

复旦大学 计算机学院

2018-2019 学年 第 1 学期 期末课程论文

课程名称: 计算机视觉 课程代码: COMP130124.01

开课院系： 计算机学院 考试形式： 课程论文

姓名： 蒋艳琪 本人签名： _____

学号: 16307130316 专业: 计算机科学与技术

提示：请同学们秉持诚实守信宗旨，谨守考试纪律，摒弃考试作弊。学生如有违反学校考试纪律的行为，学校将按《复旦大学学生纪律处分条例》规定予以严肃处理。

部分	1	2	3	4	5	6	7	8	总分
得分									

车牌检测与识别：

传统视觉算法与端到端深度学习方法之比较

1. 视觉任务介绍

车牌检测与识别技术是智能交通中不可或缺的重要一环，广泛用于高速公路车辆视频监控、停车场智能管理、电子警察系统、道路收费系统等。近年来，汽车的产销量连年攀升，为有关部门对车辆的统一管理带来了不小的挑战；同时，人们对自动化的要求越来越高，故一个识别精度高、处理速度快的车牌识别系统有着巨大的经济价值和实用意义。

实现车牌检测与识别的大致流程是：对拍摄得到的车牌图像进行分析，自动提取车牌信息，确定车牌号码。但是由于诸多因素的干扰，比如周边环境复杂、光照条件恶劣、车牌本身磨损严重等，车牌识别系统至今还没有寻找到一个完备的解决方案。

关于车牌识别系统的研究，国内外学者已经做了大量的工作。传统方法主要依赖于图像处理算法和模式识别技术，通常在某些特定条件下有较好的表现，但是很难在各种开放、复杂环境中保持鲁棒性；如今深度学习方兴未艾，基于深度学习的端到端识别技术在目标检测、文字识别等应用领域均取得了突破性的进展，而将其迁移到车牌识别领域也逐渐成为一个新的研究方向。

2. 方法阐述

2.1 国内车牌的规格和特征

我国机动车辆使用的号牌按照国家标准规定，长度为 44cm，宽度为 14cm，共有 7 个字符和一个分隔符“·”。其中，民用汽车牌照上的第一个字符为汉字，是省、直辖市、自治区的简称；第二个字符是大写英文字母，代表地级行政区的字母代码；后面是一个五位数的汽车牌号，由数字和大写英文字母组成。为了避免混淆，大写字母 O 和 I 弃之不用。^[1]

此外，不同类型的机动车车牌的颜色特征也是不同的，本文主要研究了黄底黑字车牌、蓝底白字车牌和渐变绿色的新能源车车牌。

2.2 非深度学习：传统车牌识别算法

传统的车牌识别方法的主要流程分为以下三个部分^[2]：

第一步是车牌定位。通常，拍摄到的图片含有大量无关的背景内容，这会对车牌识别造成干扰，所以需要先定位车牌。这一步需要较高的定位精度，否则会直接导致车牌识别的失败。

第二步是字符分割与颜色识别。通过适当算法对车牌进行几何校正、缩放等把每个字符切分出来，因为对单个字符分别进行识别比较简单且准确率高。同时这一步需要识别出车牌颜色，因为颜色对于字符识别而言是无关信息，后续会直接去除颜色。

第三步是字符识别，通过模式识别算法，从单个字符图像中提取特征数据，最后给出识别结果。

2.2.1 车牌定位

车牌定位是整个车牌识别系统的基础,准确的定位能为后续的字符分割和识别打下良好的基础;而定位的偏差则会直接导致整个系统的失败。在现实情况中,由于光照条件千差万别、周边背景有大量干扰因素,在整幅图像中精确地定位出车牌区域难度较大。

本文讨论的主要是两种常见的车牌定位算法:基于车牌边缘特征的车牌定位法和基于车牌颜色特征的定位法。

● 基于车牌边缘特征的定位法

该算法的主要思路是突出车牌的字符和边缘信息,与自然场景下的文字检测有相通之处。主要流程如下图:



