



(12) 发明专利

(10) 授权公告号 CN 111598080 B

(45) 授权公告日 2023. 08. 25

(21) 申请号 202010244367.3

G06V 30/19 (2022.01)

(22) 申请日 2020.03.31

G06V 30/146 (2022.01)

(65) 同一申请的已公布的文献号

申请公布号 CN 111598080 A

(56) 对比文件

CN 101329734 A, 2008.12.24

CN 105279475 A, 2016.01.27

CN 105760471 A, 2016.07.13

US 6373489 B1, 2002.04.16

(43) 申请公布日 2020.08.28

(73) 专利权人 之江实验室

地址 310023 浙江省杭州市余杭区文一西路1818号人工智能小镇10号楼

何东健等. 数字图像处理. 西安电子科技大学出版社, 2015, (第3版), 第254-261页.

(72) 发明人 林宙辰 徐鑫

审查员 尚晓娟

(74) 专利代理机构 杭州求是专利事务有限公司
33200

专利代理师 邱启旺

(51) Int. Cl.

G06V 20/62 (2022.01)

G06V 30/148 (2022.01)

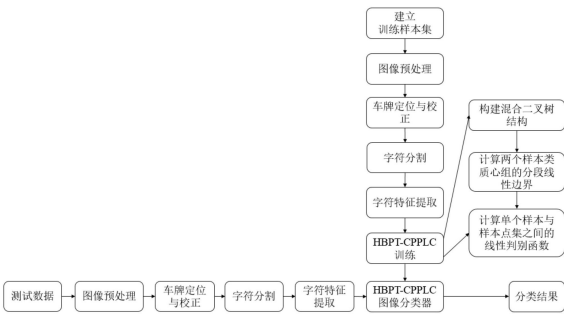
权利要求书3页 说明书16页 附图4页

(54) 发明名称

基于凸多面体分段线性分类的二叉划分树车牌识别方法

(57) 摘要

本发明公开了一种基于凸多面体分段线性分类的二叉划分树车牌识别方法,该方法首先将车牌图像分割成独立的字符图像,计算每类车牌字符图像样本集的质心并创建根节点;然后创建超平面划分质心组,对一个质心的组创建叶节点;对两个质心的组创建内部节点,并通过组合分类器创建超平面分离两个质心分别创建叶节点;对不止两个质心的组创建内部节点再超平面划分,直至所有类到达叶节点分配车牌字符图像样本类标签,建立二叉划分树用于对车牌进行字符识别。本发明基于一个快速、分类能力精度高的二叉划分树识别车牌字符图像,不需要通过冗长的优化过程来调整任何参数,可以处理复杂的多分类任务,具有良好的泛化能力,可实现车牌的快速高精度识别。



1. 一种基于凸多面体分段线性分类的二叉划分树车牌识别方法, 其特征在于, 包括以下步骤:

(1) 获得车牌图像, 提取车牌字符图像样本特征数据;

(2) 根据步骤(1)提取的车牌字符图像样本特征数据和对应的字符类别构建二叉划分树, 包括以下子步骤:

(2.1) 输入K类字符图像样本特征数据集 $T = \{\text{Class}_j | j = 1 \sim K\}$, 设置精度参数 ϵ ; 其中每类样本 $\text{Class}_j = \{\vec{X}_l^j | l = 1, 2, \dots, N_j\}$ 由 N_j 个n维的车牌字符图像样本特征数据 $\vec{X}_l^j = (x_{1l}^j, x_{2l}^j, \dots, x_{nl}^j)$ 组成;

(2.2) 计算每类样本 Class_j 的质心 \vec{C}^j , 并创建一个根节点;

(2.3) 找到距离最远的两个样本类的质心 \vec{C}^p 和 \vec{C}^q , 根据 \vec{C}^p 和 \vec{C}^q 将所有样本类的质心分成 G_{C^p} 和 G_{C^q} 两组, 包括以下子步骤:

(2.3.1) 初始化两个组 G_{C^p} 和 G_{C^q} 为:

$$G_{C^p} = \{\vec{C}^p\}, \quad G_{C^q} = \{\vec{C}^q\}$$

(2.3.2) 根据其它样本类的质心到 \vec{C}^p 和 \vec{C}^q 的距离, 将其它样本类的质心分配到与其最近的组中, 并计算超平面分离组 G_{C^p} 和 G_{C^q} ;

(2.4) 根据步骤(2.3)得到的组 G_{C^p} 和 G_{C^q} , 创建根节点的子节点为内部节点或叶节点, 具体为: 如果组 G_{C^p} 或 G_{C^q} 中只包含一个质心, 则在根节点下创建对应的叶节点;

如果组 G_{C^p} 或 G_{C^q} 中包含两个质心, 则在根节点下创建该组对应的内部节点, 并计算两个质心的组的分段线性边界分离质心组, 对两个质心分别创建叶节点作为内部节点的子节点;

如果组 G_{C^p} 或 G_{C^q} 中不止两个质心, 则在根节点下创建该组对应的内部节点, 根据步骤(2.3)的操作找到组内最远的两个质心并计算超平面分离质心组, 根据新的质心组中质心的个数创建内部节点的子节点, 如只有一个质心则对其创建叶节点, 如有两个质心则对其创建内部节点并计算两个质心的组的分段线性边界分离质心组后创建对应两个叶节点, 如不止两个质心则对其创建内部节点并计算超平面分离质心组后再根据新的质心组中质心的个数创建子节点, 直到子节点均为叶节点;

(2.5) 为步骤(2.4)创建的叶节点分配对应的样本类的标签, 最终得到二叉划分树;

(3) 将待识别车牌图像经过步骤(1)的操作得到每个车牌字符图像的样本特征数据后, 输入步骤(2)构建的二叉划分树, 找到样本特征数据对应的分类, 最后得到车牌字符的识别结果。

2. 根据权利要求1所述基于凸多面体分段线性分类的二叉划分树车牌识别方法, 其特征在于, 所述步骤(2.4)中计算两个质心的组的分段线性边界分离质心组, 包括以下子步骤:

(2.4.1) 将两个质心所属类 Class_j 的样本作为点集 X, Y ;

(2.4.2) 初始化线性判别函数集 $LDFS = \emptyset$;

(2.4.3) 计算线性判别函数候选集 $CLDFS$;

$$CLDFs = \{f_r(\vec{x}) = f(\vec{y}_r, X, \varepsilon) = \langle \vec{w}_r, \vec{x} \rangle + b_r, \vec{y}_r \in Y\}$$

如果点集Y存在 \vec{y}_r 使得 $f_r(\vec{x})$ 不存在,则从点集Y中删除 \vec{y}_r 后跳转至步骤(2.4.2);

(2.4.4) 根据下式找到距离凸包点集CH(X)最近的点 \vec{y}_p :

$$p = \arg \min_r \|\vec{w}_r\|, 1 \leq r \leq |Y|$$

(2.4.5) 从步骤(2.4.3)得到的线性判别函数候选集CLDFs中选择步骤(2.4.4)中 \vec{y}_p 对应的 $f_p(\vec{x})$,将其放入线性判别函数集LDFs;

(2.4.6) 去掉点集Y中满足 $f_p(\vec{y}_r) < 0$ 的 \vec{y}_r ;

(2.4.7) 重复迭代步骤(2.4.4) ~ (2.4.6)直到点集 $Y = \emptyset$,输出线性判别函数集LDFs,得到分段线性边界。

3. 根据权利要求2所述基于凸多面体分段线性分类的二叉划分树车牌识别方法,其特征在于,所述步骤(2.4.3)中计算线性判别函数候选集CLDFs,涉及到计算多个独立的 $\vec{y}_r \in Y$ 与凸包点集CH(X)之间的线性判别函数LDFs,包括以下子步骤:

(2.4.3.1) 取任意 $\vec{x}_{new}^* \in X$;

(2.4.3.2) 初始化 \vec{y}_r 到凸包点集CH(X)中距离最近的 $\vec{x}^* = \vec{x}_{new}^*$;其中, $\vec{y}_r \in Y$;

(2.4.3.3) 在点集X中找到满足如下条件的 \vec{x}_s :

$$m(\vec{x}_t) = \frac{\langle \vec{x}_t - \vec{y}_r, \vec{x}^* - \vec{y}_r \rangle}{\|\vec{x}^* - \vec{y}_r\|}, \vec{x}_t \in X$$

$$s = \arg \min_{t \in |X|} \{m(\vec{x}_t)\}$$

(2.4.3.4) 通过下式更新 \vec{x}_{new}^* :

$$q = \min \left\{ 1, \frac{\langle \vec{x}^* - \vec{y}_r, \vec{x}^* - \vec{x}_s \rangle}{\|\vec{x}^* - \vec{x}_s\|} \right\}$$

$$\vec{x}_{new}^* = \begin{cases} \vec{x}^* + q(\vec{x}_s - \vec{x}^*), & 0 < q < 1 \\ \vec{x}_s, & q = 1 \end{cases}$$

(2.4.3.5) 重复迭代步骤(2.4.3.2) ~ (2.4.3.4)直到满足如下条件:

$$\|\vec{x}^* - \vec{y}_r\| - \|\vec{x}_{new}^* - \vec{y}_r\| < \varepsilon$$

(2.4.3.6) 遍历 $\vec{y}_r \in Y$,计算线性判别函数 $f_r(\vec{x}) = \langle \vec{w}_r, \vec{x} \rangle + b_r$,得到分段线性边界 $f_r(\vec{x}) = 0$,将 $f_r(\vec{x})$ 放入线性判别函数候选集CLDFs;其中, $\vec{w}_r = \vec{x}_{new}^* - \vec{y}_r$, $b_r = \frac{\|\vec{y}_r\|^2 - \|\vec{x}_{new}^*\|^2}{2}$;如果存在 $\vec{x}_z \in X$ 使得 $f_r(\vec{x}_z) = \langle \vec{w}_r, \vec{x}_z \rangle + b_r \leq 0$,则 $f_r(\vec{x})$ 不存在。

