序后得到的序列为：[1, 17, 32, 33, 43, 66, 66, 93, 95]，得到的中间值为43，因此用43替换原来的像素值1作为新的像素值，处理结果如右图所示。

| | | | | |
|-----|----|----|----|-----|
| 55 | 45 | 22 | 45 | 60 |
| 123 | 17 | 66 | 33 | 23 |
| 47 | 43 | 1 | 32 | 8 |
| 68 | 66 | 93 | 95 | 171 |
| 23 | 76 | 56 | 45 | 9 |

| | | | | |
|-----|----|----|----|-----|
| 55 | 45 | 22 | 45 | 60 |
| 123 | 17 | 66 | 33 | 23 |
| 47 | 43 | 43 | 32 | 8 |
| 68 | 66 | 93 | 95 | 171 |
| 23 | 76 | 56 | 45 | 9 |

图片预处理

高斯平滑——降噪操作

- 针对左图中第3行第3列的像素点，计算其高斯平滑后的像素值。将其邻域设置为3×3大小，此时，高斯平滑处理对应的权重不再都是1，计算公式为：
- 新的像素值 = $(144 \times 0.05 + 178 \times 0.1 + 133 \times 0.05) + (119 \times 0.1 + 23 \times 0.4 + 240 \times 0.1) + (166 \times 0.05 + 93 \times 0.1 + 95 \times 0.05) = 99$
- 得到新的像素矩阵如右图所示。

| | | | | | |
|---|---|---|------|-----|------|
| 1 | 2 | 1 | 0.05 | 0.1 | 0.05 |
| 2 | 8 | 2 | 0.1 | 0.4 | 0.1 |
| 1 | 2 | 1 | 0.05 | 0.1 | 0.05 |

左图为高斯平滑中3×3权重示例，
右图为实际计算时使用的权重

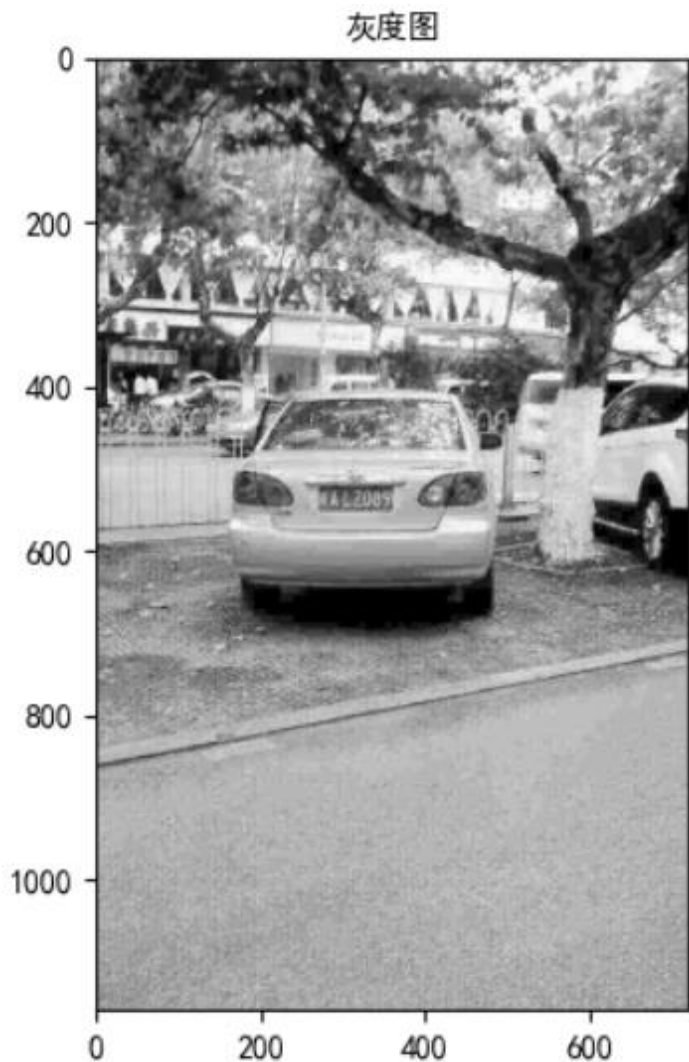
| | | | | |
|-----|-----|-----|-----|-----|
| 55 | 45 | 22 | 45 | 60 |
| 123 | 144 | 178 | 133 | 23 |
| 47 | 119 | 23 | 240 | 8 |
| 68 | 166 | 93 | 95 | 171 |
| 23 | 76 | 56 | 45 | 9 |

| | | | | |
|-----|-----|-----|-----|-----|
| 55 | 45 | 22 | 45 | 60 |
| 123 | 144 | 178 | 133 | 23 |
| 47 | 119 | 99 | 240 | 8 |
| 68 | 166 | 93 | 95 | 171 |
| 23 | 76 | 56 | 45 | 9 |

