

文章编号:1671-6833(2019)05-0091-06

一种前景提取的行人模式识别检测算法

杨忠明¹, 李子龙², 胡音文², 黄 翰², 蔡昭权³

(1.广东科学技术职业学院 计算机工程技术学院,广东 珠海 519090; 2.华南理工大学 软件学院,广东 广州 510006; 3.惠州学院,广东 惠州 516007)

摘 要:通过深入研究高斯混合模型、HOG+SVM 分类器和 Haar+AdaBoost 分类器的算法原理,提出了一种基于前景提取和模式识别的行人检测算法,首先使用高斯混合模型进行背景建模,通过前景提取的方法提取出运动物体,对运动物体进行行人检测,最后对判断为行人的对象进行人脸检测,分类区分出蒙面的可疑行人,解决基于背景建模方法的误判问题和基于统计学习方法的效率问题.实验结果表明,相对于单独使用模式识别算法,新算法显著降低了漏检率,且在运行速度和检测率方面表现良好.

关键词:背景建模;行人检测;人脸检测;AdaBoost;模式识别

中图分类号:TP317.4

文献标志码:A

doi:10.13705/j.issn.1671-6833.2019.02.017

0 引言

目前平安城市的视频监控遍布城市核心区域,利用监控视频完成行人检测有利于人流统计、安全防范等,检测蒙面的可疑行人是视频监控的核心问题.目前常用的行人检测方法可分为两类:基于背景建模的方法和基于统计学习的方法^[1].前者易受干扰,误判率较高^[2];后者受训练样本影响较大,且只能对目标进行多尺度反复扫描,处理速度较慢^[3].另有学者提出基于 ELM 与 Mean Shift 的算法解决行人遮挡带来的检测异常问题^[4].目前的行人检测方法因判别因素单一,且缺乏多因素之间的关联性研究,故难以取得令人满意的检测效果,亟需一种高效的复合型算法提高行人检测的速度和准确率^[5].

针对上述行人检测方法的缺点与不足,笔者提出一种基于前景提取和模式识别的行人检测算法,以解决基于背景建模方法的误判问题和基于统计学习方法的效率问题.

1 关键算法原理

1.1 混合高斯模型

混合高斯模型的基本思想是使用 K 个高斯分布对视频中图像的每一像素进行建模,不只是

选择一个特定的分布^[6].GMM 将当前像素点判定为背景模型的概率定义为:

$$p(X_t) = \sum_{k=1}^K \omega_{k,t} \times \eta(X_t, \mu_{k,t}, \sigma_{k,t}^2), \quad (1)$$

式中: K 为高斯模型的数量, K 取值越大,背景模型抗干扰能力越强,对应的计算量也会越大,取值一般为 $3 \sim 5$ ^[7]; $\omega_{k,t}$ 是 t 时刻第 k 个高斯分布的权值; $\mu_{k,t}$ 为与之相对应的均值; $\sigma_{k,t}^2$ 为与之相对应的方差; $\eta(X_t, \mu_{k,t}, \sigma_{k,t}^2)$ 为 t 时刻第 k 个高斯分布的概率密度函数,可表示为:

$$\eta(X_t, \mu_{k,t}, \sigma_{k,t}^2) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(X_t - \mu_t)^T \Sigma^{-1}(X_t - \mu_t)}. \quad (2)$$

每读取一个新的视频帧,都需要更新模型参数,新一帧的每一像素都需要匹配所有的 K 个高斯分布,并决定是否更新.更新公式为:

$$\omega_{k,t} = (1 - \alpha) \omega_{k,t-1} + \alpha(M_{k,t}), \quad (3)$$

$$\mu_{k,t} = (1 - \rho) \mu_{k,t-1} + \rho X_t, \quad (4)$$

$$\sigma_{k,t}^2 = (1 - \rho) \sigma_{k,t-1}^2 + \rho(X_t - \mu_{k,t})^T(X_t - \mu_{k,t}), \quad (5)$$

$$\rho = \alpha \eta(X_t, \mu_{k,t}, \sum_{k,t}), \quad (6)$$

其中, α 为权值更新率; ρ 为参数更新率.

若新的像素点与某个分布匹配,则 $M_{k,t}$ 取 1; 反之, $M_{k,t}$ 取 0,若新的像素点不与任何分布匹配,则用新的像素点建立新的高斯分布,替换掉拥有

收稿日期:2018-11-19;修订日期:2018-12-21

基金项目:国家自然科学基金资助项目(61772225);广州市科技计划资助项目(201804010276、2018070100047)

作者简介:杨忠明(1980—),男,广东科学技术职业学院副教授,主要研究方向为智能优化算法及应用,E-mail:

yzm8008@126.com.