4. 根据权利要求1所述基于凸多面体分段线性分类的二叉划分树车牌识别方法,其特征在于,所述步骤(2.2)中计算每个样本类Class_j的质心 \vec{c}^j 为:

$$\vec{c}^j = (c_1^j, c_2^j, \dots, c_i^j, \dots, c_n^j), \quad i = 1 \sim n$$

$$c_i^j = \frac{1}{N_j} \sum_{l=1}^{N_j} x_{il}^j。$$

5. 根据权利要求1所述基于凸多面体分段线性分类的二叉划分树车牌识别方法, 其特征在于, 所述步骤(2.3.2)中超平面通过下式的得到:

$$\sum_{i=1}^n \left\{ (c_i^p - c_i^q) \cdot x_i - \frac{1}{2} [(c_i^p)^2 - (c_i^q)^2] \right\} = 0$$

其中, c_i^p 、 c_i^q 为质心 \vec{c}^p 、 \vec{c}^q 所属类中的第i维坐标。

6. 根据权利要求1所述基于凸多面体分段线性分类的二叉划分树车牌识别方法, 其特征在于, 所述精度参数 $\varepsilon = 10^{-3}$ 。

7. 根据权利要求1所述基于凸多面体分段线性分类的二叉划分树车牌识别方法, 其特征在于, 所述字符图像的种类的集合为{"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","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"}, 共有K=69个类。

8. 根据权利要求1所述基于凸多面体分段线性分类的二叉划分树车牌识别方法, 其特征在于, 所述步骤(1)包括以下子步骤:

(1.1) 对车牌图像进行图像预处理, 包括灰度化、灰度拉伸、同态滤波、图像平滑;

(1.2) 定位并校正车牌图像中的字符区域, 包括以下子步骤:

(1.2.1) 采用OTSU算法对步骤(1.1)预处理的车牌图像二值化, 区分前景像素点和背景像素点; (1.2.2) 采用连通域标记法标记步骤(1.2.1)得到的二值化图像, 划分连通域;

(1.2.3) 删除步骤(1.2.2)得到的非字符连通域;

(1.2.4) 根据连通域的距离合并步骤(1.2.3)保留的连通域划分为车牌动态数组, 并删除合并的连通域少于5的车牌动态数组, 剩余车牌动态数组中的连通域的最小外接矩形区域为车牌图像中的字符区域;

(1.2.5) 采用拉东变换对步骤(1.2.4)中字符区域内的车牌图像进行倾斜校正, 得到车牌字符图像;

(1.3) 采用水平投影法分割车牌字符图像得到独立的字符图像;

(1.4) 分别提取步骤(1.3)得到的字符图像的HOG特征和LBP特征; 其中, HOG特征的维度为 n_h , LBP特征的维度为 n_l ;

(1.5) 组合步骤(1.4)得到的HOG特征和LBP特征得到最终的车牌字符图像样本特征数据, 维度为 $n = n_h + n_l$ 。

基于凸多面体分段线性分类的二叉划分树车牌识别方法

技术领域

[0001] 本发明属于机器学习中的模式分类和分段线性分类领域,尤其涉及一种基于凸多面体分段线性分类的二叉划分树车牌识别方法。

背景技术

[0002] 近年来,深度学习取得了显著的发展。然而,由于其较高的计算成本和模型复杂度,它在一些场景中的应用依然是有限的,如小型侦察机器人、智能相机和便携式设备等。由于分段线性分类器(PLC)可以使用超平面的组合来逼近真实的分类边界,因此实现起来非常简单,并且对内存的要求很低。在车牌识别问题中,PLC的设计一直是一个备受关注的研究课题。

[0003] 在过去的几十年里,人们提出了许多构建PLC的方法。线性规划是一个非常早期的方法,但是模型需要进行多次优化,计算量很大。局部训练方法在PLC的设计中表现良好,但往往导致过拟合或欠拟合问题。决策树是构建PLC的另一种策略,但模型往往需要花费大量的时间来使错误率低于某个可接受的阈值。PLC也可以通过最大-最小可分性的方法来构建,它是多面体可分性的一种泛化方法,但是最大-最小可分性方法可能需要一组预先指定的整数来描述如何组织线性函数。

[0004] 除了上述方法外,几何方法通常能提供一个简单直观的解决方案。典型的模式是利用输入点的凸包来学习分类器。众所周知的支持向量机(SVM)可以将求解最优超平面的问题转化为寻找线性可分条件下凸包之间的最近点的问题。事实上,支持向量机的几何解释在特征表示上是等价的。为了实现偶性,一些著名的最近点算法,如Gilbert-Schlesinger-Kozinec(GSK)和Mitchell-demyanov-malozemov(MDM)被提出来直接解决线性可分问题。此外,多凸线性感知器(multiconlitron)作为PLC设计的一般几何方法被提了出来,其原始的构造算法称为SMA算法。利用SMA,能生成一系列的超平面,并组装成一个多连通体。凸多面体学习为分段线性分类问题提供了另一种几何求解方法。凸多面体学习的目的是构造一个凸多面体分类器,它在一些数据不平衡的特殊分类问题上能表现出良好的效果。

[0005] 然而,现有的凸多面体分类器存在其局限性。第一点,合成一个具有合适数量超平面的凸多面体分类器是一个具有挑战性的任务,现有的构造凸多面体分类器的方法计算量大,包含太多的参数,因此调优过程非常困难。第二点,现有凸多面体分类方法本质上都局限于二分类,但大多数真实的场景都对应于多分类问题。

发明内容

[0006] 本发明的目的在于针对现有技术的不足,提供一种基于凸多面体分段线性分类的二叉划分树车牌识别方法。

[0007] 本发明的目的是通过以下技术方案来实现的:一种基于凸多面体分段线性分类的二叉划分树车牌识别方法,包括以下步骤:

- [0008] (1) 获得车牌图像, 提取车牌字符图像样本特征数据。
- [0009] (2) 根据步骤 (1) 提取的车牌字符图像样本特征数据和对应的字符类别构建二叉划分树, 包括以下子步骤:
- [0010] (2.1) 输入K类字符图像样本特征数据集 $T = \{Class_j | j=1 \sim K\}$, 设置精度参数 ε 等; 其中每类样本 $Class_j = \{\vec{X}_l^j | l = 1, 2, \dots, N_j\}$ 由 N_j 个 n 维的车牌字符图像样本特征数据 $\vec{X}_l^j = (x_{1l}^j, x_{2l}^j, \dots, x_{nl}^j)$ 组成。
- [0011] (2.2) 计算每类样本 $Class_j$ 的质心 \vec{C}^j , 并创建一个根节点。
- [0012] (2.3) 找到距离最远的两个样本类的质心 \vec{C}^p 和 \vec{C}^q , 根据 \vec{C}^p 和 \vec{C}^q 将所有样本类的质心分成 G_{C^p} 和 G_{C^q} 两组, 包括以下子步骤:
- [0013] (2.3.1) 初始化两个组 G_{C^p} 和 G_{C^q} 为:
- [0014] $G_{C^p} = \{\vec{C}^p\}, G_{C^q} = \{\vec{C}^q\}$
- [0015] (2.3.2) 根据其它样本类的质心到 \vec{C}^p 和 \vec{C}^q 的距离, 将其它样本类的质心分配到与其最近的组中, 并计算超平面分离组 G_{C^p} 和 G_{C^q} 。
- [0016] (2.4) 根据步骤 (2.3) 得到的组 G_{C^p} 和 G_{C^q} , 创建根节点的子节点为内部节点或叶节点, 具体为:
- [0017] 如果组 G_{C^p} 或 G_{C^q} 中只包含一个质心, 则在根节点下创建对应的叶节点。
- [0018] 如果组 G_{C^p} 或 G_{C^q} 中包含两个质心, 则在根节点下创建该组对应的内部节点, 并计算两个质心的组的分段线性边界分离质心组, 对两个质心分别创建叶节点作为内部节点的子节点。
- [0019] 如果组 G_{C^p} 或 G_{C^q} 中不止两个质心, 则在根节点下创建该组对应的内部节点, 根据步骤 (2.3) 的操作找到组内最远的两个质心并计算超平面分离质心组, 根据新的质心组中质心的个数创建内部节点的子节点, 如只有一个质心则对其创建叶节点, 如有两个质心则对其创建内部节点并计算两个质心的组的分段线性边界分离质心组后创建对应两个叶节点, 如不止两个质心则对其创建内部节点并计算超平面分离质心组后再根据新的质心组中质心的个数创建子节点, 直到子节点均为叶节点。
- [0020] (2.5) 为步骤 (2.4) 创建的叶节点分配对应的样本类的标签, 最终得到二叉划分树。
- [0021] (3) 将待识别车牌图像经过步骤 (1) 的操作得到每个车牌字符图像的样本特征数据后, 输入步骤 (2) 构建的二叉划分树, 找到样本特征数据对应的分类, 最后得到车牌字符的识别结果。
- [0022] 进一步地, 所述步骤 (2.4) 中计算两个质心的组的分段线性边界分离质心组, 包括以下子步骤:
- [0023] (2.4.1) 将两个质心所属类 $Class_j$ 的样本作为点集 X, Y ;
- [0024] (2.4.2) 初始化线性判别函数集 $LDFs = \emptyset$;
- [0025] (2.4.3) 计算线性判别函数候选集 $CLDFs$:

[0026] $CLDFs = \{f_r(\vec{x}) = f(\vec{y}_r, X, \varepsilon) = \langle \vec{w}_r, \vec{x} \rangle + b_r, \vec{y}_r \in Y\}$

[0027] 如果点集Y存在 \vec{y}_r 使得 $f_r(\vec{x})$ 不存在,则从点集Y中删除 \vec{y}_r 后跳转至步骤(2.4.2);

[0028] (2.4.4)根据下式找到距离凸包点集CH(X)最近的点 \vec{y}_p :

[0029] $p = \arg \min_r \|\vec{w}_r\|, 1 \leq r \leq |Y|$

[0030] (2.4.5)从步骤(2.4.3)得到的线性判别函数候选集CLDFs中选择步骤(2.4.4)中 \vec{y}_p 对应的 $f_p(\vec{x})$,将其放入线性判别函数集LDFs;

[0031] (2.4.6)去掉点集Y中满足 $f_p(\vec{y}_r) < 0$ 的 \vec{y}_r ;

[0032] (2.4.7)重复迭代步骤(2.4.4)~(2.4.6)直到点集 $Y = \emptyset$,输出线性判别函数集LDFs,得到分段线性边界。

[0033] 进一步地,所述步骤(2.4.3)中计算线性判别函数候选集CLDFs,涉及到计算多个独立的 $\vec{y}_r \in Y$ 与凸包点集CH(X)之间的线性判别函数LDFs,包括以下子步骤:

[0034] (2.4.3.1)取任意 $\vec{x}_{new}^* \in X$;

[0035] (2.4.3.2)初始化 \vec{y}_r 到凸包点集CH(X)中距离最近的 $\vec{x}^* = \vec{x}_{new}^*$;其中, $\vec{y}_r \in Y$;

[0036] (2.4.3.3)在点集X中找到满足如下条件的 \vec{x}_s :

[0037] $m(\vec{x}_t) = \frac{\langle \vec{x}_t - \vec{y}_r, \vec{x}^* - \vec{y}_r \rangle}{\|\vec{x}^* - \vec{y}_r\|}, \vec{x}_t \in X$

[0038] $s = \arg \min_{t \in |X|} \{m(\vec{x}_t)\}$

[0039] (2.4.3.4)通过下式更新 \vec{x}_{new}^* :

[0040] $q = \min \left\{ 1, \frac{\langle \vec{x}^* - \vec{y}_r, \vec{x}^* - \vec{x}_s \rangle}{\|\vec{x}^* - \vec{x}_s\|} \right\}$

[0041] $\vec{x}_{new}^* = \begin{cases} \vec{x}^* + q(\vec{x}_s - \vec{x}^*), & 0 < q < 1 \\ \vec{x}_s, & q = 1 \end{cases}$

[0042] (2.4.3.5)重复迭代步骤(2.4.3.2)~(2.4.3.4)直到满足如下条件:

[0043] $\|\vec{x}^* - \vec{y}_r\| - \|\vec{x}_{new}^* - \vec{y}_r\| < \varepsilon$

[0044] (2.4.3.6)遍历 $\vec{y}_r \in Y$,计算线性判别函数 $f_r(\vec{x}) = \langle \vec{w}_r, \vec{x} \rangle + b_r$,得到分段线性边界 $f_r(\vec{x}) = 0$,将 $f_r(\vec{x})$ 放入线性判别函数候选集CLDFs;其中, $\vec{w}_r = \vec{x}_{new}^* - \vec{y}_r, b_r = \frac{\|\vec{y}_r\|^2 - \|\vec{x}_{new}^*\|^2}{2}$;

如果存在 $\vec{x}_z \in X$ 使得 $f_r(\vec{x}_z) = \langle \vec{w}_r, \vec{x}_z \rangle + b_r \leq 0$,则 $f_r(\vec{x})$ 不存在。

[0045] 进一步地,所述步骤(2.2)中计算每个样本类Class_j的质心 \vec{c}^j 为:

[0046] $\vec{c}^j = (c_1^j, c_2^j, \dots, c_i^j, \dots, c_n^j), \quad i = 1 \sim n$

[0047] 进一步地,所述步骤(2.3.2)中超平面通过下式的得到:

[0048] $\sum_{i=1}^n \left\{ (c_i^p - c_i^q) \cdot x_i - \frac{1}{2} [(c_i^p)^2 - (c_i^q)^2] \right\} = 0$

[0049] 其中, c_i^p 、 c_i^q 为质心 \vec{c}^p 、 \vec{c}^q 所属类中的第 i 维坐标。

[0050] 进一步地, 所述精度参数 $\varepsilon = 10^{-3}$ 。

[0051] 进一步地, 所述字符图像的种类的集合为 {"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", "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"}, 共有 $K=69$ 个类。

[0052] 进一步地, 所述步骤 (1) 包括以下子步骤:

[0053] (1.1) 对车牌图像进行图像预处理, 包括灰度化、灰度拉伸、同态滤波、图像平滑;

[0054] (1.2) 定位并校正车牌图像中的字符区域, 包括以下子步骤:

[0055] (1.2.1) 采用OTSU算法对步骤 (1.1) 预处理的车牌图像二值化, 区分前景像素点和背景像素点;

[0056] (1.2.2) 采用连通域标记法标记步骤 (1.2.1) 得到的二值化图像, 划分连通域;

[0057] (1.2.3) 删除步骤 (1.2.2) 得到的非字符连通域;

[0058] (1.2.4) 根据连通域的距离合并步骤 (1.2.3) 保留的连通域划分为车牌动态数组, 并删除合并的连通域少于5的车牌动态数组, 剩余车牌动态数组中的连通域的最小外接矩形区域为车牌图像中的字符区域;

[0059] (1.2.5) 采用拉东变换对步骤 (1.2.4) 中字符区域内的车牌图像进行倾斜校正, 得到车牌字符图像;

[0060] (1.3) 采用水平投影法分割车牌字符图像得到独立的字符图像;

[0061] (1.4) 分别提取步骤 (1.3) 得到的字符图像的HOG特征和LBP特征; 其中, HOG特征的维度为 n_h , LBP特征的维度为 n_l ;

[0062] (1.5) 组合步骤 (1.4) 得到的HOG特征和LBP特征得到最终的车牌字符图像样本特征数据, 维度为 $n = n_h + n_l$ 。

[0063] 本发明的有益效果是: 本发明基于凸多面体分段线性分类的二叉划分树车牌识别方法, 是一种基于速度更快、分类能力更强、精度更高的分类器的车牌识别方法, 可以处理复杂的多分类任务, 具有良好的泛化能力, 可实现数据样本的快速分类, 并且可以调整精度; 实现起来非常简单, 具有直观的几何解释和较低的内存要求; 不需要执行冗长的优化过程, 也不需要调优任何参数; 在分类效率和计算时间上都优于现有的分段线性分类器, 其计算复杂度更低, 在维度上的适应性更好, 能以较低的模型复杂度实现较高的车牌字符识别精度; 对于将分段线性分类器嵌入到实时系统或便携式终端中具有重要意义, 也为凸多面体学习提供了一个几何基础, 可以应用于计算几何领域, 解决凸包相关的问题。因此, 本发明是分段线性分类和图像识别领域的一个显著的进步。

附图说明

[0064] 图1为本发明设计的计算 $\vec{y}_r \in Y$ 与凸包点集 $CH(X)$ 之间的超平面的示意图;

[0065] 图2是本发明设计的超平面不存在的特例以及计算两个质心的组的分段线性边界 (CPPLC) 的示意图;

[0066] 图3是本发明提出的基于凸多面体分段线性分类的二叉划分树车牌识别方法(HBPT-CPPLC)的示意图;

[0067] 图4将本发明用于车牌图像识别的流程示意图;

[0068] 图5是同态滤波函数的示意图。

具体实施方式

[0069] 本发明构造了一个基于凸多面体分段线性分类的二叉划分树车牌识别方法(HBPT-CPPLC)。首先从二叉划分树的根节点开始,以相距最远的两个样本类质心的垂直平分线创建一个超平面,将所有样本类按照超平面上的正或负方向分成两组。然后判断根节点的子节点是内部节点或叶节点,如是叶节点,则给待分类样本分配对应类的标签;如是内部节点,则判断其样本类的质心是否超过两个。如果超过两个,重复创建超平面和判断子节点的步骤直到所有节点包含的质心都不超过两个。对于只有两个质心的内部节点,通过两个凸多面体可分的点集,以组合分类器的形式学习一系列超平面,用凸多面体包围内部的点集,排除外部的点集,将一个类与另一个类分离,直到所有节点都到达叶节点。最后,建立了一个二叉划分树,该树表示给定类的层次划分。本发明提出了一个速度更快、分类能力更强的车牌识别器,该算法不需要通过冗长的优化过程来调整任何参数,可以处理复杂的多分类任务,具有良好的泛化能力和快速的运行速度。

[0070] 下面结合附图,通过实施例进一步描述本发明,但不以任何方式限制本发明的范围。本发明可以应用于嵌入式实时终端设备进行车牌图像识别,通过构建混合二叉树架构来设计凸多面体分段线性车牌识别器。

[0071] 实施例:本发明通过车牌图像识别任务来展示提出的基于凸多面体分段线性分类的二叉划分树车牌识别方法的有效性。本发明在分类效率和计算时间上都优于现有的分段线性分类算法,验证了其竞争力和优越性。

[0072] 本实施例流程如图4所示,具体包括如下步骤:

[0073] 步骤1,获取训练样本集和测试样本集;

[0074] 数据集来源于中国科学技术大学开源的车牌识别数据集CCPD(<https://github.com/detectRecog/CCPD>),随机选取数据集中的CCPD-Base部分的20000张车牌图像,其中15000张图像用作训练集,50000张用作测试集。

[0075] 步骤2,进行图像预处理;

[0076] 由于图像中存在许多干扰、噪声和差异,为了将图像变换成适合图像特征提取的形式,需要对车牌图像进行预处理,包括灰度变换、图像滤波、图像平滑。

[0077] 灰度变换是为了增强车牌图像的对比度,车牌位于车身的下部,属于低灰度级范围,而车头上部及背景中一些区域一般较亮,属于高灰度级范围。为了突出牌照的灰度细节,抑制高灰度区域,本发明实施例对车牌图像进行了灰度拉伸。在进行灰度拉伸之前,先对车牌图像进行灰度化,对图像的RGB三个分量进行加权平均得到灰度值。假定车牌图像的灰度级范围为 $[a, b]$,本发明实施例做了如下灰度变换:

$$[0078] \quad g(i, j) = \begin{cases} \frac{a'}{a} f(i, j) & 0 \leq f(i, j) < a \\ \frac{b' - a'}{b - a} [f(i, j) - a] + a' & a \leq f(i, j) < b \\ \frac{c' - b'}{c - b} [f(i, j) - b] + b' & b \leq f(i, j) < c \end{cases}$$

[0079] 上式中, $f(i, j)$ 是灰度变换前坐标为 (i, j) 的像素值, $g(i, j)$ 是灰度变换后坐标为 (i, j) 的像素值。设图像为 $M_g \times N_g$ 像素, 其直方图为 $h(i)$, a 取满足 $\sum_{i=0}^a h(i) \geq \frac{M_g N_g}{10}$ 的最小整数, b 取满足 $\sum_{i=0}^b h(i) \leq \frac{9M_g N_g}{10}$ 的最大整数。 a' 和 b' 分别取 $\frac{a}{2}$ 和 $\frac{3b}{2}$, $c = c' = 256$ 。