图片预处理

灰度处理——加快运算速度

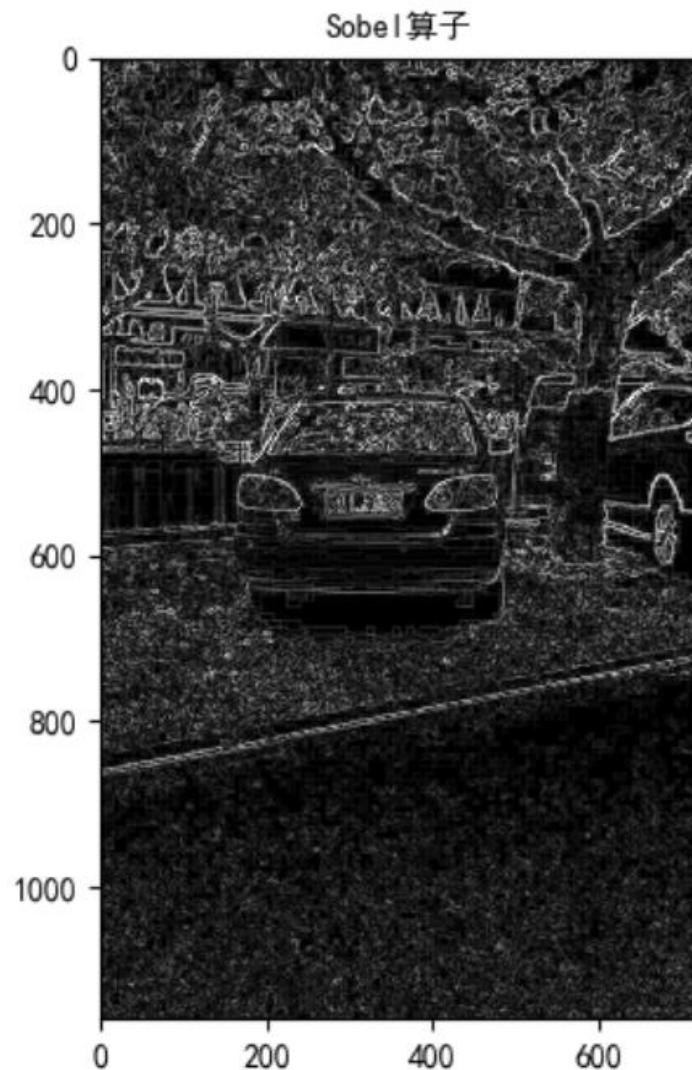
- 通常情况下，计算机会将灰度处理为256个灰度级别，取值区间为[0, 255]，其中数值255表示纯白色，数值0表示纯黑色，其余的数值表示纯白色到纯黑色之间不同级别的灰度。用灰度表示的图像称作灰度图。除了常见的卫星图像、航空照片外，许多地球物理观测数据也以灰度表示。
- 车牌识别过程中车牌颜色对识别不起作用，所以为了可以加快运算速度，将定位到的车牌图像进行灰度处理。



图片预处理

Sobel算子——计算图像边缘信息

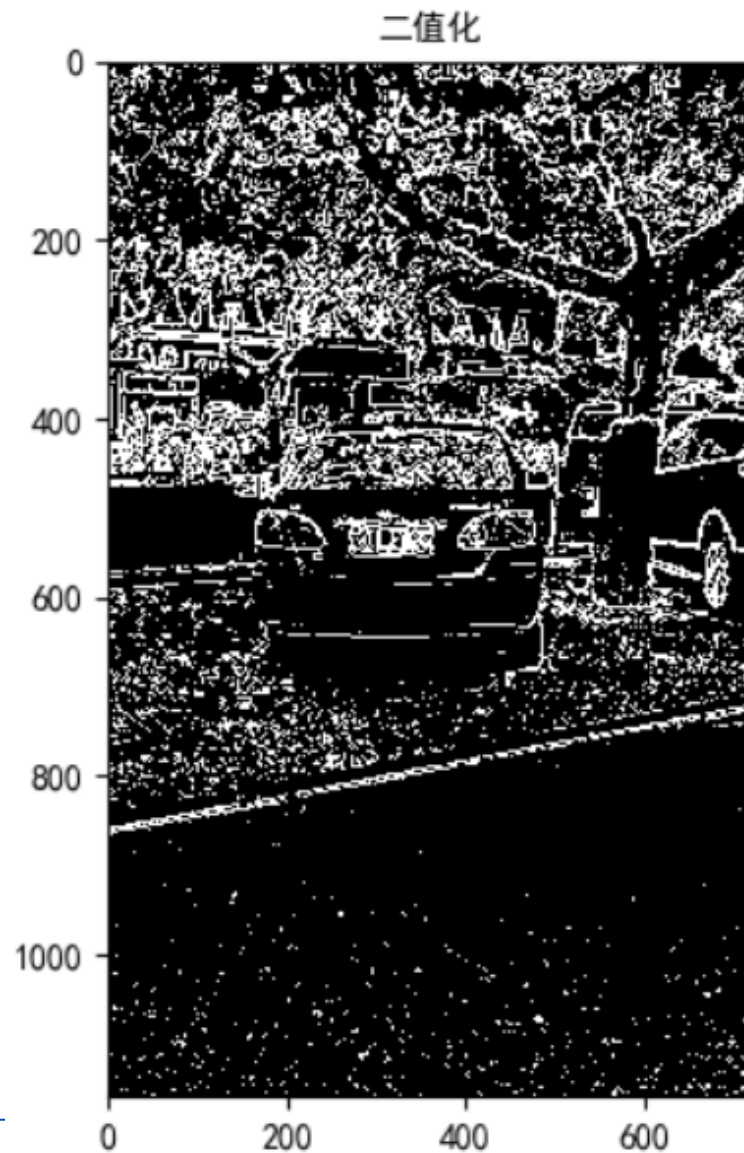
- Sobel算子是一种离散的微分算子，该算子结合了高斯平滑和微分求到的运算，利用局部差分寻找边缘，实现边缘信息计算。