第一步是对输入的彩色车牌图像进行灰度化处理,合理的灰度化将对图像信息的提取有很大帮助,也是后面边缘检测的基础。本文直接采用了 Opencv 的 `cvtColor` 函数进行处理。

第二步是边缘检测。边缘检测的目的是考察图像的像素变化情况,标识图像中亮度有明显变化的点。由于车牌的字符特征明显,很适合用边缘检测来保留字符边缘特征,剔除不相关信息。本文主要比较了 Sobel 算子、Canny 算子、Laplacian 算子的边缘检测效果。

第三步是对图像进行二值化处理。为了减少前面边缘检测的噪声的影响,采用最大类间方差法(OTSU)^[3]对图像进行阈值化处理,剔除一些梯度值较小的像素,使图像边缘更清晰。该方法按照图像的灰度特性,将图像分为目标部分和背景部分,通过最大化目标和背景的类间方差来获取分割的最佳阈值,取得了良好效果。

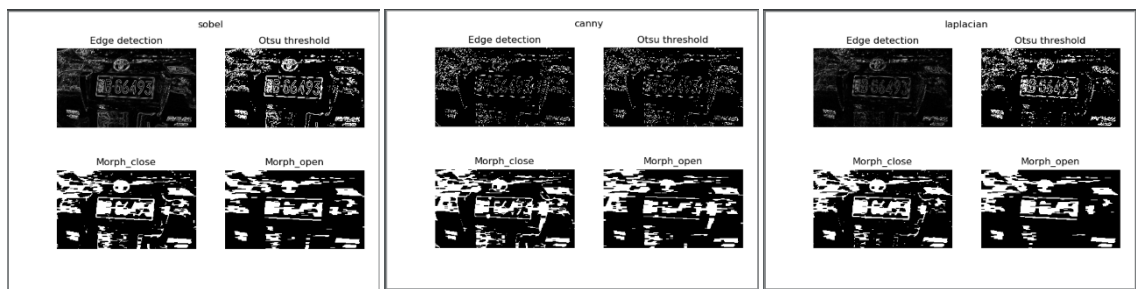
第四步是数学形态学操作。先进行一次闭运算,对图像进行先膨胀后腐蚀,使得车牌的字符区域连接起来,然后进行一次开运算,对图像进行先腐蚀后膨胀,来消除车牌噪声。经过形态学运算,原图像的候选车牌区域变为长方形的连通区域。

最后是去除伪车牌。对得到的白色连通区域的轮廓进行处理得到矩形边界框,去除面积过小的点;再考虑我国车牌的长宽比特征为 22:7,在车牌定位过程中可能会损失部分车牌信息,故取长宽比阈值为 2-5.5,这样就可以去除伪车牌,进一步缩小范围。

原图像:



下图比较了各种边缘检测算子的效果。



上面三张图从左到右分别是 **Sobel/Canny/Laplacian** 算子的效果。每张图有四幅子图，显示了边缘检测/二值化/闭运算和开运算之后的结果。从中可以发现，**Sobel** 算子抗噪声能力较强，但是对边缘定位不是很准确，图像的边缘不止一个像素；**Canny** 算子使用两种不同的阈值分别检测强边缘和弱边缘，容易造成边缘丢失；**Laplacian** 算子容易丢失部分边缘方向信息，造成一些不连续的检测边缘，同时抗噪声能力较差。

由图中对比可知，**Sobel** 算子在二值化后对比强烈，边缘清晰。比较数学形态学运算结果，发现 **Canny** 算子由于边缘较弱，最终导致车牌区域和周围连在一起，无法正确框出矩形轮廓。这一步边缘检测的主要目的不是检测出具体边缘，而是为后面框出车牌区域做准备，所以 **Canny** 算子反而不是很适用。再综合考虑车牌区域区分度、计算速度和抗噪声能力，最终选择 **Sobel** 算子作为边缘检测算子。