[0080] 真实场景下的车牌图片是不同光照条件下的图像, 这是一种灰度级动态范围很大的图像, 其暗处很黑, 亮处很白, 而本发明实施例感兴趣的对象灰度级范围又很小, 分不清物体的灰度层次和细节。这时, 采用一般的灰度线性变换是行不通的, 因为扩展灰度级虽然可以提高物体图像的反差, 但会使动态范围更大。而压缩灰度级, 虽可以减少动态范围, 但物体灰度层次和细节就会更看不清。对这种矛盾状态的处理方法, 本发明实施例采用的图像滤波为同态滤波方法。

[0081] 图像 $g(i, j)$ 是由光源产生的照度场 $g_1(i, j)$ 和目标的反射系数场 $g_2(i, j)$ 共同作用下产生的, 三者的关系是:

$$[0082] \quad g(i, j) = g_1(i, j) g_2(i, j)$$

[0083] 照度场 $g_1(i, j)$ 变化缓慢, 在频谱上其能量集中于低频, 反射系数场 $g_2(i, j)$ 包含了所需要的图像信息, 在空间中的变化较快, 能量集中在高频部分。采用同态滤波方法, 把这两个分量分开处理, 处理步骤如下:

[0084] (1) 对上式两边取对数, 得到两个加性分量, 这是同态滤波方法中的第一步:

$$[0085] \quad \ln g(i, j) = \ln g_1(i, j) + \ln g_2(i, j)$$

[0086] (2) 对上式两边同时进行傅立叶变换, 得到:

$$[0087] \quad G(u, v) = F\{\ln g(i, j)\} = F\{\ln g_1(i, j)\} + F\{\ln g_2(i, j)\} = G_1(u, v) + G_2(u, v)$$

[0088] (3) 为了消除照度的起伏, 应该压缩 $G_1(u, v)$ 分量, 为了增加图像本身的对比度, 应该增强 $G_2(u, v)$ 分量。因此使用同态滤波器来处理 $G(u, v)$, 如图5所示, 其中频域中的点 (u, v) 与频率中心的距离 $D(u, v) = \sqrt{u^2 + v^2}$, 低频增益 $H_L = 0.5$, 高频增益 $H_H = 2$ 。

$$[0089] \quad G^*(u, v) = H(u, v) G(u, v) = H(u, v) G_1(u, v) + H(u, v) G_2(u, v)$$

[0090] (4) 取傅立叶反变换和指数运算, 得到图像 $g_3(i, j)$:

$$[0091] \quad g_3(i, j) = \exp(F^{-1}\{G^*(u, v)\})$$

[0092] 为了减少图像中的噪声, 需要对图像进行平滑处理, 本发明实施例采用的图像平滑是空域的局部平均法, 这种方法易于实现、效果也比较好。经局部平均法处理后的 $g_4(i, j)$ 为:

$$[0093] \quad g_4(i, j) = \frac{1}{N_s} \sum_{(i, j) \in S} [g_3(i, j)]$$

[0094] 其中, S 是点 (i, j) 邻域内的点集, 由于本发明实施例选取的卷积模板为: $\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$,

所以邻域内的点的个数为 $N_s = 16$ 。

[0095] 步骤3, 进行车牌定位和校正。

[0096] 本发明使用OTSU算法对车牌图像进行二值化, 然后通过提取连通域进行车牌定位最后通过Radon拉东变换对定位后的车牌进行旋转校正, 具体包含以下子步骤:

[0097] (1) 采用OTSU算法对步骤2预处理的车牌图片二值化;

[0098] 对一幅 $M_g \times N_g$ 个像素的车牌图像来说, 包含 L 个灰度级 $(0, 1, 2, \dots, L-1)$, 图像总平均灰度 e 计算如下:

$$[0099] \quad e = w_0 e_0 + w_1 e_1$$

[0100] 其中, w_0 表示前景像素点数占整个图像的比例, e_0 表示前景像素点的平均灰度, w_1 表示背景像素点数占整个图像的比例, e_1 表示背景像素点的平均灰度。前景和背景的方差用 E 表示, T_0 为区分前景像素点和背景像素点的灰度级, 本发明实施例选取使得方差 E 最大的灰度级 T_0 作为OTSU算法二值化阈值, 方差 E 的计算公式如下:

$$[0101] \quad E = w_0 (e - e_0)^2 + w_1 (e - e_1)^2$$

[0102] 对步骤2中预处理的图像, 其二值化的计算公式为:

$$[0103] \quad g_5(i, j) = \begin{cases} 255, & g_4(i, j) \geq T_0 \\ 0, & g_4(i, j) \leq T_0 \end{cases}$$

[0104] 其中, 0 表示白色的背景像素点, 255 表示黑色的前景像素点数。

[0105] (2) 对二值化后的车牌图像基于连通域进行车牌定位, 具体包含以下子步骤:

[0106] (2.1) 连通域标记。对每个像素的邻域进行连通域标记。本发明实施例采用8邻域进行标记, 具体定义为:

$$[0107] \quad N_8(i, j) =$$

$$[0108] \quad \{(i-1, j-1), (i-1, j), (i-1, j+1), (i, j-1), (i, j+1), (i+1, j-1), (i+1, j), (i+1, j+1)\}$$

[0109] 首先初始化连通域标记矩阵 $\{g_c(i, j) \mid i=0 \sim M_g-1; j=0 \sim N_g-1\}$ 的值都为0, 用数标 1_c 标记黑色像素点, 初始化数标 1_c 为1; 然后遍历车牌图像 $\{g_5(i, j) \mid i=0 \sim M_g-1; j=0 \sim N_g-1\}$, 从图像的左上角开始, 从左到右, 从上到下扫描每个像素点, 当扫描到黑色像素点时, 为该点标记当前的数标 1_c 值, 其中每次扫描至某个像素点的左 $(i-1, j)$ 、左上 $(i-1, j+1)$ 、上 $(i, j+1)$ 、右上 $(i+1, j+1)$ 的像素值为0时, 将当前的数标 1_c 值加1, 得到第一次扫描标记后的 $g_c(i, j)$ 。

[0110] 经过第一次扫描后, 可能有的标记会出现偏差。本发明实施例为保证标记的准确性, 进行第二次扫描, 具体方式如下: 从上到下, 从左到右扫描标记后的图像 $g_c(i, j)$, 判断已标记像素点(黑色)的左、右上的像素点是否标记, 如果这两个邻域点均已标记, 则比较标记的数标 1_c 值是否一样; 如果一样, 则当前点数标值不变; 如果右上点和左前点数标值不同, 则当前点的数标值取左点的数标值。

[0111] 经过两次扫描后, 把相同数标值的像素点归为一个连通域, 得到图像标记后的连通域 $Array_c, c=1, 2, \dots, N_c, N_c$ 表示连通域的个数。

[0112] (2.2) 删除非字符的连通域,具体为对步骤(2.1)得到的连通域作最小外接矩形,最小外接矩形的提取公式为:

$$[0113] \quad \text{Rect}_c = \{\text{Array}_c^L(i_L, j_L), \text{Array}_c^R(i_R, j_R), \text{Array}_c^B(i_B, j_B), \text{Array}_c^T(i_T, j_T)\}$$

[0114] 其中, $\text{Array}_c^L, \text{Array}_c^R, \text{Array}_c^B, \text{Array}_c^T$ 分别是连通域外接矩形最左边,最右边,最底部,最顶部的像素点坐标。根据最小外接矩形的面积 Rect_s 、最小外接矩形宽度 Rect_w 、最小外接矩形高度 Rect_h 和最小外接矩形高宽比 Rect_D 来判断是否是字符的连通域:

$$[0115] \quad \text{Rect}_s = (i_R - i_L) * (j_T - j_B)$$

$$[0116] \quad \text{Rect}_w = i_R - i_L$$

$$[0117] \quad \text{Rect}_h = j_T - j_B$$

$$[0118] \quad \text{Rect}_D = \frac{\text{Rect}_h}{\text{Rect}_w}$$

[0119] 本实施例中,当连通域的最小外接矩形的上述四个参数在以下范围内时,判断为车牌字符连通域:

$$[0120] \quad \text{Rect}_s > 1500, 30 < \text{Rect}_w < 55, 75 < \text{Rect}_h < 100, 1.5 < \text{Rect}_D < 2.5$$

[0121] 得到新的连通域 $\text{Array}_c, c = 1 \sim N_n, N_n$ 表示剔除非字符后的连通域的个数。

[0122] (2.3) 连通域的合并

[0123] 遍历步骤(2.2)得到的 Array_c , 任意两个连通域 Array_{c1} 与 Array_{c2} 的最小外接矩形定义为:

$$[0124] \quad \text{Rect}_{c1} = \{\text{Array}_{c1}^L(i_{L1}, j_{L1}), \text{Array}_{c1}^R(i_{R1}, j_{R1}), \text{Array}_{c1}^B(i_{B1}, j_{B1}), \text{Array}_{c1}^T(i_{T1}, j_{T1})\}$$

$$[0125] \quad \text{Rect}_{c2} = \{\text{Array}_{c2}^L(i_{L2}, j_{L2}), \text{Array}_{c2}^R(i_{R2}, j_{R2}), \text{Array}_{c2}^B(i_{B2}, j_{B2}), \text{Array}_{c2}^T(i_{T2}, j_{T2})\}$$

[0126] 当其最小横坐标之差 $(i_{L2} - i_{R1})$ 小于设定的阈值(这里设定为5个像素点)、最大纵坐标之差 $(j_{T2} - j_{B1})$ 也小于设定的阈值(这里设定为105个像素点),则将两个连通域合并并划分为一个车牌动态数组,记录到 Array_p 中,然后重复上述操作直到不能合并连通域为止。由于车牌有七个字符,删除合并的连通域个数小于5的数组,最后对合并后的连通域再做最小外接矩形得到车牌定位结果 $g_6(i, j)$ 。

[0127] (3) 使用Radon拉东变换对定位后的车牌图像 $g_6(i, j)$ 进行校正, Radon拉东变换是指图像按照某一角度方向上进行投影,投影的距离处于极值时所对应的角度为车牌的倾斜角,其投影的计算公式为:

$$[0128] \quad R(i') = \int_{-\infty}^{+\infty} g_6(i' \cos \theta - j' \sin \theta, i' \sin \theta + j' \cos \theta) dj'$$

$$[0129] \quad \text{其中}, \begin{bmatrix} i' \\ j' \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \times \begin{bmatrix} i \\ j \end{bmatrix}。$$