图片预处理

图像二值化——凸显轮廓

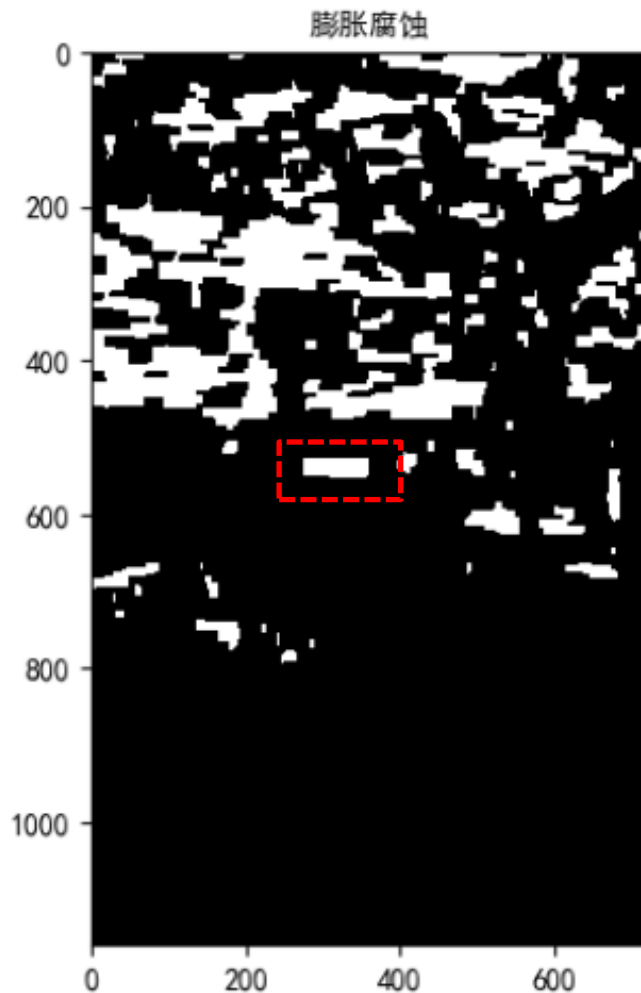
- 图像二值化（Image Binarization）就是将图像上的像素点的灰度值设置为0或255，也就是将整个图像呈现出明显的黑白效果的过程，能够让图像中数据量大为减少，从而能凸显出目标的轮廓。



图片预处理

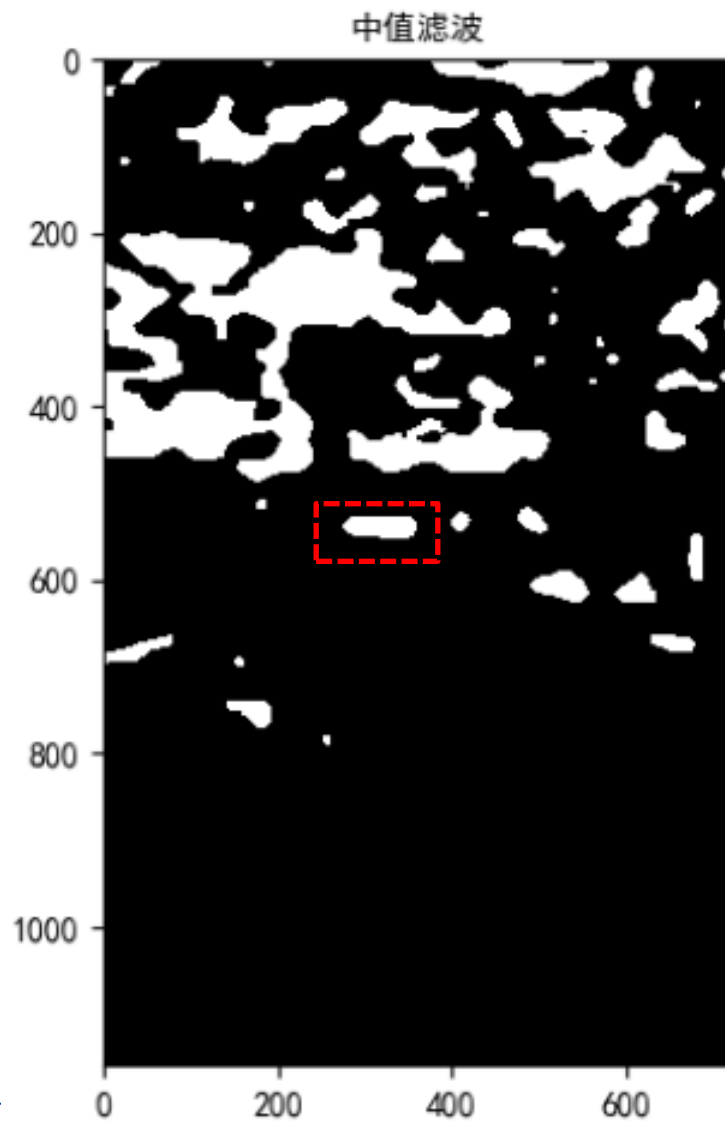
膨胀腐蚀——形态学特征

- 形态学是图像处理中一个非常重要的研究方向，主要从图像内提取分量信息，该分量信息通常对于表达和描绘图像的形状具有重要意义，通常是图像理解时使用的最本质的形状特征。
- 形态学操作包括：腐蚀、膨胀、开运算、闭运算、顶帽运算、黑帽运算等。