最低优先级的高斯分布^[8].更新高斯模型参数后,以 $\omega_{k,t}/\sigma_{k,t}$ 作为评测标准,将 K 个高斯分布从高分到低的顺序进行排列.序列顶部的高斯分布最可能描述背景,故可取前 B 个高斯分布来构成初始的背景模型,为之后的前景提取做准备,

$$B = \arg \min_k \left(\sum_{k=1}^K \omega_{k,t} \right) > T, \quad (7)$$

式中: T 为背景阈值.

将每帧的每个像素与背景模型匹配,如果与其中一个高斯分布相匹配,则将此像素点归为背景点,如果不与任何一个高斯分布相匹配,则将此像素点归为前景点.

1.2 基于 HOG 特征的 SVM 分类器

HOG 算子是一种用于目标检测的特征描述符,通过计算图像梯度或边缘方向直方图来构成特征^[9].图像中像素点 (x, y) 的梯度为:

$$G_x(x, y) = H(x+1, y) - H(x-1, y), \quad (8)$$

$$G_y(x, y) = H(x, y+1) - H(x, y-1), \quad (9)$$

其中, $G_x(x, y)$ 为 (x, y) 处的水平方向梯度; $G_y(x, y)$ 为 (x, y) 处的垂直方向梯度幅值; $H(x, y)$ 为 (x, y) 处的像素值.像素点和梯度方向分别为:

$$G(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2}, \quad (10)$$

$$\alpha(x, y) = \tan^{-1}(G_y(x, y)/G_x(x, y)). \quad (11)$$

计算 Hog 特征的步骤为:首先将图像划分成多个小区域,记为细胞单元,把梯度方向平均分成 9 个区间(bin).计算单元中各个像素点的梯度或边缘的方向直方图,求取一个 9 维的特征向量.将相邻的 4 个单元组成一个块(block),把块内的特征向量联合起来得到 36 维的特征向量.用块对目标图像进行扫描,取一个细胞单元作为步长,最后将所有块的特征串连起来得到人体的特征^[10].

支持向量机 SVM^[11]解决了在线性可分情况下的最优分类面问题,最优分类不仅正确分开两类且使分类间隔达到最大,在高维情况下,最优分类则是要求出分类面.我们可以用一个非线性映射,求出用于数据分类的超平面,如图 1 所示.

1.3 AdaBoost 分类器

AdaBoost 算法^[18]将多个弱分类器迭代成一个强分类器,然后将多个强分类器进行级联,构成一个最优的分类器,用来对图像进行检测^[12].常用的特征向量为 Viola^[17]提出的类 Haar 特征^[13].类 Haar 特征是一种“矩形”对特征,描述图像模式相邻区域的特征差异,如图 2 所示.

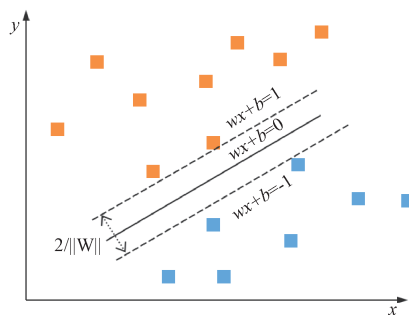


图 1 超平面

Fig.1 Hyperplane

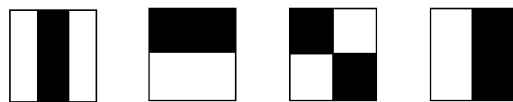


图 2 常用的 Haar 特征

Fig.2 Commonly used Haar features

取图像中与 Haar 特征大小相同的子区域,将 Haar 特征中白色区域对应的图像像素强度值之和减去黑色区域对应的图像像素强度值之和,便能求得特征值. AdaBoost 算法的训练过程则是挑选最优弱分类器,并赋予权重的过程.

步骤 1. 首先计算出样本 Haar 特征位置参数.

步骤 2. 根据 Haar 特征的位置参数求出各弱分类器.

步骤 3. 按 AdaBoost 算法的训练流程迭代并挑选出最优弱分类器.

步骤 4. 记录当前最优弱分类器及其迭代权重.

步骤 5. 提高上一轮误判的样本权重.

步骤 6. 将新样本与上一轮误判的样本进行新一轮训练.

步骤 7. 重复步骤 4 至步骤 6, 循环 N 次获得 N 个最优弱分类器.