但是，基于车牌边缘特征的定位法是存在诸多问题的。比如下图：



不仅车牌部分没有完全框准确，而且框出了车牌以外的其他内容。所以一般还需要依赖于对框出区域的颜色判断等方法，进一步去除伪车牌。

● 基于车牌颜色特征的定位法

该算法的主要思路是利用车牌的颜色特征信息（字符颜色或背景颜色）快速定位到与车牌颜色有关的区域。

主要流程如下图：

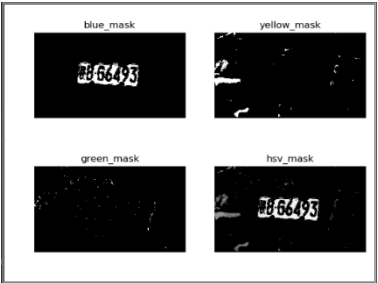


第一步是把输入的 **RGB** 颜色空间的彩色车牌图像转换到 **HSV** 颜色空间。**HSV** 三个分量分别代表色相、饱和度、明度，更符合人的视觉感知结果。尤其是 **HSV** 颜色空间把明度作为

一个单独的分量提取出来，应用在车牌定位中能实现对颜色敏感而对光照不敏感的算法，从而有效克服车牌褪色和光照不理想的恶劣条件。

第二步是对每种颜色设置对应的掩模（mask）以去除背景。这一步主要是利用 Opencv 自带的 `inrange` 函数。对于给定的 HSV 图像，设置一个颜色的范围，就可以把不在这个颜色范围的像素点全部设为 0，在这个颜色范围内的像素点全部设为 255。根据网上查到的蓝色、黄色、绿色的大致范围，结合实际的实验结果，本文取了这三种颜色的经验阈值为：

```
lower_blue = np.array([100, 110, 110])
upper_blue = np.array([130, 255, 255])
lower_yellow = np.array([15, 55, 55])
upper_yellow = np.array([50, 255, 255])
lower_green = np.array([50, 50, 50])
upper_green = np.array([100, 255, 255])
```



上图展示了对图片分别进行三种颜色的掩模操作之后的结果，最后把这三幅图片进行按位与操作，得到输出的 `hsv mask`。

在第三步数学形态学操作之后，本文又增加了一步固定阈值二值化。因为前面得到的是三种颜色 `mask` 的叠加，所以还存在一定的干扰信息。但是可以看到，真正的车牌区域相对其他部分更亮，所以本文通过尝试调整，设置了经验阈值为 110，这样可以最大程度地去除其他颜色的干扰。

最后一步去除伪车牌与基于车牌边缘特征的定位法是类似的。

最终定位的结果如图所示。



2.2.2 字符分割与颜色识别

车牌字符分割是为了在已经定位出的车牌区域内获取单个字符。车牌字符分割的准确与否直接影响到识别效果的好坏。由于车牌污染、拍摄角度倾斜、边框的影响等，字符分割是一个巨大的挑战。

主要流程如下图：



车牌定位完成之后，由于提取出来的车牌区域内图像可能存在倾斜，所以需要倾斜校正。倾斜校正算法主要依据仿射变换，因为它可以保持二维图形的平行性和平直性。步骤是：获

取上面框出的矩形框的四个顶点，根据坐标判断其旋转角度；计算得到其中三个顶点变换后对应的坐标；利用变换前后的对应坐标求得变换矩阵；根据变换矩阵计算变换后图像。倾斜校正的结果如下图所示：



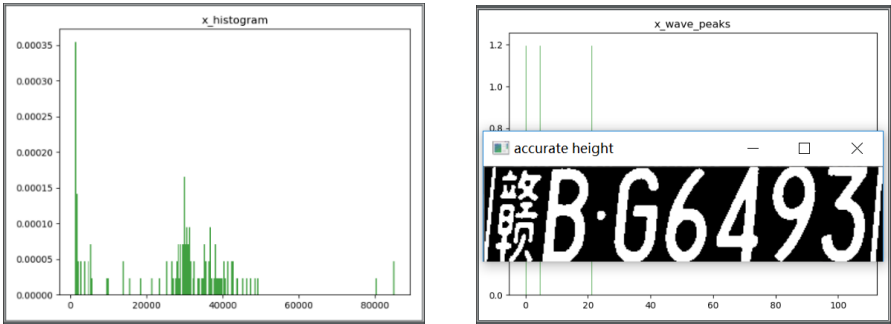
同时，这一步也对车牌的颜色进行了判断。本文简单地采用了统计各个像素点颜色的方法。思路就是扫描图像，对每个像素点判断其颜色，统计出来最多的就是车牌的颜色。此时还可以利用这个颜色的阈值对车牌区域进一步调整范围，去除无用部分。

