上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

计算机视觉课程报告

题目：基于多层次视觉与任务的车牌号识别算法

评分：_____

学生姓名：朱楷文

学生学号：520030910178

专　　业：计算机科学与技术(IEEE 试点班)

学院(系)：电子信息与电气工程学院

目 录

第一章 实验要求分析

给定含有中国大陆中小型汽车（含新能源汽车）车牌号的图像，本实验要求设计算法识别出其中的车牌号。按识别难度划分，给出的图像有三种类型：easy, medium 和 difficult，其中 easy 类型的图像为只包含车牌的正视图，medium 类型的图像为车头或车尾的正视图，difficult 类型的图像为车头或车尾的斜视图（包含背景），并具有一定的透视投影效果。

自然地，我们可以将该任务分成多个层次的子任务：

(1) 车牌定位：找到图像中车牌的位置，将其分割出来；
(2) 视角矫正：将车牌的图像矫正为正视图；
(3) 字符分割：将车牌号中的各个字符分割开来；
(4) 字符识别：分别识别每个分割出的字符。

对于 easy 类型的图像，完成第 (3)、(4) 个子任务即可；对于 medium 类型的图像，需要完成第 (1)、(3)、(4) 个子任务；对于 difficult 类型的图像，四个子任务均需完成。

第二章 相关工作

本实验的主要难点在于目标检测和分类两个经典任务，即定位车牌和将字符识别（分类）为特定的汉字、字母或数字。这两个任务一直是计算机视觉领域研究的重点，尤其是深度学习兴起以来，发展出了许多卓有成效的算法。

2.1 目标检测

目标检测的发展历史可以划分为两个阶段：传统算法阶段 (1998-2014) 和深度学习算法阶段 (2014-今) [1]。在传统算法阶段，基于阈值的二值分割的算法得到广泛应用，常常需要手动设计特征并进行调参。在深度学习算法阶段，发展出了两条路线：anchor-based 方法和 anchor-free 方法，其中 anchor-based 方法又可以分为 one-stage 和 two-stage 两种。Anchor-free 方法的著名模型有 CornerNet (2018)，CenterNet (2019) 等，one-stage 方法的著名模型有 YOLO (2016)，SSD (2016) 等，two-stage 方法的著名模型有 RCNN (2014), FPN (2017) 等[1]。最近，Meta AI 发布了 Segment Anything (2023) 模型，证明通用大模型在目标检测、图像分割任务中能够达到优异的性能[2]。

2.2 图像分类

图像分类任务中，常见的流程一般是先编码图像特征，然后通过分类器进行分类。特征提取的经典方法有 SIFT (Scale-Invariant Feature Transform), HOG (Histogram of Oriented Gradient), LBP (Local Bianray Pattern) 等。传统的方法中，表现较好的分类器有支持向量机、随机森林、K-Nearest Neighbors 等算法。深度学习兴起以来，AlexNet (2012), VGG (2014), GoogLeNet (2014), ResNet (2015) 等模型取得了很好的表现，分类精度甚至可以超越人类。

第三章 算法原理

受限于数据和计算资源，本实验采用的算法不使用深度学习，而是充分利用人类认知中的图像特征，包括颜色、形状和纹理等等。

3.1 方法论

本算法的主要指导思想是将视觉和任务层次化，这是符合人类的认知逻辑的，有较好的可解释性，并且有利于调试和改进。这正是可行人工智能的一大目标。

3.1.1 多层次视觉

David Marr 将视觉信息处理分为三个层次：底层视觉、中层视觉和高层视觉[3]。大致而言，底层视觉指对作为原始信号（像素的排列）的输入图像进行改造，转换为人类希望看到的自然图像，例如去噪、增强等；中层视觉指将图像转换为中等抽象的图像，例如分割、拟合等；高层视觉指将图像转换为高度抽象的语义，理解其内容，例如识别等。对于车牌号识别任务，视觉的层次化可以得到很好的体现：首先根据具体任务改造图像，如去噪、平滑、二值化，此为底层视觉；然后分割出车牌，进而分割出车牌上的字符，此为中层视觉；最后识别字符，此为高层视觉。