图片预处理

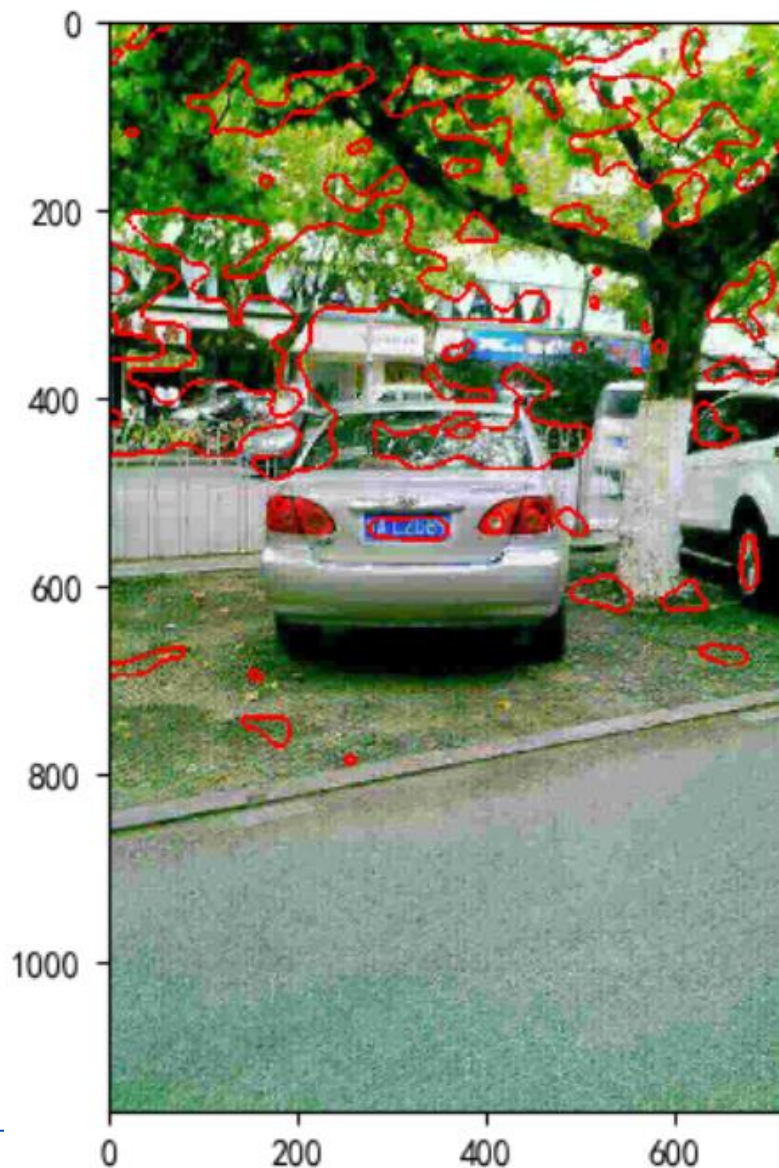
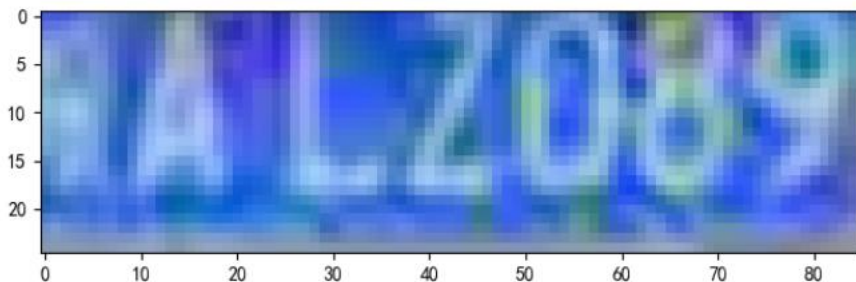
中值滤波——降噪操作



图片预处理

绘制轮廓——定位车牌的位置

- OpenCV中提供cv2.findContours函数能够查找轮廓，并根据参数返回特定表达方式的轮廓（曲线）。
- 此外，还提供cv2.drawContours函数将查找到的轮廓绘制到图像上，该函数可以根据参数在图像上绘制不同样式（实心、空心，以及线条的不同粗细、颜色等）的轮廓，可以绘制全部轮廓或特定轮廓。