为了使得车牌的字符更加清晰，统一进行二值化处理，同样采用 OTSU 算法。而黄绿色车牌字符比背景暗，蓝色车牌字符比背景亮，为了统一处理，对黄绿色车牌进行按位取反操作。二值化后得到的灰度图如下：

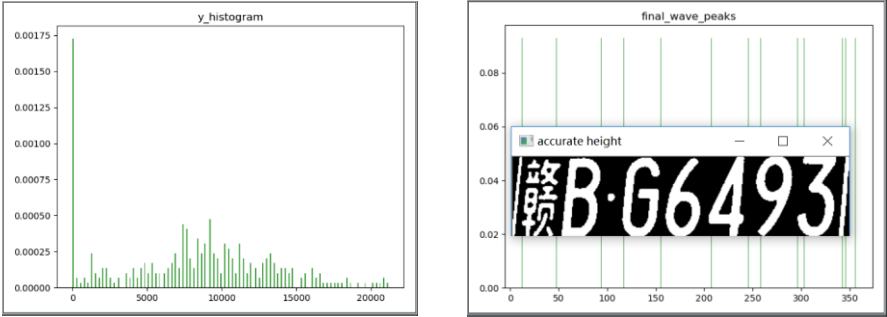


对车牌字符的分割主要是采用了投影分割法，对二值化图片的像素分布直方图进行分析，利用其波峰波谷特征，找出相邻字符的分界点进行分割。投影分割包括水平方向投影和垂直方向投影，分别用于车牌上下边界的进一步校正和各个字符的分割。

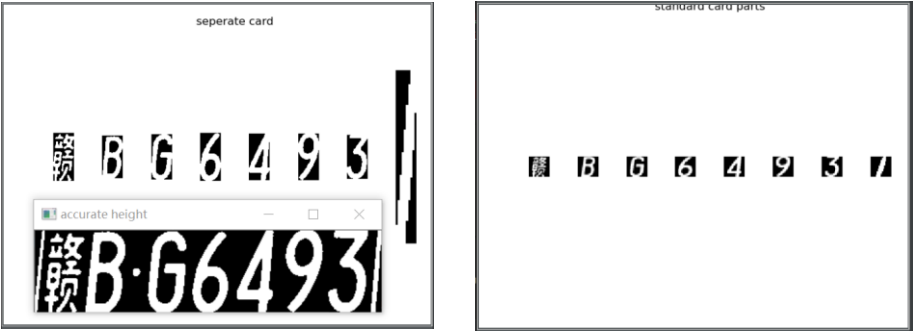
对二值化图像，画出其水平方向投影直方图，即统计每一行白色像素点的个数。然后是对这张直方图进行峰谷分析。设定合理的阈值后，在直方图中区分出上升点和下降点，上升点和下降点之间被认为是一个波峰。可以预测到直方图必定会出现峰谷现象，至少会出现一个波峰（即车牌区域，有大量的白色像素点），也可能出现多个波峰（因为前面定位可能会有一些偏差，如果车牌图像含上下边框，就可能有多余的峰出现）。自然地，水平方向投影宽度最大的波峰对应车牌区域，从而可以依据这个波峰对应的上升点和下降点位置，去除车牌上下多余的部分。水平方向投影直方图及去除上下多余部分后效果如下：