3.1.2 多层次任务

近年来随着深度学习的兴起，端到端的算法越来越流行，即，输入原始数据后，通过算法处理后直接输出任务需要的结果，从外部来看没有明显的模块化。但更符合人类认知逻辑的流程是，将整个任务分解为多层次的子任务，各个子任务相对独立、环环相扣。如第一章所述，本算法采用后一种流程，将车牌号识别分解为车牌定位、视角矫正、字符分割、字符识别四个子任务。同时，对每个子任务，也会从多个层次的视觉出发进行思考。

3.2 各子任务算法原理

3.2.1 车牌定位和字符分割

为了分割出车牌和其上的字符，本算法主要利用了目标的颜色与形状特征：

(1) 颜色：考虑到车牌为蓝底白字或绿-白渐变底黑字，可以直接将相应的颜色提取出来，将其它的颜色过滤掉，这样可以极大地排除干扰。进行颜色提取，还需将原来的 RGB 颜色转化为 HSV 颜色。这是因为，相较于适应于人类识别的 RGB 颜色，HSV 颜色编码了色调、饱和度、明度，更适合计算机进行处理[4]。

(2) 形状：形态学操作是一种针对几何结构的处理，适合处理二值图。本算法利用了两种基本的形态学操作：腐蚀和膨胀。这两种操作使用一个结构元（通常为小正方形），在其上选择一个锚点（通常为几何中心），令锚点遍历图像中的像素，如果结构元与图像重叠的像素为白（腐蚀）/ 黑（膨胀），则令锚点处的像素变为黑（腐蚀）/ 白（膨胀）[5]。腐蚀/膨胀操作的效果是令图像的黑色/白色部分扩张，可以用来去噪、平滑等。结合目标特征，就可以用这样的操作对图像施加理想的变换以便后续处理。

通过以上底层视觉的操作后，就可以得到增强目标特征而抑制非目标特征的二值化图像，然后从中层视觉的观点考虑就可以很容易地分割出目标。

具体地，对于车牌定位，首先通过中值滤波平滑图像，然后提取出蓝色、绿色、白色

并膨胀，车牌就会变为标准的四边形，其它物体则很难获得这样的形状，此时对每个轮廓拟合多边形，并找到最大的四边形即可。值得一提的是，完成底层视觉的操作后，为了拟合出四边形以分割出车牌，我尝试了 Canny 边缘检测、Hough 变换直线检测和 Harris 角点检测，但效果均不理想。这可能是因为图像中存在较多的非目标物体的轮廓，造成了较多的假阳错误。而直接对每个轮廓拟合多边形并筛选出四边形可以得到正确的结果。可见，关于形状的先验知识可以极大地帮助中层视觉的处理。

对于字符分割，首先仍使用中值滤波平滑图像，然后通过 OTSU 自适应阈值算法将图像二值化（此时若图像白色多于黑色，说明是绿-白渐变底黑字的车牌，需要进一步处理，可以将原图像的黑色部分提取出来；之所以不对蓝底白字车牌进行颜色提取的操作，是因为白色的字符容易脏污，不易提取）。二值化后，通过腐蚀去除噪点，再通过膨胀使每个字符连通为整体，计算出每个轮廓的外接矩形框，筛选出具有合适大小和宽高比的矩形框作为字符区域即可。最后，还可以通过中值滤波进一步去噪、平滑。此外，对于第一个字符以外的字符，考虑到它们均为字母或数字，是连通的整体，可以利用这一点进一步去噪：在图像中拟合外围轮廓，如果轮廓数量大于一，只保留所围面积最大的轮廓。

3.2.2 视角矫正

在小孔成像系统中，相机通过线性变换将空间中的 3D 物点转换为成像平面上的 2D 像点。形式地，将一个透视投影矩阵作用在物点的齐次坐标上，就可以得到对应的像点的齐次坐标。投影矩阵取决于相机内参和姿态（即成像平面的位置）[6]。对于同一相机拍摄下的同一物体，视角（相机姿态）不同，投影矩阵就会不同，进而得到的图像就会不同。

在视角矫正任务中，我们希望对于同样的车牌，将视角转变为正面，即，使得成像平面平行于车牌平面。变换前后的差别在于，对于同样的物点，施加了不同的投影矩阵。因此该变换是一个线性变换，只需要施加一个矩阵即可。要计算出这个矩阵，需要有四组变换前后的对应点，而车牌的四个顶点恰好可以满足这个要求，这正是可以由此前的车牌定位任务得到的。