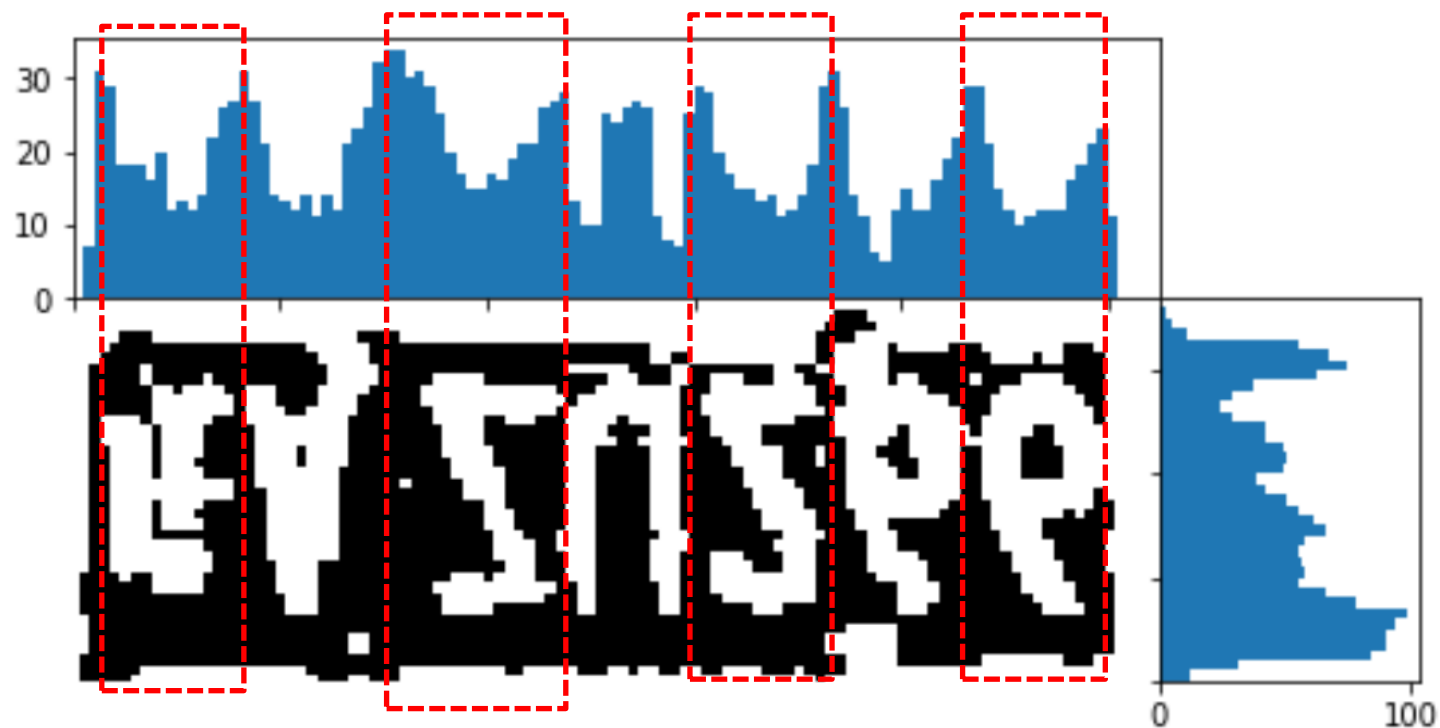
目录

| | |
|---|--------|
| 1 | 案例背景 |
| 2 | 车牌定位 |
| 3 | 字符分割 |
| 4 | 车牌识别模型 |
| 5 | 模型评估 |
| 6 | 拓展思考 |

字符分割

车牌峰度分割

- 车牌分割主要采用的是基于二值化图像0的频率峰度分割车牌的方法。



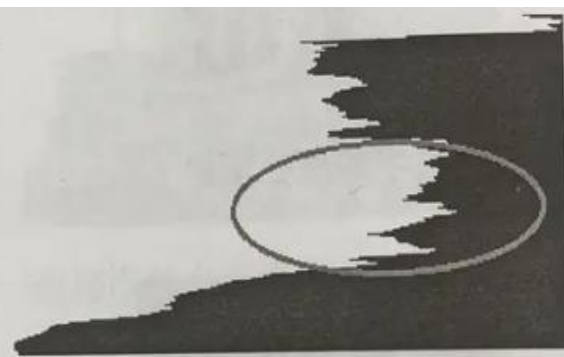
字符分割

方向纠正

- 在车牌图片倾斜的情况下，得到的投影峰谷模糊，不利于后续分割车牌字符
- 标准情况下的车牌投影峰谷鲜明，故进行车牌字符分割前，先对车牌进行倾斜图片纠正操作。



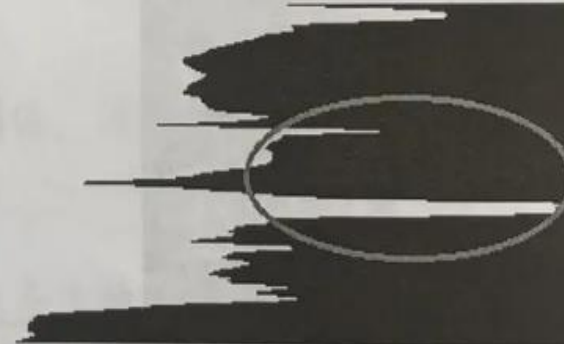
(a)倾斜的车牌



(b)倾斜车牌的投影



(c)标准的车牌

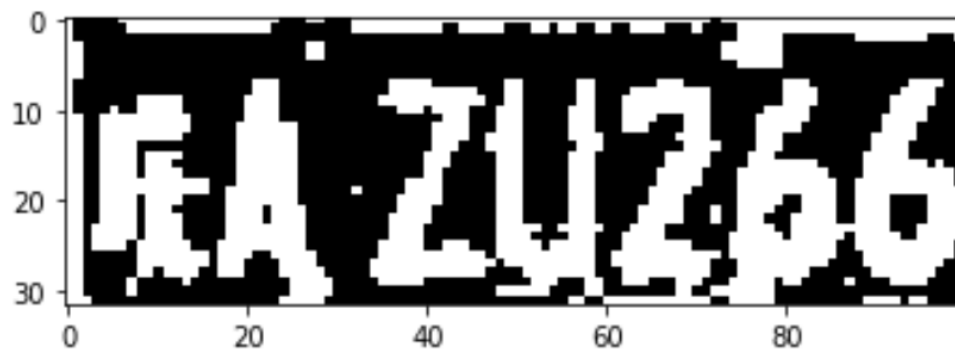
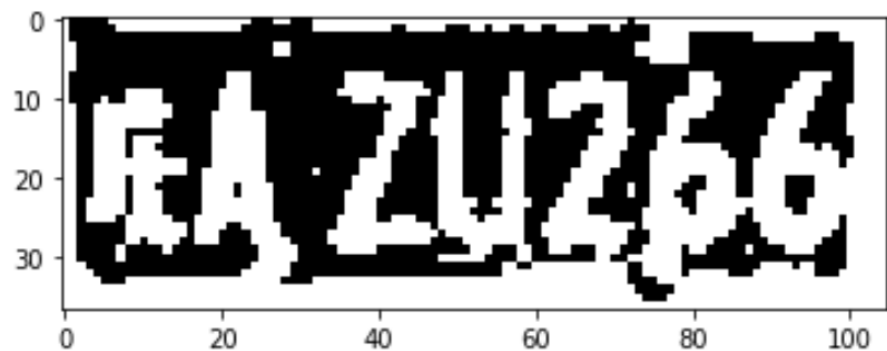