步骤 8. 联合 N 个最优弱分类器获得强分类器^[14].

2 算法设计及实现

首先使用混合高斯模型进行背景建模,提取出运动物体,而后对前景图像进行行人检测,判断前景图像中是否含有行人以及行人的位置,再对判断出的行人进行人脸检测,算法整体结构如图 3 所示.

2.1 基于混合高斯模型的前景目标发现算法

首先利用混合高斯模型建立视频背景模型,从而实现视频前景目标的发现.算法 2-1 给出了基于混合高斯模型的前景目标发现算法的伪代码.

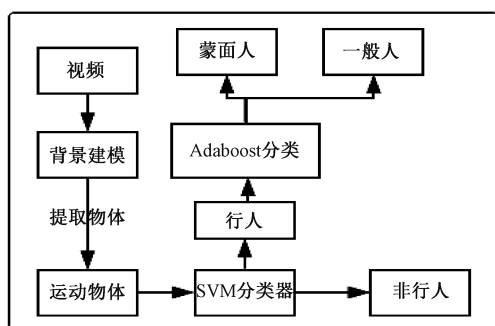


图3 算法结构

Fig.3 The algorithm structure

基于混合高斯模型的前景目标发现算法代码如下:

输入: 高斯模型的数量 K .

输出: 前景目标外接矩形集合 S_{final}

- 1: **while** 视频未读取完 **do**
- 2: 从视频中读取一帧图像
- 3: $I \leftarrow$ 图像标准化
- 4: **if** 为第一帧图像 **then**
- 5: $GMM \leftarrow$ 初始化 K 个高斯模型
- 6: $I_b \leftarrow$ 初始化背景
- 7: **else**
- 8: $GMM \leftarrow$ 更新高斯模型
- 9: $I_b \leftarrow$ 更新背景
- 10: $J \leftarrow I - I_b$
- 11: $BW_s \leftarrow$ 对 J 进行二值化
- 12: $BW \leftarrow$ 缩放 BW_s 至视频大小
- 13: $I'_t \leftarrow$ 对 BW 进行形态学处理
- 14: $I''_t \leftarrow$ 计算 I'_t 的轮廓
- 15: $S_{rect} \leftarrow$ 计算 I''_t 中各个目标的外界矩形
- 16: $S_{final} \leftarrow$ 精简轮廓
- 17: **end if**
- 18: **end while**

其中, 算法第3行的图像标准化需要将首帧灰度化, 并将其长宽放缩到原图的1/3. 灰度化和放缩的目的是为了降低时间消耗. 算法第12行将高斯模型提取到的前景图像的长宽恢复到原始大小. 这里恢复的原因是, 若输入的图像过小, 会大大提高后面行人检测和人脸检测的漏检率.

算法第13行形态学处理的输入为前景图像 BW , 输出为已填补好缝隙的前景图 I'_t . 包括两个步骤: ①对提取出的前景图像进行开运算, 以消除前景图像的噪声, 即那些离散、独立的点; ②填补前景缝隙, 对提取出的前景图像进行闭运算, 对图像分开的、离散的区域进行连接, 填补图像中狭小

的缝隙.

算法第15行生成前景轮廓, 输入为前景图像 I'_t , 输出为提取到的矩形框集合 S_{rect} . 因为前景提取的效果并不理想, 即使经过形态学处理, 提取出来的图像依然过于零碎, 无法构成一幅较圆满的运动物体前景图. 为了更准确地找出前景图像即运动物体的矩形框, 首先找出所有前景点的轮廓, 然后找出这些轮廓的凸包及其外接矩形; 为防止边缘丢失, 其长宽分别增加 O_w 像素和 O_h 像素, 并将其加入到集合 S_{rect} .

算法第15行对应的精简前景目标输入为前景矩形框集合 S_{rect} , 输出为已被精简的集合 S_{final} . 遍历 S_{rect} , 每次取出单一 $rect_1$ 并与 S_{rect} 中剩余的 $rect_2$ 进行比对, 若 $|rect_1 \cdot x - rect_2 \cdot x| < t$, 则求 $rect_1$ 与 $rect_2$ 的外接矩形, 并将其添加到 S_{rect} , 删除 $rect_1$ 、 $rect_2$. 否则, 将 $rect_2$ 加入 S_{final} .

2.2 基于方向梯度直方图与支持向量机的行人检测

步骤1. 初始化行人检测分类器. 创建 SVM 分类器, 分类器类型为 *epsilon* 类支持向量回归机, 使用核函数为线性核函数.