3.2.3 字符识别

(1) 特征编码：本实验采用的特征编码方法为 HOG (Histogram of oriented gradients) 描述子。HOG 算法描述了图像各个局部的特征，其基本思路是，将图像划分为多个子块，对各个子块的梯度幅值和方向进行投票统计，形成基于梯度特性的直方图作为特征，然后将各子块的局部特征拼接起来作为总特征[7]。由于是在各个子块中统计梯度，该算法提供了一定的位置不变性，并且可以很好地刻画边缘和角点的特征。

(2) 分类算法：本实验采用的分类算法为 KNN (K-Nearest Neighbors) 算法。假设已有标注好的样本（高维向量），预测一个新的样本的标签时，考察其最近的 K 个样本的标签，将它们的众数作为新样本标签的预测值[8]。对于样本间距离的度量，本实验采用的是余弦相似度。

第四章 整体流程

本算法的整体流程如下，大多步骤可以直接调用 OpenCV 的函数完成。

4.1 车牌定位

(1) 对图像进行中值滤波以平滑图像。
(2) 提取出图像中的蓝色、绿色、白色部分，将其转换为白色，其余部分转换为黑色。
(3) 对图像进行膨胀操作使得各对象更加平滑、连通。
(4) 找出图像中的所有外围轮廓。
(5) 对所有轮廓拟合多边形，筛选出最大的四边形。
(6) 用最小的四边平行于图像边框的矩形将筛选出的四边形框住，作为车牌区域返回，同时返回该四边形的四个顶点。

4.2 视角矫正

(1) 将源点设为车牌定位任务返回的四边形的四个顶点，对应的目标点设为图像边框的四个顶点，由此计算相应的透视变换矩阵。
(2) 将计算出的变换矩阵作用在车牌图像上，返回得到的矫正后的图像。

4.3 字符分割

(1) 对图像进行中值滤波以平滑图像。
(2) 通过 OTSU 自适应阈值算法将图像二值化，若白色多于黑色，则将原图像的黑色部分变为白色，其余部分变为黑色。
(3) 对图像进行腐蚀以去除噪点。
(4) 对图像进行膨胀以使各对象连通。
(5) 找出图像中的所有外围轮廓。
(6) 计算出每个轮廓的外接矩形框，筛选出具有合适大小和宽高比的矩形框作为字符区域。
(7) 对每个字符图像，进行中值滤波以去噪、平滑。
(8) 对于左起第一个字符以外的字符，拟合外围轮廓，若轮廓数量大于一，则去除轮廓所围面积不是最大的对象。
(9) 对每个字符图像去除四边的黑色边界，将结果返回。

4.4 字符识别

(1) 加载有标签的训练数据（标签为：中国大陆 31 个省级行政区的汉字简称，除 I 和 O 的 24 个大写英文字母，阿拉伯数字 0-9；每个标签含有 7 张图像），对每个样本计算其 HOG 描述子（计算 HOG 描述子前，首先将图像大小调整至一固定值，然后在其四周添加固定大小的黑色边界，这是因为图像中的字符紧贴边缘，填充黑色边界可以更充分地利用字符边缘的信息，下面计算待预测字符的描述子时同样需要这一操作）。
(2) 对于一个需要预测标签的字符图像，计算其 HOG 描述子。
(3) 通过 KNN 算法进行预测，将结果返回。