(d)标准车牌的投影

字符分割

边框处理

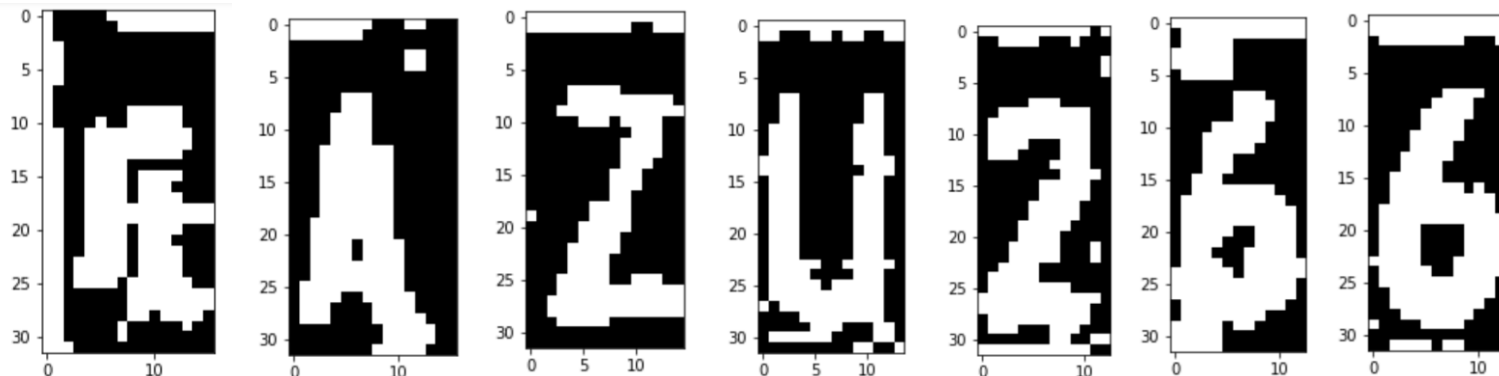
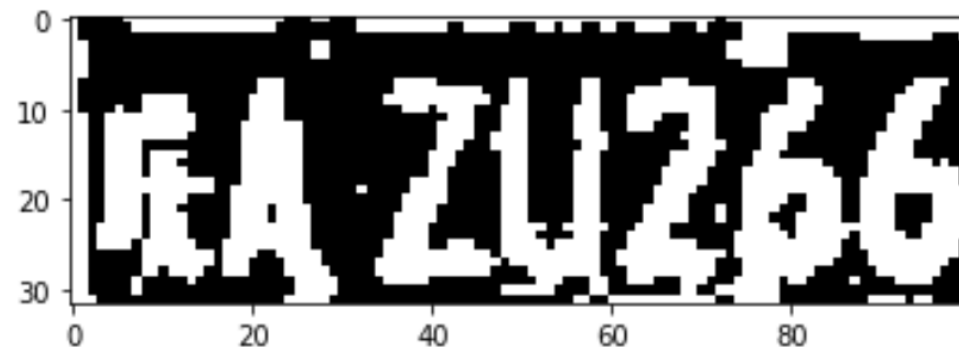
- 为避免车牌图片的边框给车牌图片的峰谷计算带来误差，故在进行车牌字符识别之前，对车牌的边框进行处理。



字符分割

车牌字符分割

- 首先实现各级设定的阈值和图片直方图找出波峰的功能，然后分别找出水平波峰和垂直波峰，根据得到的波峰进行字符分割。
- 需要注意字符 “L” 、 “U” 等情况。