步骤2. 将2.1节中得到的 S_{rect} 设置为当前帧 I_{curr} 的感兴趣区域, 并裁出, 得到 I_{dest} .

步骤3. 将感兴趣区域提取出来保存至新的图像中, 并放缩至固定大小 64×128 . 这里将图像放缩至固定大小不仅是为了缩小输入图像以提高算法运行速度, 而且也防止输入图像过小以至小于检测窗口的最小限制.

步骤4. 使用直方图均衡化增强 I_{dest} 的亮度与对比度, 并对 I_{dest} 进行多维度检测. 将结果保存至集合 S_{person_result} 中.

步骤5. 遍历集合 S_{person_result} , 处理得到的行人矩形框, 将不嵌套的矩形框加入新集合 S_{person_final} 中, 而嵌套的矩形框则取大的矩形框加入 S_{person_final} .

步骤6. 求出矩形框在原图像 I_{curr} 中的坐标, 绘制行人检测结果.

步骤7. 重置 I_{curr} 的感兴趣局域, 重复步骤2~7, 直到 S_{rect} 遍历完毕.

2.3 基于 Haar 特征与 AdaBoost 的人脸检测

步骤1. 初始化人脸检测分类器.

步骤2. 将2.2节中得到的 S_{person_final} 设置为当前帧 I_{curr} 的感兴趣区域, 并裁出.

步骤3. 使用直方图均衡化增强图片亮度与对比度.

步骤 4. 进行人脸检测, 得到结果序列.

步骤 5. 遍历结果序列, 求出检测出的矩形框在当前帧 I_{curr} 中的位置.

步骤 6. 在 I_{curr} 中绘出结果矩形框.

3 实验结果分析

3.1 实验设置

针对行人检测任务, 本文使用的数据集为 Duke 大学的 MTMC 数据集^[19], 该数据集包含 8 个摄像头拍摄的超过 160 GB 数据, 时长约 85 min. 本实验所采用的计算环境为 Ubuntu16.04 + E5 2630v4 (10 核, 20 线程, 2.2 GHz). 实验通过检测耗时、检测框数、错检数、漏检率 4 个性能指标定量描述算法的性能. 实验中提出方法所涉及的参数设置如下: 混合高斯模型中高斯模型的数量一般取 3~5, K 取值越大, 背景模型抗干扰能力越强, 但对应的计算量也会越大^[7]. 为了取得背景模型的精度与速度之间的平衡, 笔者取 K 值为 3. 背景阈值 $T=26$, 最小轮廓面积为 5 000 像素, 长宽扩增分别为 $O_w=60$ 像素和 $O_h=200$ 像素. 在人脸检测应用实验中考虑到 Adaboost 检出率及计算代价随着弱分类器数量增加而增加, 弱分类器数量达到一定数量时 (大于 500 个弱分类器), 其检出率提升不明显^[20], 本实验中弱分类器数量取 500.

3.2 实验结果

实验结果如表 1 所示. 笔者算法在运行速度和检测效率方面表现良好, 可在实际程序中应用. 通过实验可以看出, 相对于单独使用模式识别算法大大地降低了漏检率, 先提取出运动物体再检测, 也会大大地降低错检数. 这里的总框数是指该模式识别算法所检测到的所有矩形框的数量. 下面针对表 1 的实验结果从准确率和计算耗时进行分析.

表 1 实验结果

Tab.1 Experimental results

	HOG+ SVM	Haar+ AdaBoost	笔者 算法
行人检测时间/(ms·帧 ⁻¹)	853.54	—	77.66
行人检测结果框数/(个·帧 ⁻¹)	2.34	—	0.47
人脸检测时间/(ms·帧 ⁻¹)	—	2 121.86	129.99
人脸检测结果总框数	—	1.36	0.08
错检数/(个·帧 ⁻¹)	4.90	—	0.93
漏检率/%	80.06	—	77.49

3.3 准确率及分析

经过背景建模、前景提取之后得到的前景矩

形框会出现过小, 产生一定的误差. 但由于本算法适当扩大了前景矩形框, 实际检测效果并未受到较大影响. 通过背景建模与前景提取, 可以有效减小行人的搜索范围, 缩减行人检测器的滑窗数量, 从而降低行人检测的误检数和漏检率.

由于人脸是跟随行人一同运动的实体. 与行人检测相似, 由于降低了搜索范围, 人脸检测的滑窗数量大大降低, 同样降低了人脸检测的误检数和漏检率.