同样地，对二值化图像画出垂直方向投影直方图，即统计每一列的白色像素点个数。因为相邻字符中间是有空隙的，反映在直方图中就是一个一个的波谷。下图（右侧）显示的竖线对应的就是每个峰的上升点和下降点，每个峰对应一个字符。所以按照这些竖线进行切割，就能得到单独的字符图像。垂直方向投影直方图及字符切割线效果如下：



最后是对分割好的图片进行归一化处理。根据前面得到的上升点和下降点分割字符，但此时每张图片大小可能不同。为了便于后续的识别操作，本文将单个字符图像缩放至预先设定好的大小 20*20。



2.2.3 字符识别

车牌的字符识别是车牌识别系统最终能够成功的关键因素。车牌的字符识别与一般的 OCR 文字识别相比难度较低，因为字符总量少，仅仅包括有限的汉字、大写英文字母和数字，而且字体是一致的。但是它又要求识别率要尽量高，识别速度尽量快。

本文主要比较了支持向量机和卷积神经网络在字符识别任务中的表现。由于中文字符和英文数字字符有很大的不同，对汉字和数字字母分别训练了模型。

● SVM

SVM 方法首先是对图片进行预处理，进行抗扭矩操作，再提取方向梯度直方图特征(HOG)^[4]作为描述符。方向梯度直方图特征(HOG)的思路是：把图像分成小的连通区域(cell)，采集每个 cell 中各个像素点的梯度或边缘方向的直方图，把直方图组合起来即可构成特征描述符(descriptor)。设置模型的惩罚系数 $C=1$ ，采用径向基核， $\gamma=0.5$ 。采用的训练及测试样本为 20*20 大小的二值化字符图片。训练与测试情况如右表：

可见 SVM 方法在汉字、数字和字母的识别中都取得了较高的准确率，且一旦训练完成，识别速度非常快。

	数字&字母	汉字
Train set	12563	3087
Test set	612	150
Accuracy	96.4%	85.3%
Time cost	558ms	94ms

● CNN

训练图像大小：width=32，height=40。在训练的时候，采用的 CNN 结构如下：

Conv(kernel size 8*8, stride [1, 1, 1, 1])*16 个 feature map→Relu→

Max_pooling(kernel size 2*2, stride [1, 2, 2, 1])→

Conv(kernel size 5*5, stride [1, 1, 1, 1])*32 个 feature map→Relu→

Max_pooling(kernel size 1*1, stride [1, 1, 1, 1])→

FullConnectedLayer→Dropout(keep_prob 0.5)→

Adam Optimizer, softmax_cross_entropy_with_logits

因为训练样本量较小，增加一层 dropout 在训练时随机让网络的某些节点不工作，防止过拟合。采用 Adam 优化器，最小化 Softmax 交叉熵损失函数。训练与测试情况如下表：

	数字&字母	汉字
Train set	12192	2975
Validation set	371	148
Accuracy(validation)	98.6%	91.2%
Test set	612	150
Accuracy(test)	97.1%	97.3%
Time cost	840.2s	265.2s

对数字&字母训练时迭代了 100 次，对汉字训练时迭代了 300 次，可见在验证集和测试集上都取得了很高的准确率，但是耗时与 SVM 方法相比太长了。这可能与电脑配置有关，侧面反映出卷积神经网络运算量大的弊端。

2.3 深度学习：端到端车牌识别算法

传统的车牌识别算法通常把车牌的检测和识别分成两个子任务，分别通过数字图像处理算法和模式识别算法来解决。然而，这两个任务具有高度的相关性。实践中常常会发现，如果车牌检测得到的位置准确，切割出来的车牌图像清晰标准，车牌识别的准确率就会很高；但是一旦检测和定位过程出现偏差，其错误就会逐步累积，最终导致最后的车牌字符识别失败。而且，整个检测和识别过程依赖大量人工调整的参数，而人工调参最多只能达到局部最优，无法达到全局最优。

反观深度学习模型，不需要人工精心提取特征，全部参数由深度神经网络自主学习得到。目前，基于深度学习的端到端识别技术在目标检测、文字识别等应用领域均取得了突破性的进展，而将其迁移到车牌识别领域也逐渐成为一个新的研究方向。