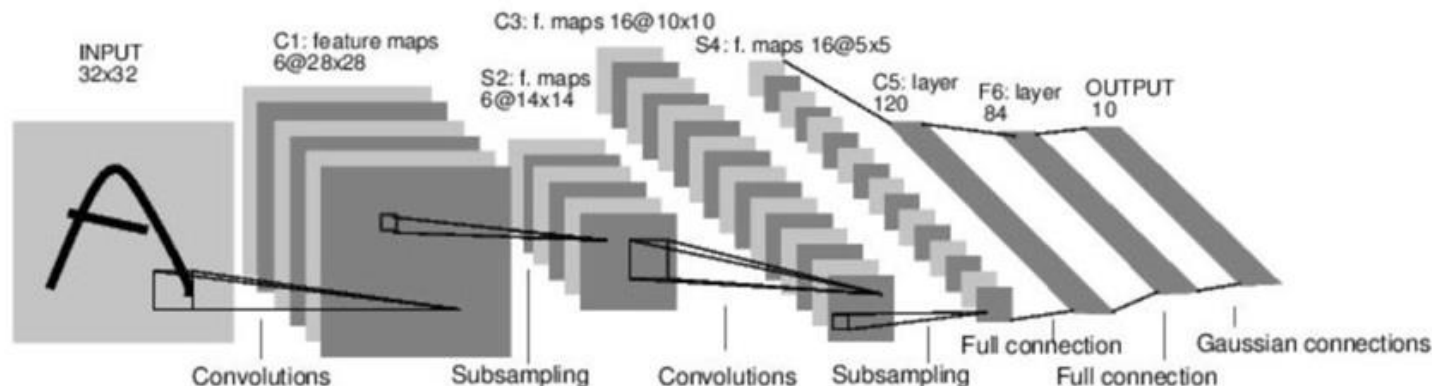
目录

| | |
|---|--------|
| 1 | 案例背景 |
| 2 | 车牌定位 |
| 3 | 字符分割 |
| 4 | 车牌识别模型 |
| 5 | 模型评估 |
| 6 | 拓展思考 |

车牌识别模型

车牌识别建模

- 本案例采用CNN卷积神经网络作为车牌识别算法，这是因为卷积神经网络中用到“权值共享”的概念，使得网络的结构变得非常简单，所以在图像识别领域中应用的十分广泛。
- 一个深度卷积神经网络通常包含输入层、多个卷积层、对应的非线性映射层和池化层。最后通过全连接层将二维的特征图连接成一个向量输入到最后的分类器，得到概率的输出，如下图：

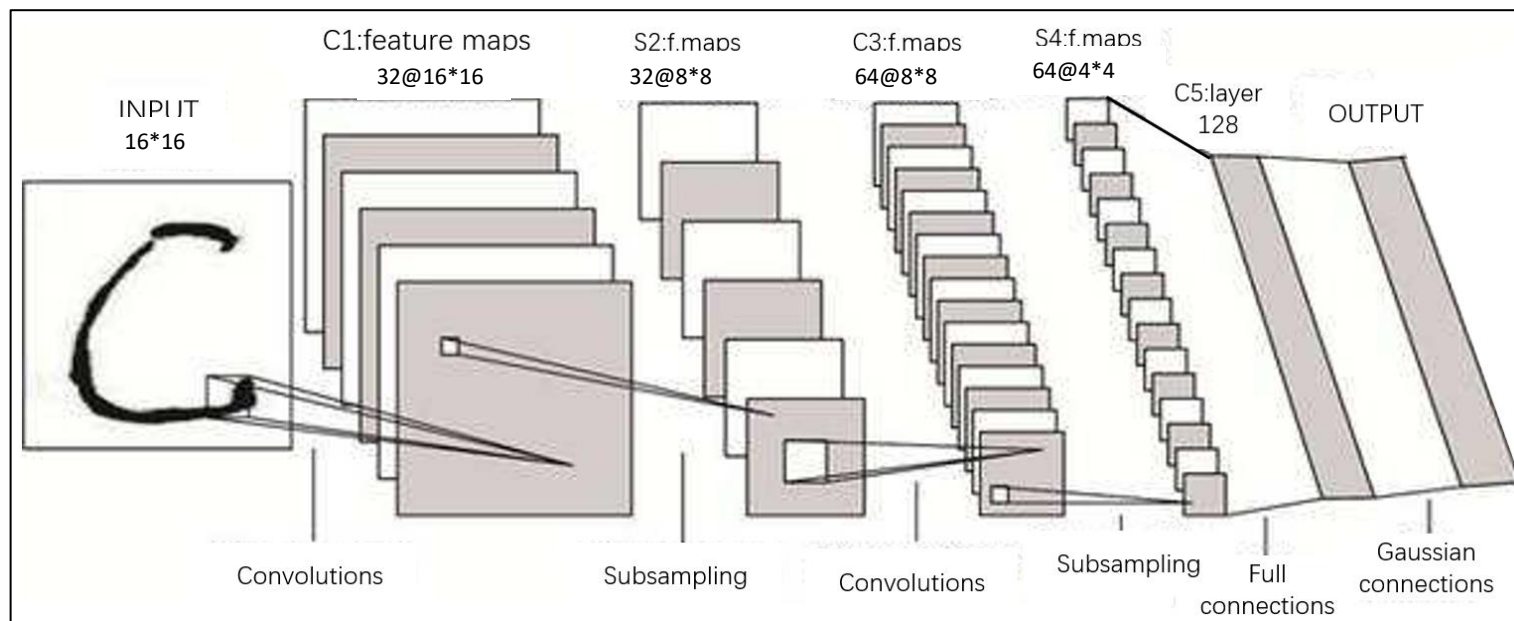


Conv filters were 5x5, applied at stride 1
Subsampling (Pooling) layers were 2x2 applied at stride 2
i.e. architecture is [CONV-POOL-CONV-POOL-CONV-FC]

车牌识别模型

卷积神经网络

- 在车牌识别模型中构建两个分类模型，分别对应车牌中省份缩写的字符分类模型（此时输出神经元个数为34）和车牌中字母和数字的分类模型（此时输出神经元个数为34，中国车牌规定26个字母中的O和I不能使用）
- 卷积过程如右图。