[0130] 首先使用Radon拉东变换求水平倾斜角,对车牌图像在0度到180度进行投影,使得投影后的非零值相加达到最大时的角度 θ_1 , 车牌的水平倾斜角 $\alpha = 90 - \theta_1$ 。然后使用Radon拉东变换求垂直倾斜角,对车牌图像在-45度到45度进行投影,使得投影后的非零值相加达到最大时的角度 θ_2 , 车牌的垂直倾斜角 $\beta = \theta_2 - 45$ 。采用获得的水平倾斜角和垂直倾斜角,通过剪切变换对二值化后的图像进行校正,得到图像 $g_7(i, j)$ 。

$$[0131] \quad g_7(i, j) = \begin{bmatrix} 1 & -\sin(\alpha \times \pi/180) \\ -\sin(\beta \times \pi/180) & 1 \end{bmatrix} \times g_6(i, j)$$

[0132] 步骤4,进行车牌字符分割。

[0133] 本发明对车牌图像 $g_7(i, j)$ 自左向右逐列统计各列所包含的像素点,采用的是一种对水平投影法进行改进的车牌字符分割法,将字符分割后的结果存储到一个 $1 \times d$ 向量中 $\{g_{7d}(i, j) | d=1, 2, \dots, 7, d \text{ 为车牌上字符的个数,包含以下步骤:}$

[0134] (1) 利用车牌图像中,第一个字符是汉字的特点。首先对车牌图像 $g_7(i, j)$ 自左向右逐列扫描,将各列包含的黑色像素点的数目与某一固定阈值(本实施例为2个像素点)进行比较,当遇到第一个黑色像素点数大于固定阈值的列时,则该列即被认为是字符分割的起始位置,然后继续扫描下一列,直至遇到黑色像素点数小于固定阈值的列时停止。为了克服水平投影法对解决汉字不连通问题的不足,本发明将该列与字符分割的起始列的黑色像素点数作差,其差值再与另一固定阈值(本实施例设定为车牌图像宽度的九分之一)作比较,若差值大于等于阈值,则说明第一个字符分割结束;若小于阈值,则说明分割得到的不是一个完整的汉字,还需继续扫描下一列,直至第一个字符分割结束,得到汉字字符图像样本。第一个汉字字符的集合为{"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"},总共33类。

[0135] (2) 车牌图像中第二到第七个字符,一般为字母或数字,不存在不连通的问题,其分割方法较第一个汉字字符简单。自左向右扫描图像各列,当遇到黑色像素点数大于固定阈值(本实施例为2个像素点)的列时,则该列即被认为是字符分割的起始位置,继续扫描,直至黑色像素点数小于固定阈值(车牌宽度的六分之一)时,第二个字符分割结束。利用相同的方法,可以分割出其它的字符,得到非汉字字符图像样本。第二到第七个字符的集合为{'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'},总共36类。

[0136] 步骤5,车牌字符特征提取;

[0137] 将车牌分割后的字符图像样本在HOG特征上结合LBP特征进行特征提取,从形状、纹理特征来描述目标特征,形成包含充分信息量的低维的特征向量集合。HOG特征能充分提取车牌的形状信息,具备区分车牌与其它目标的能力;LBP特征计算高效且对单调的灰度级变化具有不变性。将窗口内的HOG特征和LBP直方图结合起来,也就结合了形状特征和纹理特征,能够显著提升检测性能。具体包含以下子步骤:

[0138] (1) 提取HOG特征:

[0139] 对步骤4分割出来的字符图像 $\{g_{7d}(i, j) | d=1, 2, \dots, 7$ 进行HOG特征提取。假设字符图像为 $M_d \times N_d$ 个像素,设置block块的大小 8×8 ,块步长为8,则字符图像共包含 $\frac{M_d N_d}{64}$ 个不重叠的block。在每个block上计算其梯度方向和幅值。本发明实施例采用中心对称算子 $[-1, 0, 1]$ 计算梯度,如下式所示:

$$[0140] \quad \theta(i, j) = \frac{\tan^{-1}(g_{7d}(i, j+1) - g_{7d}(i, j-1))}{g_{7d}(i, j+1) - g_{7d}(i, j-1)}$$

$$[0141] \quad m_r(i, j) = \sqrt{(g_{7d}(i+1, j) - g_{7d}(i-1, j))^2 + (g_{7d}(i, j+1) - g_{7d}(i, j-1))^2}$$

[0142] 其中, $\theta(i, j)$ 为像素点 (i, j) 处的梯度方向, $m_r(i, j)$ 为像素点 (i, j) 处的幅度值。

[0143] 设置cell大小为 4×4 ,在每个block按cell大小统计梯度直方图,应用梯度的幅度值进行权重投影。将所有block内的直方图向量组合得到最终的HOG特征向量 $\vec{f}_h\{1,2,3, \dots, n_h\}$,其特征向量维度为 $n_h = 36 \left(1 + \frac{M_d-8}{4}\right) \left(1 + \frac{N_d-8}{4}\right)$ 。

[0144] (2)提取LBP特征:

[0145] 对步骤4分割出来的字符图像 $\{g_{7d}(i, j) \mid d=1, 2, \dots, 7\}$ 进行LBP特征提取,具体包含以下步骤:

[0146] (2.1)用LBP算子对字符图像进行特征提取。

$$[0147] \quad LBP(i_c, j_c) = \sum_{x=0}^{x-1} 2^x s(L_x - L_c)$$

[0148] 其中, (i_c, j_c) 为 3×3 领域内的中心像素, L_c 是灰度值, L_x 是相邻像素的灰度值, s 是一个符号函数 $s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$ 。

[0149] (2.2)将特征图像进行分块,分成8行8列,共64块区域。

[0150] (2.3)计算每块区域特征图像的直方图,将直方图进行归一化,由于其选取 3×3 领域算子,其LBP特征值的模式为256种,直方图大小为 1×256 。

[0151] (2.4)将每块区域特征图像的直方图按分块的空间顺序排列成一行,形成LBP特征向量 $\vec{f}_l\{1,2,3, \dots, n_l\}$,其特征向量维度为 $n_l = 64 \times 256$ 。

[0152] (3)组合HOG+LBP特征

[0153] 提取完HOG特征和LBP特征后,将两种特征向量联合起来得到联合特征 $\vec{f}_{hl}\{1,2,3, \dots, n\}$,其维度为 $n = n_h + n_l$ 。

[0154] 步骤6,基于HBPT-CPPLC方法进行分类训练;

[0155] 通过步骤5将字符图像样本映射为特征向量后,下一步本发明实施例通过HBPT-CPPLC方法在特征空间中训练一个分类器,把不同类别的样本集合分开,包含三部分:构建二叉划分树结构来实现多类样本的层次划分,计算两个样本类质心组的分段线性边界(CPPLC算法)和计算单个样本与样本点集之间的线性判别函数,最终输出具有二叉划分树结构的线性判别函数集合的分类边界,得到HBPT-CPPLC分类器。

[0156] 对于构建二叉划分树结构来实现多类样本的层次划分,包括以下步骤:

[0157] (1)输入步骤5得到的69类字符样本特征向量集合 $T = \{Class_j \mid j=1 \sim 69\}$,设置精度参数 $\varepsilon = 10^{-3}$;其中每类样本 $Class_j = \{\vec{X}_l^j \mid l = 1 \sim N_j\}$ 由 N_j 个 n 维特征向量 $\vec{X}_l^j = (x_{1l}^j, x_{2l}^j, \dots, x_{nl}^j)$ 组成;

[0158] (2)计算每类样本 $Class_j$ 的质心 \vec{C}^j ,并创建一个根节点;

$$[0159] \quad \vec{C}^j = (c_1^j, c_2^j, \dots, c_i^j, \dots, c_n^j), \quad i = 1 \sim n$$

[0160] 其中 $c_i^j = \frac{1}{N_j} \sum_{l=1}^{N_j} x_{il}^j$ 为第 j 类字符样本的个数;

[0161] (3)找到距离最远的两类字符样本的质心 \vec{C}^p 和 \vec{C}^q ,根据 \vec{C}^p 和 \vec{C}^q 将所有类字符样本

的质心分成 G_{C^p} 和 G_{C^q} 两组;例如图3(a)中的 $\vec{C}^1, \vec{C}^2, \vec{C}^3, \vec{C}^4$ 四个质心分别被分配到 $G_{C^p}(\vec{C}^1, \vec{C}^3)$ 或 $G_{C^q}(\vec{C}^2, \vec{C}^4)$;包括以下步骤:

[0162] (3.1)初始化两个组 G_{C^p} 和 G_{C^q} 为:

[0163] $G_{C^p} = \{\vec{C}^p\}, G_{C^q} = \{\vec{C}^q\}$

[0164] (3.2)根据其它类字符样本的质心到 \vec{C}^p 和 \vec{C}^q 的距离,将其它类字符样本的质心分配到与其最近的组中,并通过式1计算超平面H分离组 G_{C^p} 和 G_{C^q} :

$$[0165] \sum_{i=1}^n \left\{ (c_i^p - c_i^q) \cdot x_i - \frac{1}{2} [(c_i^p)^2 - (c_i^q)^2] \right\} = 0 \quad (\text{式1})$$

[0166] 如图3(b)所示,超平面 H_1 是连接着第一类和第二类字符样本的质心 \vec{C}^1 和 \vec{C}^2 的线段的垂直平分线;

[0167] (4)根据步骤(3)得到的组 G_{C^p} 和 G_{C^q} ,将字符样本特征空间按顺序划分为较小的子区域,进一步创建内部节点或叶节点,并分配叶节点对应的字符样本类的类别标签,具体为:

[0168] 如果 G_{C^p} 或 G_{C^q} 中只包含一个质心,对应一个字符样本类,则在根节点下创建对应该类的叶节点;

[0169] 如果 G_{C^p} 或 G_{C^q} 中包含两个质心,包含两个字符样本类,则在根节点下创建该组对应的内部节点,并采用CPPLC算法计算该组的分段线性边界分离质心,对两个质心分别创建叶节点作为内部节点的子节点;