据此，本文参考了开源的中文车牌识别框架 Hyperlpr^[5]，模仿实现了一个基于深度学习的端到端车牌识别的方法，同时完成车牌的检测和无分割识别任务。

2.3.1 生成模拟车牌数据

深度学习方法依赖于大量训练样本，且样本需要涵盖所有的省份和地区。但由于车牌数据有一定的隐私性，很难找到现成的数据集。所以首先要模拟生成具有真实感的车牌图像。利用网上开源的车牌生成器^[6]，先按照规则随机生成 7 位长的车牌字符串，再在空白的车牌

底板上插入对应字符，添加车牌背景，就可以得到一张标准车牌。为了进一步增强训练样本的差异性，必须模拟自然环境下车牌图像拍摄的种种影响因素，包括透视畸变、仿射畸变、不同光照强度的噪声、高斯模糊处理、高斯噪声、污迹等等；再加入适当的车辆背景图片，使得最终生成的图片更接近自然环境下的真实效果。把对应的车牌字符串存为 `label`，即可得到训练所用的大量车牌数据。生成的部分车牌图像如下：



对应标签文件如下：

```
00000.jpg:鄂XVS6EP
00001.jpg:陕Z7BA05
00002.jpg:苏WH5LH4
00003.jpg:晋SHQ61K
00004.jpg:藏CQ5K6U
00005.jpg:新HCS12B
00006.jpg:宁PD3XKQ
00007.jpg:闽KX7CB4
00008.jpg:闽E7ZAQV
00009.jpg:鄂LSY6C6
00010.jpg:甘P7H8RW
00011.jpg:黑DF1HRA
00012.jpg:黑EVZ1QH
00013.jpg:京L8HTML
00014.jpg:晋D1PKMK
```

2.3.2 网络结构

● 车牌目标检测模型

对于车牌位置的检测，采用了 `Opencv` 的级联分类器 `CascadeClassifier`，其中封装的目标检测机制是滑动窗口机制+基于 `Harr` 特征的级联分类器的方式。主要步骤是：使用 `Harr-like` 矩形特征做检测，使用积分图对 `Harr-like` 特征求值进行加速，使用 `Gentle Adaboost` 算法训练区分车牌和非车牌的弱分类器，使用筛选式级联把分类器级联到一起，提高准确率。使用的训练样本 `width=51`，`height=13`，采用 20 级分类器，大约 4700 张正样本车牌和 12000 张负样本进行了分类器训练。检测框架为一个多尺度缩放+滑动窗口遍历搜索的框架。对检测得到的矩形框按照经验值裁剪掉上下各占比 0.1 的部分，再进行一定比例的缩放调整，将宽度左右各扩大 0.14 倍，高度上下各扩大 0.15 倍，即可粗定位车牌区域，最大程度保留车牌图像。

● 车牌边界拟合模型

对于粗定位得到的车牌区域，还需要进一步拟合其上下左右边界以减少多余信息的干扰。使用 `CNN` 回归车牌左右边界，训练图像大小：`width=66`，`height=16`。采用的 `CNN` 结构如下：

```
Conv(kernel size 3*3, stride [1, 1, 1, 1])*10 个 feature map→Relu→
Max_pooling(kernel size 2*2)→
Conv(kernel size 3*3, stride [1, 1, 1, 1])*16 个 feature map→Relu→
Conv(kernel size 3*3, stride [1, 1, 1, 1])*32 个 feature map→Relu→
Flatten→FullConnectedLayer→Relu
```