车牌识别模型

在本例中，详细的训练过程如下：

- 1、确定网络的输入变量和输出变量：
- 输入变量：即经过车牌字符裁剪后的字符灰度图片，并且为了统一大小，把图片缩放到宽度为16和高度为28的图片，所以一张图片的维度是 $28*16*1$ ，网络可以接受多个图片，所以网络的输入层维度为： $N*28*16*1$ ，N为图片数量；
- 输出变量：原始的输出变量是对应字符数量，是离散变量，这里将其转换为独热编码。独热编码过程：有多少个不同的分类值就有多少个状态值，每次只有一个状态值被激活，换句话说，独热编码是一个只有一位是1，其他都是0的编码。例如在0~9的数字分类中，6的独热编码为 “0000001000” ；

车牌识别模型

- 2、划分训练集和测试集：

把图像数据划分为训练集和数据集，训练集用以训练卷积神经网络参数，测试集用以验证网络的准确性；

- 3、一共建立 2 层卷积层、2 层池化层和 1 层全连接层, 接下来利用损失函数和优化器来优化模型，提高模型的准确率。

- 4、损失函数：交叉熵

优化器：自适应优化器

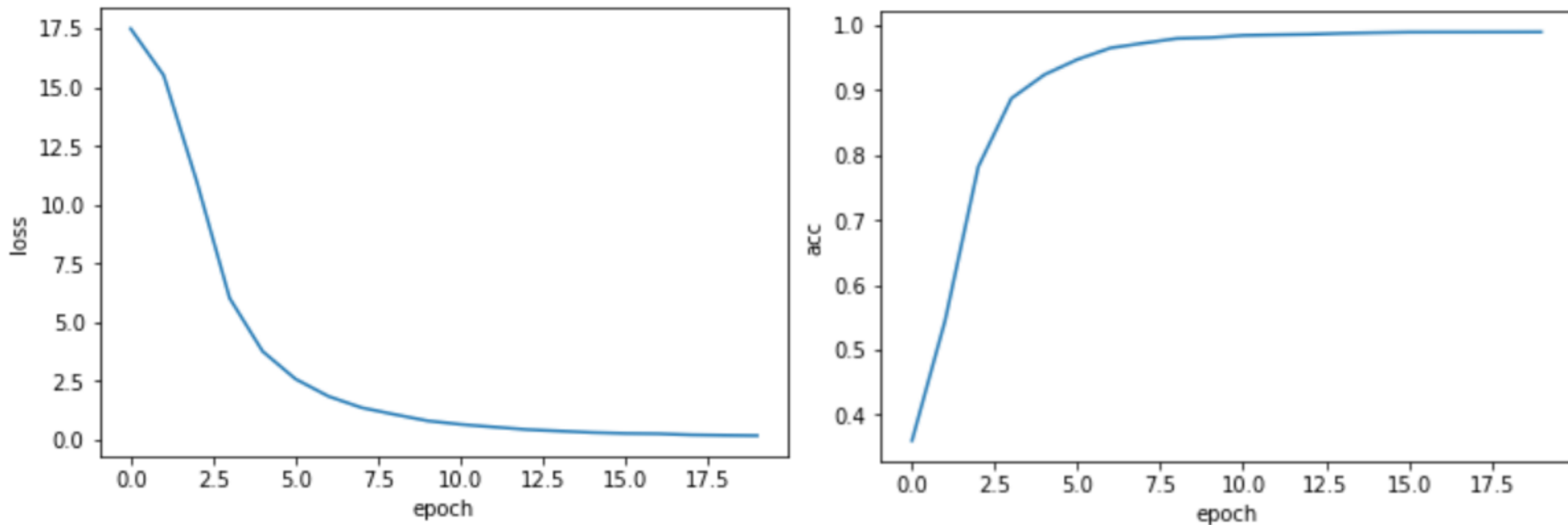
目标：最小化损失函数

目录

| | |
|---|--------|
| 1 | 案例背景 |
| 2 | 车牌定位 |
| 3 | 字符分割 |
| 4 | 车牌识别模型 |
| 5 | 模型评估 |
| 6 | 拓展思考 |

模型评估

根据一开始划分的训练集和测试集，利用测试集来验证模型的准确率，准确率高达 98%



模型评估

根据一开始划分的训练集和测试集，利用测试集来验证模型的准确率，准确率高达 98%

```
Reading checkpoint...  
loading: E:\TD\第十周--十一周  
INFO:tensorflow:Restoring par  
预测结果: 皖A34H50
```



```
Reading checkpoint...  
loading: E:\TD\第十周--十一周 (大  
INFO:tensorflow:Restoring paramet  
预测结果: 皖AU2087
```



```
Reading checkpoint...  
loading: E:\TD\第十周--十一周 (大  
INFO:tensorflow:Restoring param  
预测结果: 皖AY5T83
```



```
Reading checkpoint...  
loading: E:\TD\第十周--十一周  
INFO:tensorflow:Restoring pa  
预测结果: 皖A739V1
```



目录

| | |
|---|--------|
| 1 | 案例背景 |
| 2 | 车牌定位 |
| 3 | 字符分割 |
| 4 | 车牌识别模型 |
| 5 | 模型评估 |
| 6 | 拓展思考 |

拓展思考

- 在修改模型的损失函数、神经网络层数及神经元个数的时候，这些参数都会在一定程度上影响模型的准确率
- 其他具有挑战性的任务场景没有考虑：
 - 光线暗或者比较亮
 - 距离摄像头相当的远或者相当近
 - 在雨天，雪天，或者雾天
 - 由于相机抖动造成的模糊
 - 其他的比较有挑战性的车牌
 - 没有车牌的新车



Thank you!