第五章 实验过程和结果

5.1 实验测试

下面以测试数据中的 3-3.jpg 图像为例，参照前一节叙述的流程展示本算法识别车牌号的过程。

5.1.1 车牌定位

(1) 图 1 展示了原图像和中值滤波后的图像，可见中值滤波操作有效地平滑了图像，使车牌作为蓝色平行四边形的特征更鲜明。

(a) 原图像 **(b) 中值滤波后的图像**

图 1 平滑前后的图像

(2) 图 2 展示了提取蓝色、绿色、白色并二值化后的图像，可见大部分无关元素被排除了。

图 2 提取部分颜色并二值化后的图像

(3) 图 3 展示了膨胀操作后的图像，可见各个对象变得更加平滑、连通，尤其是车牌的平行四边形形状更为标准，易于识别。

图 3 膨胀操作后的图像

(4) 图 4 展示了原图像上画出的识别到的轮廓。

图 4 识别到的轮廓

(5) 图 5 展示了对找到的轮廓拟合多边形的结果，可见车牌对应最大的四边形。

图 5 对轮廓拟合多边形的结果

(6) 图 6 展示了分割出的车牌区域及车牌边缘，这是车牌定位的结果。

图 6 车牌定位的结果

5.1.2 视角矫正

图 7 展示了将车牌图像矫正为正视图的结果。

图 7 将车牌图像矫正为正视图的结果

5.1.3 字符分割

(1) 图 8 展示了二值化后的车牌图像。

图 8 二值化后的车牌图像

(2) 图 9 展示了腐蚀、膨胀后的图像，可见腐蚀操作去除了部分噪点，膨胀使得各个字符连通为整体。

(a) 腐蚀后的图像

(b) 膨胀后的图像

图 9 形态学操作后的图像

(3) 图 10 展示了图像中各轮廓的外接矩形框，可见各字符均被完全框住，但还有一些冗余元素需要去除。

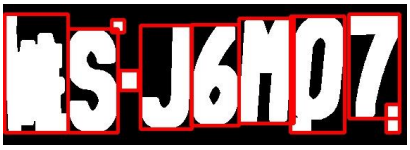


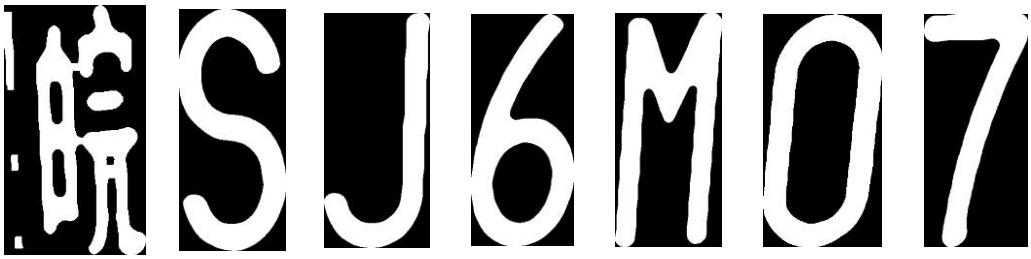
图 10 各轮廓的外接矩形框

(4) 图 11 展示了筛选出的各字符图像。

(a) (b) (c) (d) (e) (f) (g)

图 11 筛选出的各字符

(5) 图 12 展示了进一步处理后的各字符（包括中值滤波，从左起第二个字符开始去除面积较小元素，去除黑色边界）。与图 11 对比可见，处理后的字符更平滑，且部分冗余元素被去除，如两图中的子图 (f)、(g) 所示。这是字符分割的结果。



(a) (b) (c) (d) (e) (f) (g)

图 12 字符分割的结果

5.1.4 字符识别

计算出标注好的训练数据与待识别字符的 HOG 描述子，使用 KNN 算法进行分类，得到识别结果“皖 SJ6M07”，正确。图 13 展示了通过 t-SNE 算法将训练数据和待识别字符描述子降至 2 维后的结果，其中彩色标记对应训练数据，黑色标记对应待识别字符，其旁边的文字为真实值（同时也是识别结果），可见相同的字符呈现出明显的聚集现象，这说明 HOG 描述子确实可以有效捕捉到图像特征以供 KNN 算法正确分类。

5.2 结果与分析

5.2.1 实验结果

在提供的 9 张图像（车牌号真实值如表 1 所示）上进行测试，识别结果如表 2 所示，可见只有图像 3-1.jpg 中的“沪”被错误识别成了“浙”，其余识别结果均正确。

表1 测试数据车牌号真实值

难度	图像 1	图像 2	图像 3
easy	沪 E·WM957	沪 A·F02976	鲁 N·BK268
medium	沪 E·WM957	豫 B·20E68	沪 A·93S20
difficult	沪 E·WM957	沪 A·DE6598	皖 S·J6M07