对车牌的上下边界按照经验值，对粗定位得到的车牌区域进行上下边缘拟合。

● 字符识别序列模型

字符识别采用了基于 GRU^[7]的序列模型，没有区分汉字、英文字母和数字，直接实现多标签分类。网络模型由三层卷积神经网络 CNN 和两层双向内控循环单元 GRU 组成。首先通过 CNN 识别特征，然后经过一个全连接层降维，再按照水平顺序输入到 GRU 循环神经网络。输入 tensor 大小为 164*48*3 通道，网络结构如下：

```
Conv(kernel size 3*3, stride [1, 1, 1, 1])*32 个 feature map→BatchNormalization→Relu→  
Max_pooling(kernel size 2*2)→  
Conv(kernel size 3*3, stride [1, 1, 1, 1])*64 个 feature map→BatchNormalization→Relu→  
Max_pooling(kernel size 2*2)→  
Conv(kernel size 3*3, stride [1, 1, 1, 1])*128 个 feature map→BatchNormalization→Relu→  
Max_pooling(kernel size 2*2)→  
FullConnectedLayer(units=32)→BatchNormalization→Relu→  
Add(Gru1(units=256), Gru1b(units=256, go_backwards=True))→  
Concatenate(Gru2(units=256), Gru2b(units=256, go_backwards=True))→  
Dropout(drop rate 0.25)→FullConnectedLayer(units=n_class, activation = softmax)
```

其中 GRU 层返回整个序列，权重初始化采用 He 正态分布。

第一层 GRU，一路正向输入序列，一路反向输入序列，然后将其输出结果相加；第二层 GRU，一路正向输入序列，一路反向输入序列，然后将其输出结果直接相连。最后经过一个全连接，输出每个字符的概率。取概率最大者作为预测值。

2.3.3 最终实现

整个算法使用 Keras 框架和 Opencv 自带级联分类器实现，对车牌目标检测、车牌边界回归和字符识别分别构建模型。其中，字符识别问题看作是一个序列标注问题，采用 CNN+GRU 来解决。又因为利用目标检测可以同时检测到一张图片中的多个车牌并对其进行识别，满足一定置信度即可认为是有效车牌。识别多张车牌效果如图：



3. 实验和讨论

3.1 实验设置及评价指标

本实验一共选取了 240 张包含车牌的图片，车牌颜色包含蓝牌、黄牌、绿牌，拍摄角度包含正视图、侧视图，光照条件包含白天、黑夜，图像分辨率也有差异。测试集模拟了小区门口高清红外线摄像头所得图像、普通手机拍摄图像和高速公路视频监控图像等多种情况。

结合现实生活中对车牌识别系统的一般要求，即识别准确率高、识别速度快，本文对算法的评价指标主要设置为车牌区域定位准确率、字符识别准确率、处理时间。其中，正确识别出对应位置的三个字符以上被认为是定位准确；在定位准确的图片中，比较字符识别准确率，分别考察全部字符识别正确的比例和只有一个字符识别错误的比例。

3.2 实验结果

● 非深度学习：传统车牌识别算法

最终方案采用基于车牌颜色特征的定位算法，若颜色定位失败则回退至基于车牌边缘特征的定位算法。比较 SVM 和 CNN 的识别效果如下表：

	定位正确率	字符识别准确率 (0 err + 1 err)	耗时	比较
SVM	61.7%	41.9%+41.2%	101s 0.42s/pic	✓
CNN	60.8%	43.0%+37.3%	1065s 4.4s/pic	切割出来的字符 不如训练样本图 像清晰规范

可以发现，传统车牌识别的定位准确率不高；在定位成功的图片中，SVM 方法和 CNN 方法的字符识别准确率相当，但 CNN 非常耗时。经过分析，主要原因是切割出来的字符不如训练样本清晰规范，导致 CNN 过拟合，泛化能力差。

● 深度学习：端到端车牌识别算法

	定位正确率	字符识别准确率 (0 err + 1 err)	耗时
CNN+GRU	87.9%	88.2%+7.6%	1126s 4.7s/pic

基于 CNN+GRU 的端到端识别方法，相比传统车牌识别算法，定位正确率和字符识别正确率都有了大幅度的提升，其关键在于提高了车牌目标检测的准确率，为字符识别打下了良好的基础。可见在开放、复杂的环境中，深度学习方法由于数据的多样性可以提取到车牌更为本质的特征，且抗干扰能力更强，而这些是传统的视觉算法所无法比拟的。但是较长的处理时间仍然是一个不容忽视的问题。