[0170] 如果 G_{C^p} 或 G_{C^q} 中不止两个质心,包含两个以上的字符样本类,则在根节点下创建该组对应的内部节点,根据步骤(3)的操作找到组内最远的两个质心并计算超平面分离质心组,根据新的质心组中质心的个数创建内部节点的子节点,如只有一个质心则对其创建叶节点,如有两个质心则对其创建内部节点并采用CPPLC算法计算分段线性边界分离质心后创建对应两个叶节点,如不止两个质心则对其创建内部节点并计算超平面分离质心组后再根据新的质心组中质心的个数创建子节点,直到子节点均为叶节点。

[0171] 对于步骤(4)中计算两个样本类质心组的分段线性边界(CPPLC算法),包括以下子步骤:

[0172] (4.1)将两个质心所属样本类 $Class_j$ 的集合作为点集 X, Y 。

[0173] (4.2)初始化线性判别函数集 $LDFs = \emptyset$;

[0174] (4.3)计算线性判别函数候选集CLDFs:

[0175] $CLDFs = \{f_r(\vec{x}) = f(\vec{y}_r, X, \varepsilon) = \langle \vec{w}_r, \vec{x} \rangle + b_r, \vec{y}_r \in Y\}$

[0176] 如果点集Y存在 \vec{y}_r 使得 $f_r(\vec{x}) = NULL$,则从点集Y中删除 \vec{y}_r 后跳转至步骤(4.2);

[0177] 对于步骤(4.3)中计算线性判别函数候选集CLDFs,涉及到计算多个独立的 $\vec{y}_r \in Y$ 与凸包点集CH(X)之间的线性判别函数LDFs,包括以下子步骤:

[0178] (4.3.1)输入点集 $X \subseteq R^n$, $\vec{y}_r \subseteq R^n$ 和精度参数 ε ,取任意 $\vec{x}_{new}^* \in X$;

[0179] (4.3.2) 初始化 \vec{y}_r 到凸包点集 $CH(X)$ 中距离最近的 $\vec{x}^* = \vec{x}_{new}^*$;

[0180] (4.3.3) 在点集 X 中找到满足如下条件的 \vec{x}_s :

$$[0181] \quad m(\vec{x}_t) = \frac{\langle \vec{x}_t - \vec{y}_r, \vec{x}^* - \vec{y}_r \rangle}{\|\vec{x}^* - \vec{y}_r\|}$$

$$[0182] \quad s = \arg \min_{t \in |X|} \{m(\vec{x}_t)\}$$

[0183] 其中 $m(\vec{x}_t)$ 并不总是正的,它实际上是在特征空间中向量 $\{\vec{x}_t - \vec{y}_r\}$ 在向量 $\{\vec{x}^* - \vec{y}_r\}$ 上的投影,见图1(a)和(b)。

[0184] (4.3.4) 通过下式更新向量 \vec{x}_{new}^* :

$$[0185] \quad q = \min \left\{ 1, \frac{\langle \vec{x}^* - \vec{y}_r, \vec{x}^* - \vec{x}_s \rangle}{\|\vec{x}^* - \vec{x}_s\|} \right\}$$

$$[0186] \quad \vec{x}_{new}^* = \begin{cases} \vec{x}^* + q(\vec{x}_s - \vec{x}^*), & 0 < q < 1 \\ \vec{x}_s, & q = 1 \end{cases}$$

[0187] 其中, q 的两种情况如图1(c)和(d)所示;

[0188] (4.3.5) 重复迭代步骤(4.3.2)~(4.3.4),在下一次迭代中找到距离最近的 \vec{x}_s 更新为 \vec{x}_{new}^* ,直到满足如下条件:

$$[0189] \quad \|\vec{x}^* - \vec{y}_r\| - \|\vec{x}_{new}^* - \vec{y}_r\| < \varepsilon$$

[0190] 这里存在一种特殊情况,即实际中超平面并不一定存在,图2(a)给出了 \vec{y}_r 在凸包点集 $CH\{\vec{x}_z, \vec{x}^*, \vec{x}_s\}$ 中的一个例子,此时超平面并不能把 \vec{y}_r 和凸包点集 $CH(X)$ 分开;这种情况意味着 \vec{y}_r 和凸包点集 $CH(X)$ 是线性不可分的;为了解决这个问题,本发明添加了一个步骤(4.3.6)。

[0191] (4.3.6) 计算线性判别函数 $f_r(\vec{x}) = \langle \vec{w}_r, \vec{x} \rangle + b_r, \vec{y}_r \in Y$,得到分段线性边界 $f_r(\vec{x}) = 0$;

其中, $\vec{w}_r = \vec{x}_{new}^* - \vec{y}_r, b_r = \frac{\|\vec{y}_r\|^2 - \|\vec{x}_{new}^*\|^2}{2}$; 如果存在 $\vec{x}_z \in X$ 使得 $f_r(\vec{x}_z) = \langle \vec{w}_r, \vec{x}_z \rangle + b_r \leq 0$, 则 $f_r(\vec{x}) = NULL$; 否则输出 $f_r(\vec{x})$ 。

[0192] (4.4) 根据下式找到距离凸包点集 $CH(X)$ 最近的 \vec{y}_p :

$$[0193] \quad p = \arg \min_r \|\vec{w}_r\|, 1 \leq r \leq |Y|$$

[0194] (4.5) 从步骤(4.3)得到的线性判别函数候选集CLDFs中选择步骤(4.4)中 \vec{y}_p 对应的 $f_p(\vec{x})$ 放到线性判别函数集LDFs中;

[0195] (4.6) 去掉点集 Y 中满足 $f_p(\vec{y}_r) < 0$ 的 \vec{y}_r 。

[0196] (4.7) 重复迭代步骤(4.4)~(4.6)直到点集 $Y = \emptyset$,输出线性判别函数集LDFs,得到分段线性边界。例如图2(b)是一组由线性判别函数集 $LDFs = \{f_1(\vec{x}), f_2(\vec{x}), f_3(\vec{x})\}$ 组成的分段线性边界。

[0197] (5) 69个字符样本类都到达底部的叶节点,分配叶节点对应的字符样本类的类别标签,最终得到二叉划分树。如图3(c)和(d)所示,CPPLC1和CPPLC2分别对Class₁和Class₃、

Class₂和Class₄计算分段线性边界,当只剩单个字符样本类时,将其标记为叶节点,最终生成了四个叶节点,每个叶节点代表一个字符样本类的分类决策。

[0198] 完成以上步骤,训练好的基于凸多面体分段线性分类的二叉划分树车牌识别方法可以用于执行车牌图像识别任务。

[0199] 步骤7,根据步骤2~5的操作对将步骤1中车牌数据集CCPD设定的5000张测试图像进行图像预处理、车牌定位与校正、车牌字符分割、特征提取;

[0200] 步骤8,将步骤7处理后的车牌字符图像的特征向量集合输入步骤6构建的HBPT-CPPLC分类器,输出每张车牌图像的识别结果;

[0201] 对待分类样本,从二叉划分树的根节点开始,计算待分类样本与根节点的子节点的欧氏空间距离,判断距离更近的子节点是内部节点或叶节点,如是叶节点,则给待分类样本分配对应的类的标签;如是内部节点,则计算待分类样本与该内部节点的子节点的欧氏空间距离,判断距离更近的子节点是内部节点或叶节点,重复执行计算距离和判断子节点的步骤直到到达叶节点。

[0202] 通过以下评价指标进行算法的分类效果评测:准确率A(accuracy)、精确率P(precision)、召回率R(recall)、F₁分数(F₁score)、训练(测试)时间,以及对抗样本的鲁棒性。本发明在分类效率和计算时间上都优于现有的分段线性分类算法,验证了其竞争力和优越性。本发明报告50次实验的平均结果,并设定算法迭代终止的精度参数为 $\epsilon=10^{-3}$ 。

[0203] 在车牌识别的字符图像分类任务中,本发明将HBPT-CPPLC与三种多类分段线性分类器:LIN-SVM,RBF-SVM和KDT进行了比较。本发明同时还评估了HBPT-CPPLC在测试集上的对抗噪声鲁棒性。本发明使用Fast Gradient Sign方法(FGSM)在测试集中生成对抗性的字符图像样本。

[0204] 在K类字符图像分类的问题中,本发明实施例定义了新的性能度量评价指标,其测量值以平均形式来计算:

$$[0205] \quad A = \frac{\sum_{j=1}^K TP_j}{N}$$

$$[0206] \quad P = \frac{\sum_{j=1}^K \frac{TP_j}{TP_j + FP_j}}{K}$$

$$[0207] \quad R = \frac{\sum_{j=1}^K \frac{TP_j}{TP_j + FN_j}}{K}$$

$$[0208] \quad F_1 = \frac{2 \times P \times R}{P + R} \quad (\text{式 } 5)$$

[0209] 其中TP、TN、FP和FN分别表示给定字符图像样本类的真阳性、真阴性、假阳性和假阴性,N表示所有字符图像类样本的数量N_j之和。

[0210] 在对抗图像样本测试的实验中,为了使结果更易于解释,本发明实施例计算了字符图像样本信噪比(SNR)。对于输入的字符图像x和扰动噪声 δ_x ,SNR定义为

$$[0211] \quad SNR(x, \delta_x) = 20 \log_{10} \frac{\|x\|_2}{\|\delta_x\|_2} \quad (\text{式 } 6)$$

[0212] 字符图像分类任务上,HBPT-CPPLC与LIN-SVM,RBF-SVM和KDT在准确率(A)、精确率(P)、recall(R)和 F_1 分数(F_1 score)方面的对比情况如表1所示。表2列出了各算法以秒为单位的训练时间和以毫秒为单位的测试时间对比情况,其中括号里的是测试时间。表3总结了不同信噪比下,HBPT-CPPLC与LIN-SVM,RBF-SVM和KDT的对抗样本鲁棒性的对比情况。

[0213] 表1:HBPT-CPPLC与LIN-SVM,RBF-SVM和KDT的评价指标(%)对比情况

[0214]	LIN-SVM	<i>A</i>	92.37
		<i>P</i>	92.11
		<i>R</i>	91.14
		F_1 score	90.19
	RBF-SVM	<i>A</i>	95.24
		<i>P</i>	94.22
		<i>R</i>	93.19
		F_1 score	94.46
	KDT	<i>A</i>	86.28
		<i>P</i>	85.59
		<i>R</i>	84.29
		F_1 score	87.13
	HBPT-CPPLC	<i>A</i>	99.01
		<i>P</i>	98.38
		<i>R</i>	97.14
		F_1 score	97.98