表2 识别结果

难度	图像 1	图像 2	图像 3
easy	沪 E·WM957	沪 A·F02976	鲁 N·BK268
medium	沪 E·WM957	豫 B·20E68	沪 A·93S20
difficult	浙 E·WM957	沪 A·DE6598	皖 S·J6M07

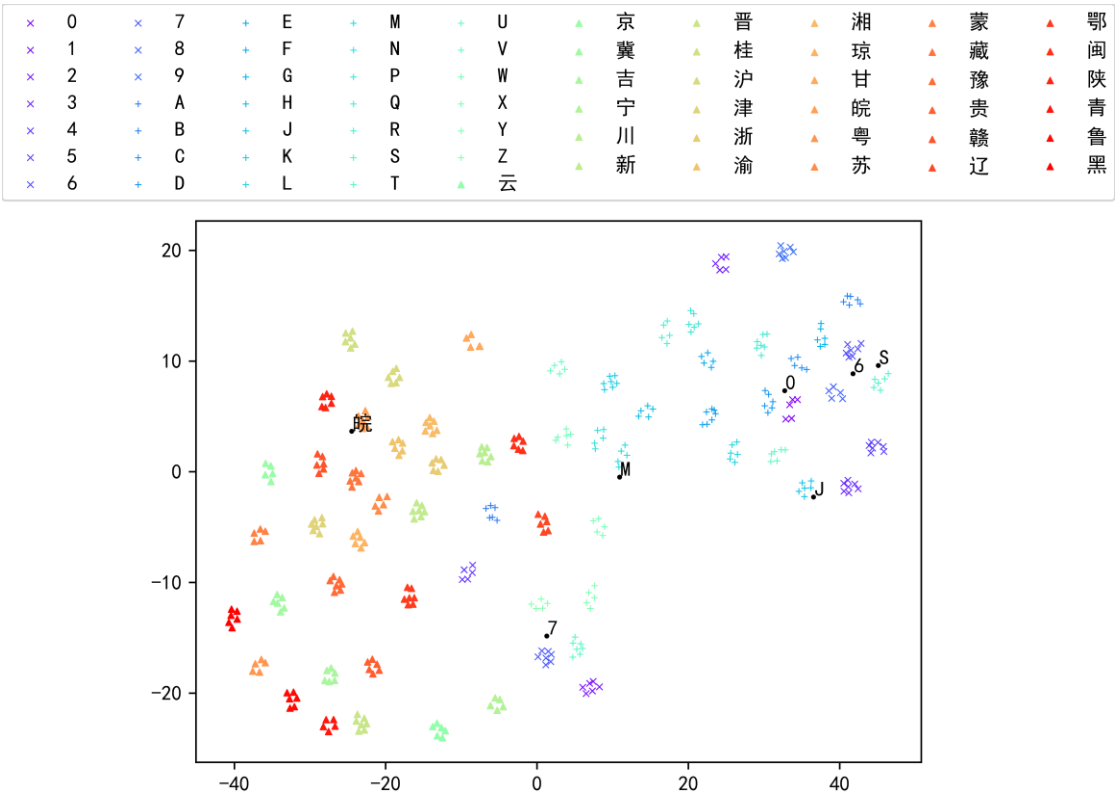


图 13 经 t-SNE 算法降维后的训练数据及测试数据的 HOG 描述子

5.2.2 评价指标

从应用的角度来看，对于算法最直接的评价指标是识别车牌号的准确率，即正确识别的车牌号数量与车牌号总数之比。在提供的 9 张图像上，本算法的准确率为 8/9 ≈ 88.9%。

然而，算法是在字符级别上进行识别的，因此，车牌号的评价粒度过大，在字符级别的评价能更好地衡量算法性能。考虑到算法未必能准确分割出所有的字符，其识别出的字符数量不一定正确，因此不能直接计算识别字符的准确率，只能比较两个序列（车牌号真实值与识别结果）。

为了衡量两个序列的相似度，我们将序列转换为允许包含重复元素的集合，然后考察集合的相似度。一个常用的集合相似度指标是 Jaccard 相似度，它是交集的势与并集的势之比[9]。而对于允许包含重复元素的集合，我们将其表示为键值对的集合，其中键为原集合的元素，值为该键在原集合中出现的次数，各个键值对的键互不相同。两个这样的集合的交集定义为，其键是两个集合的键的交集，对应的值是该键在两个集合中的值的最小值；并集定义为，其键是两个集合的键的并集，对应的值是该键在两个集合中的值的最大值（如果集合不包含某键，则称该键在该集合中的值为 0）。集合的势定义为所有键值对的值的和。