3.3 效果和优缺点

车牌识别系统是模式识别和数字图像处理高度融合的一项技术。由于车牌背景的复杂性，车牌特征的多样性，模糊车牌、掉色车牌和其他人为因素的干扰，车牌识别系统有较大难度，至今没有一个完全通用的方案。在传统方案陷入瓶颈之时，以大量数据为基础的深度学习方法或许能为车牌识别任务提供新的思路，并进一步简化处理过程，增强鲁棒性。

● 非深度学习：传统车牌识别算法

优点主要有：

1. 识别速度总体较快，能实现实时检测
2. 定位过程与识别过程分离，对算法的各部分分别进行优化
3. 充分利用了车牌的形状、颜色等种种特征

缺点主要有：

1. 严重受到光照条件、周围颜色干扰、车牌形变影响，车牌定位的准确性不是很高
2. 定位和识别分离，定位不准确直接导致识别错误和失败，错误具有累积放大效应
3. 无法识别双行车牌、一张图片中的多张车牌、视频中的车牌，应用场景不够广泛

● 深度学习：端到端车牌识别算法

优点主要有：

1. 实现了车牌字符的无分割识别，减少由于分割带来的信息损失

2. 定位准确率和识别率高，各种测试环境下均有良好表现
3. 轻量，模型预先训练，检测识别代码量很小

缺点主要有：

1. 车牌检测和车牌识别没有共享卷积特征，运算量大
2. 需要大量标注数据集，成本高、难度大
3. 神经网络复杂，收敛较慢，检测耗时

4. 总结与展望

本文主要针对车牌检测与识别任务，实现了传统车牌检测识别算法与基于深度学习的端到端车牌检测识别算法，做了一组实验以评价其定位准确率、识别准确率和处理时间，并根据算法本身和实验结果对这两种算法进行了优缺点分析。

但是由于时间不足和个人能力欠缺，主要还有以下不足之处：

1. 能够识别的车牌种类局限于蓝底白色车牌、黄底黑色车牌和新能源车牌，对一些特殊的车牌，如武警、公安、使领馆车牌，或者多行车牌无法识别。未来应当进一步考虑这些特殊车牌的特点，设计新的算法以兼容更多种类的车牌。
2. 对于运动中较为模糊的车牌、光照条件不好或者褪色的车牌识别效果相对较差，在实际应用中，特别是高速公路车辆视频监控的应用场景下表现欠佳。这个问题仅凭传统的视觉算法很难解决，只能寄希望于增加神经网络的训练样本，进一步优化网络结构，使其具有更强的泛化性能。
3. 深度学习所用神经网络依赖于充足的、均衡的训练样本，而大量真实的训练样本很难获得，标注也会消耗巨大的人力物力。可以引入弱监督学习思想以扩充训练样本数量，降低标注难度。
4. 神经网络运算量巨大，耗时较长，且受到硬件条件的限制，可以考虑将其部署至 Hadoop 等分布式云计算平台，进一步提高车牌识别的效率和性能。

参考文献

- [1] GA36-2007 《中华人民共和国机动车号牌》
- [2] <https://github.com/liuruoze/EasyPR>
- [3] Otsu N . A Threshold Selection Method from Gray-Level Histograms[J]. IEEE Transactions on Systems, Man, and Cybernetics, 2007, 9(1):62-66.
- [4] Dalal, Navneet, Triggs, et al. Histograms of Oriented Gradients for Human Detection[C]// IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, 2005:886-893.
- [5] <https://github.com/zeusees/HyperLPR>
- [6] <https://github.com/huxiaoman7/mxnet-cnn-plate-recognition>
- [7] Cho K, Van Merriënboer B, Gulcehre C, et al. Learning Phrase Representations using RNN Encoder--Decoder for Statistical Machine Translation[J]. empirical methods in natural language processing, 2014: 1724-1734.