[0215] 表2:HBPT-CPPLC与LIN-SVM,RBF-SVM和KDT在训练时间/s(测试时间/ms)的对比情况

[0216]	LIN-SVM	23.16 (118.33)
	RBF-SVM	77.98 (2.99×10^4)
[0217]	KDT	15.26 (1.97×10^3)
	HBPT-CPPLC	7.99 (125.68)

[0218] 表3:HBPT-CPPLC在不同信噪比下与LIN-SVM,RBF-SVM和KDT的分类准确率(%)对比情况

[0219]

方法	信噪比SNR	准确率
LIN-SVM	无噪声	92.37
	50	90.13
	45	86.62
	40	83.24
	35	77.25
RBF-SVM	无噪声	95.24
	50	92.31
	45	88.35
	40	84.99
	35	81.97
KDT	无噪声	86.28
	50	82.39
	45	79.77
	40	76.24
	35	69.98
HBPT-CPPLC	无噪声	99.01
	50	96.24
	45	93.27
	40	91.88
	35	89.39

[0220] 根据表1~3的实验结果,本发明实施例得到下面的结论:

[0221] (1) 与LIN-SVM相比,HBPT-CPPLC的A,P,R, F_1 score指标值都更高。与KDT相比,HBPT-CPPLC的精确度都高于KDT。KDT具有类似于HBPT-CPPLC的树结构,但KDT仅通过中心点来计算每个超平面,这可能会产生大量的累积误差。在所有数据集上,HBPT-CPPLC的性能都优于RBF-SVM。

[0222] (2) 在训练时间上,HBPT-CPPLC比LIN-SVM,RBF-SVM和KDT需要的时间更短,因为对于K类字符图像分类问题,HBPT-CPPLC算法最多只需要训练K-1个CPPLC二分类分类器。在检测时间上,HBPT-CPPLC也明显优于LIN-SVM,RBF-SVM和KDT。

[0223] (3) 表3说明HBPT-CPPLC在对抗性样本的鲁棒性方面明显优于LIN-SVM,RBF-SVM和KDT。这是因为超平面产生了一个训练前期的快速划分,大规模的问题被分解成多个子问题。然后利用CPPLC进行子问题的分类识别,保证了泛化能力。

[0224] 本发明构造了一种基于凸多面体分段线性分类的二叉划分树车牌识别方法(HBPT-CPPLC)。本发明首先提出了一种凸多面体分段线性分类器(CPPLC)算法。如果两个点

集是凸多面体可分的,则以组合分类器的形式学习一系列超平面,用凸多面体包围内部的点集,排除外部的点集。作为该方法的推广,本发明采用分治策略,将多分类问题分解成多个二分类问题,提出了一种基于凸多面体分段线性分类的二叉划分树车牌识别方法(HBPT-CPPLC)。本发明也验证了模型在对抗性样本下具有很强的鲁棒性。在标准车牌识别图像数据集上的实验表明,HBPT-CPPLC在精度和速度方面都优于现有的分段线性分类器。

[0225] 需要注意的是,公布实施例的目的在于帮助进一步理解本发明,但是本领域的技术人员可以理解:在不脱离本发明及所附权利要求的精神和范围内,各种替换和修改都是可能的。因此,本发明不应局限于实施例所公开的内容,本发明要求保护的范围以权利要求书界定的范围为准。

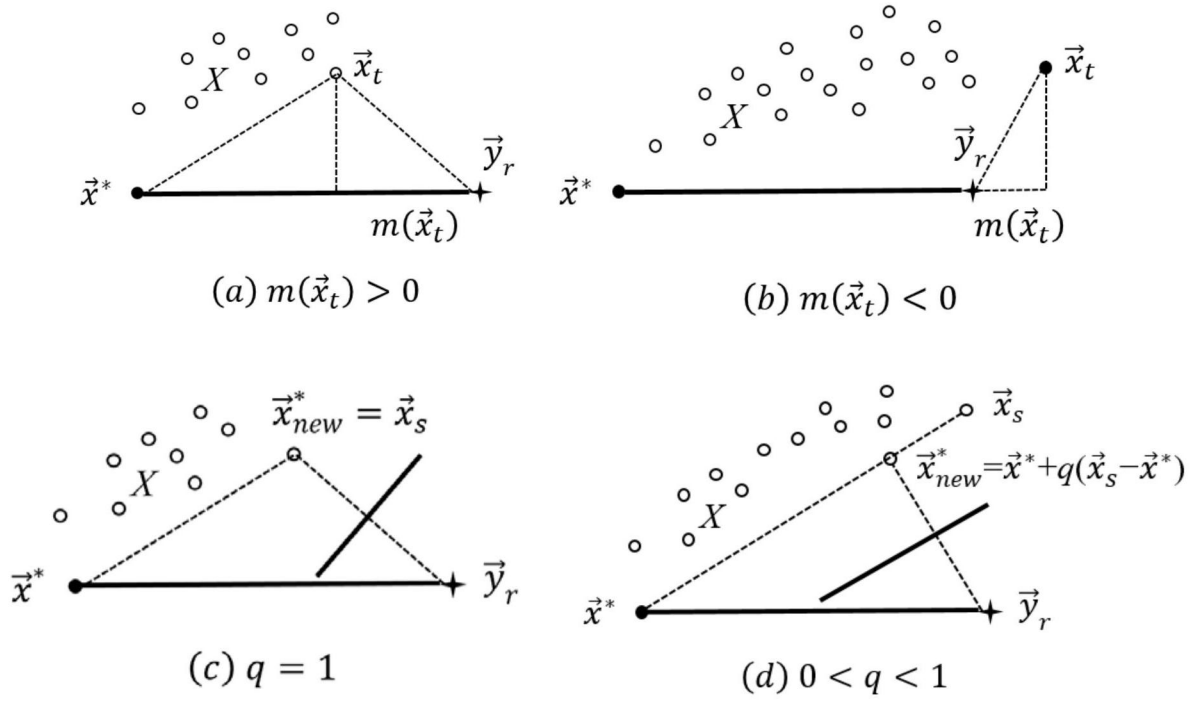


图1

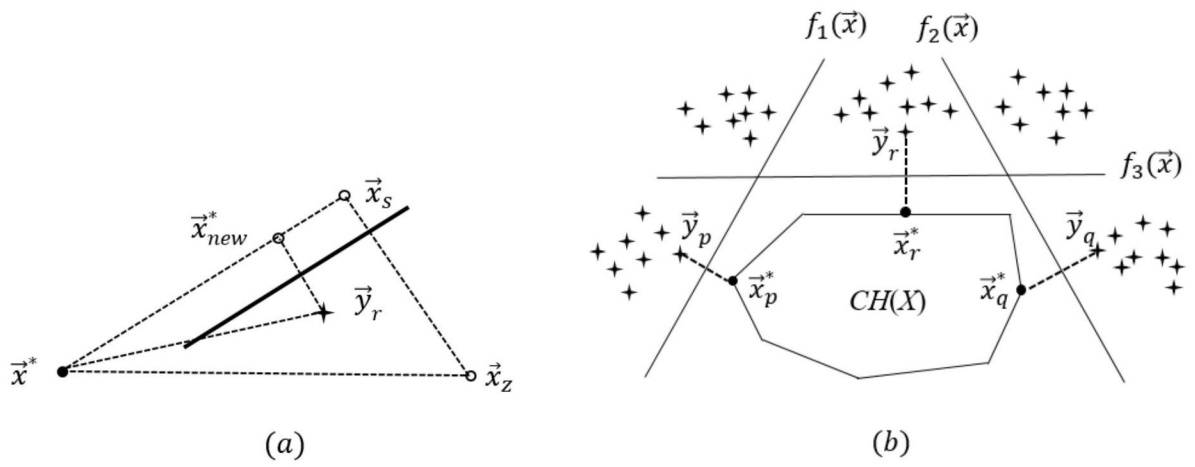
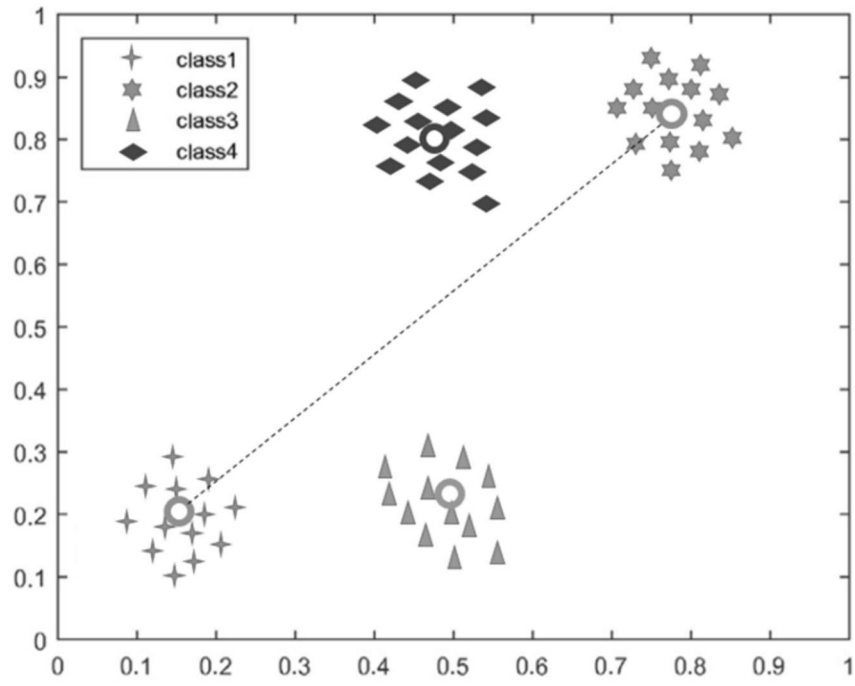
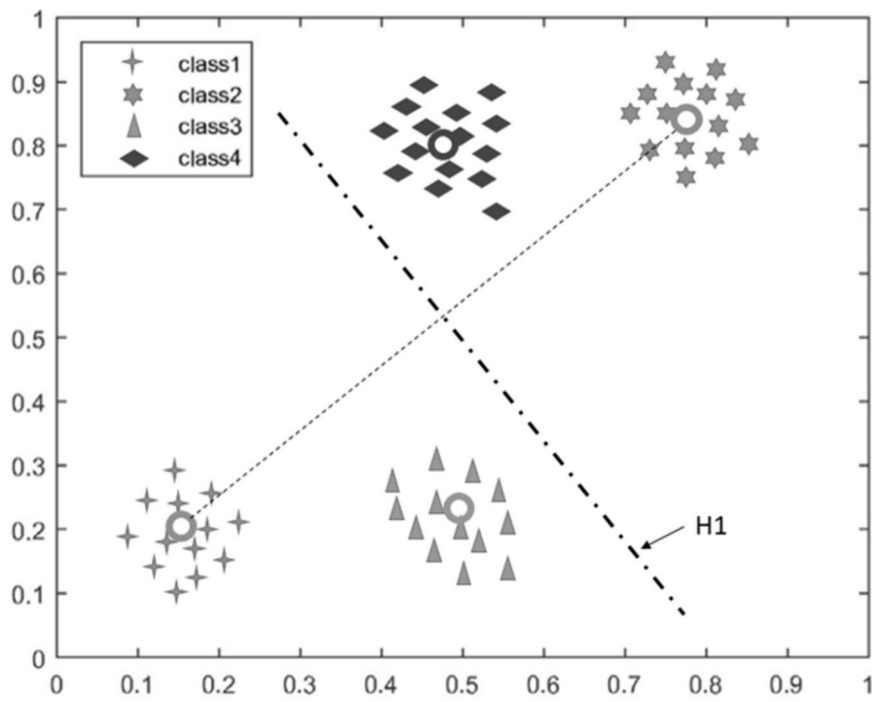