至于如何将序列转换为集合，最简单的方法是转换为包含序列中所有字符的集合。但

这样会丢失序列中字符顺序的信息。另一种方法是，对于某正整数 n，将序列转换为序列中所有 n-gram 的集合，其中 n-gram 指序列中连续的 n 个字符形成的整体（n=1 时即为字符）。由于车牌号长度较短，令 n 取 1、2 即可。

表 3、表 4 分别展示了 n 取 1、2 时，在提供的 9 张图像上，由两个序列（车牌号真实值与识别结果）转换成的 n-gram 的集合的 Jaccard 相似度，其平均值分别约为 0.972、0.968。

表 3 字符集合的 Jaccard 相似度

难度	图像 1	图像 2	图像 3
easy	1	1	1
medium	1	1	1
difficult	0.75	1	1

表 4 2-gram 集合的 Jaccard 相似度

难度	图像 1	图像 2	图像 3
easy	1	1	1
medium	1	1	1
difficult	0.714	1	1

5.2.3 讨论

从以上评价指标来看，本算法达到了令人满意的性能。这得益于算法设计过程中，在多层次视觉、多层次任务思想的指导下，我们能够充分利用图像在颜色、形态、纹理等方面的特征，排除干扰、筛选目标，出现问题时，能够精准定位到问题所属的视觉和任务层次，采取针对性措施解决问题，最终达到理想效果。

在提供的 9 张测试图像上，唯一的错误出现在 3-1.jpg 中的“沪”字的识别上。图 14 展示了对 3-1.jpg 进行车牌定位的结果，图 15 展示了字符分割后左起第一个字符的图像。可见，识别错误的主要原因很可能是图像左边存在来自车牌边缘的白边。想要避免这样的错误，可以改进车牌定位算法，使得定位区域能够进一步缩小范围，避开车牌边缘，也可以改进字符分割算法，使得算法能够对噪点有更强的鲁棒性。

图 14 车牌定位的结果

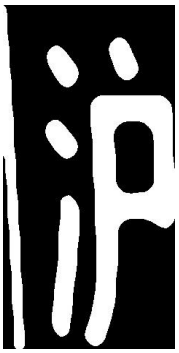


图 15 字符分割后左起第一个字符的图像

参考文献

[1] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, Jieping Ye. Object Detection in 20 Years: A Survey[J/OL]. CoRR abs/1905.05055, 2019.
[2] Alexander Kirillov, Eric Mintun et al. Segment Anything[J/OL]. arXiv: 2304.02643, 2023.
[3] David Marr. Vision[M]. Massachusetts: MIT Press, 2010.
[4] Wikimedia Foundation Inc. HSL and HSV[OL]. 2023-4-11[2023-5-4]. https://en.wikipedia.org/wiki/HSL_and_HSV.
[5] Wikimedia Foundation Inc. Mathematical morphology[OL]. 2023-3-20[2023-5-4]. https://en.wikipedia.org/wiki/Mathematical_morphology.
[6] Andrea Fusiello, Emanuele Trucco, Alessandro Verri. A compact algorithm for rectification of stereo pairs[J]. *Machine Vision and Applications*, 12(1):16–22, 2000-7.
[7] Wikimedia Foundation Inc. Histogram of oriented gradients[OL]. 2023-1-28[2023-5-4]. https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients.
[8] Wikimedia Foundation Inc. k-nearest neighbors[OL]. 2023-4-29[2023-5-4]. https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm.
[9] Wikimedia Foundation Inc. Jaccard index[OL]. 2023-3-12[2023-5-4]. https://en.wikipedia.org/wiki/Jaccard_index.

谢辞

感谢赵老师的授课，在计算机视觉领域的哲学、架构、方法等多个层次给予了我深刻的教诲与启发，让我对计算机视觉乃至人工智能有了更深入的理解。感谢助教的辛勤付出和耐心答疑。