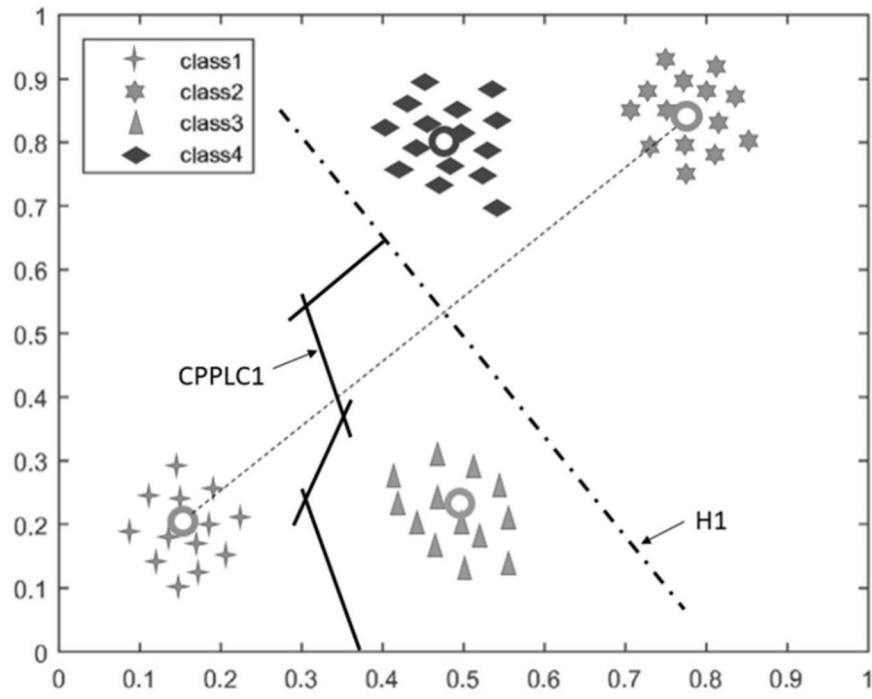
图2



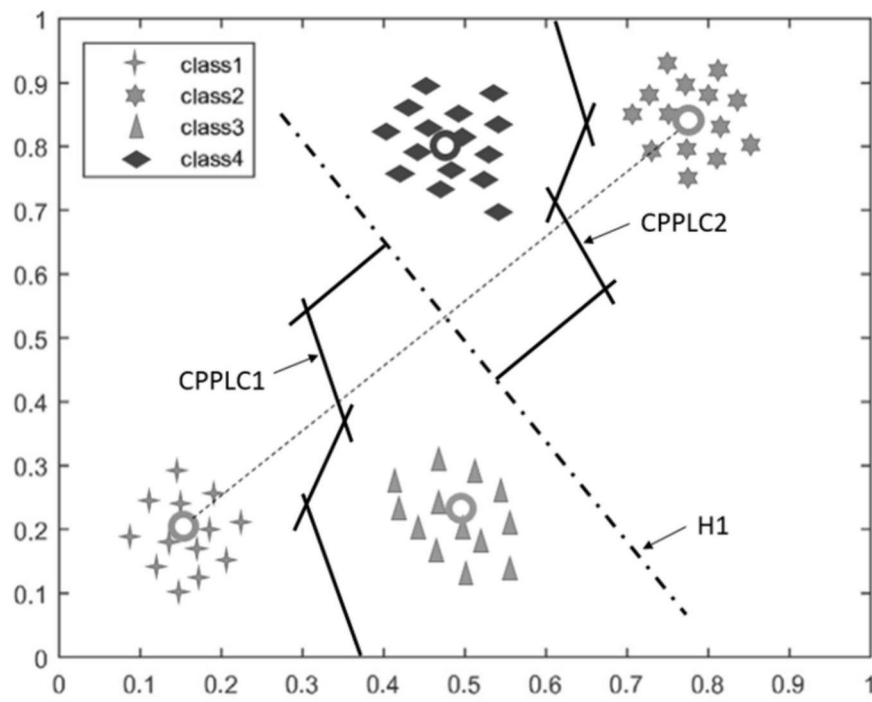
(a)



(b)



(c)



(d)

图3

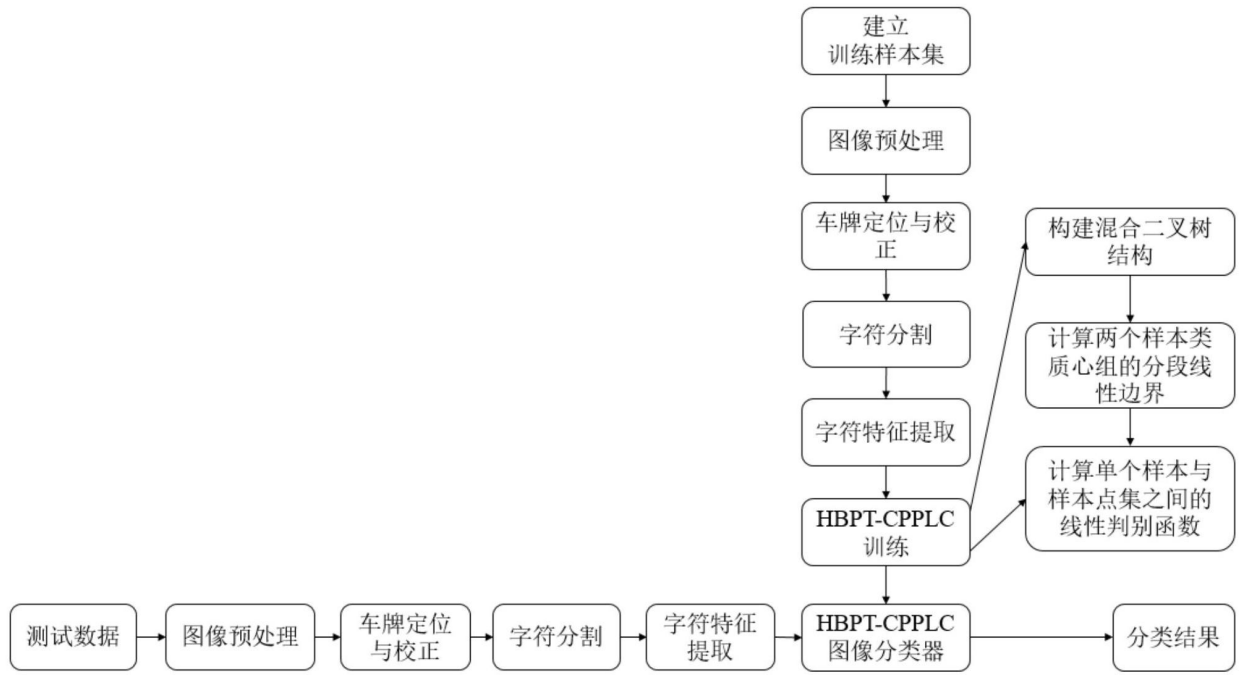


图4

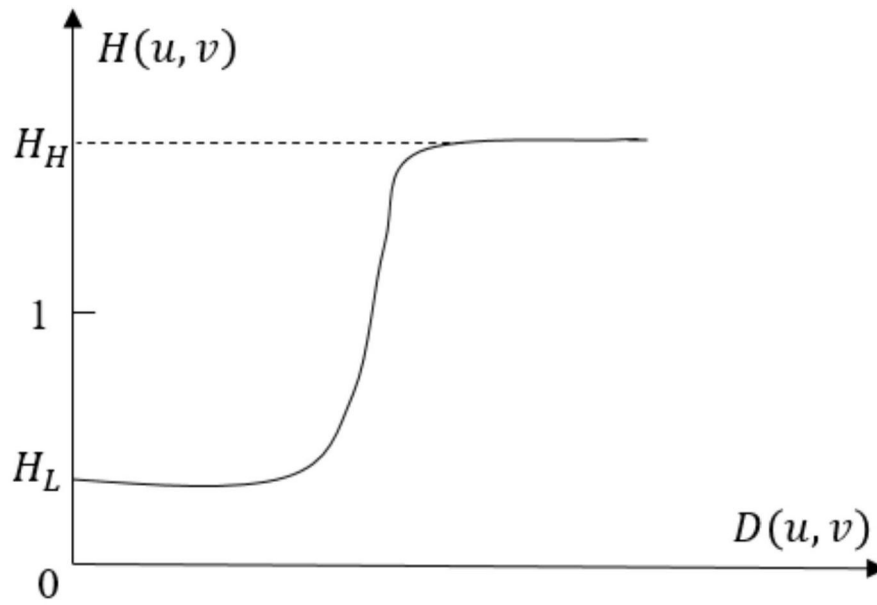


图5