3.4 计算耗时及分析

对码率为 26 295 Kbps, 分辨率为 1 920 × 1 080 的 MTS 格式视频文件进行处理, 其中背景建模部分每一帧平均耗时约为 115.47 ms, 行人检测部分每一帧平均耗时约为 77.66 ms, 而人脸检测部分每一帧平均耗时约为 129.99 ms.

根据过去图像处理领域研究经验可知, 若将平常的彩色图像转换为灰度图进行处理, 将能很大地提高背景建模的速度; 将输入的图像缩小, 则其运行时间也会按比例减少. 本实验将输入图像转换成灰度图并且缩小到原来的 1/3 来进行背景建模, 取得了不错的效果.

同时输入的图像对函数的运行速度影响很大, 所以本实验在行人检测之前使用背景建模再进行前景提取的方法提取出运动物体, 力求将检测区域缩到最小, 提高运行速度, 类似的方法可见文献[16].

另外一个影响较大的参数为 padding, 是用于对要检测的图像进行边界的扩充, 如果设置为 (0,0), 那么检测窗口的中心就不是从图像的 (0,0) 坐标开始扫描的. 这样虽然能提高运行速度, 却会降低检测正确率, 所以这是一个需要权衡的参数, 经对比, 本实验使用参数 (32,32), 这样较好地平衡运行速度和检测正确率.

对于人脸检测部分的运行速度, 受到输入图像的质量和大小影响, 经过背景建模和前景提取之后, 将识别区域限制在一个较小的范围, 依然耗时大约在 130 ms. 所以只能在行人检测检测到结果的前提下才启动人脸检测, 该方法人脸检测的漏检率会提高, 不过能有效地提升总体的运行速度.

4 结论

研究了一种基于前景提取和模式识别的行人检测算法, 通过对混合高斯模型、HOG+SVM 分类器、Harr+AdaBoost 等基础算法的综合应用, 解决

前景提取的误检率问题和模式识别的速度问题,并且该思路还可以完成蒙面人这一类特殊对象的初步甄别,并通过实验验证了新型混合算法的可行性。

在部分特殊场景下,会存在混合高斯模型提取出的矩形框比行人对象要小的情况,下一步的工作将针对混合高斯模型提取出的矩形框不准确的问题,在矩形框附近进行多次采样的方法进行研究。

参考文献:

- [1] 杨涛,李静,潘泉,等.基于场景模型与统计学习的鲁棒行人检测算法[J].自动化学报,2010,36(4):499-508.
- [2] 张江钰.依据移动终端和机器视觉的机器人路径跟随控制方法研究[D].武汉:湖北工业大学电气与电子工程学院,2016.
- [3] 李蹊.深度学习在行人检测中的研究与应用[D].北京:北京邮电大学信息与通信工程学院,2017.
- [4] 郝向东,毛晓波,梁静.ELM与Mean Shift相结合的抗遮挡目标跟踪算法[J].郑州大学学报(工学版),2016,37(1):1-5.
- [5] SHIPOVA K G, SAVCHENKO A V. Video-Based pedestrian detection on mobile phones with the cascade classifiers[C]// International Conference on Network Analysis. Springer, Cham, 2014:209-216.
- [6] RUMAKSARI A N, SUMPENO S, WIBAWA A D. Background subtraction using spatial mixture of Gaussian model with dynamic shadow filtering[C]// International Seminar on Intelligent Technology and ITS Applications. IEEE, 2017:296-301.
- [7] 范文超,李晓宇,魏凯,等.基于改进的高斯混合模型的运动目标检测[J].计算机科学,2015,42(5):286-288+319.
- [8] SHAHBAZ A, KURNIANGGORO L, JO K H. A comparative study of foreground detection using Gaussian mixture models-novice to novel[C]// International Conference on Control, Automation and Systems. IEEE, 2017:1333-1336.
- [9] 符林清.采用HOG算子的行人检测实时处理器的性能优化[D].深圳:深圳大学信息工程学院,2016.
- [10] 徐琳,张明.基于特征融合的行人检测技术[J].计算机系统应用,2015,24(10):238-242.
- [11] ANSARI M A, DIXIT M, Kurchaniya D, et al. An effective approach to an image retrieval using sVM classifier[J]. International journal of computerences & Engineering, 2018, 5(6):1410-1417.
- [12] WANG J, GAO L, ZHANG H, et al. Adaboost with SVM-Based classifier for the classification of brain motor imagery tasks [M]// Universal Access in Human-Computer Interaction. Users Diversity. Springer Berlin Heidelberg, 2011:629-634.
- [13] OUALLA M, Sadiq A, Mbarki S. A survey of Haar-Like feature representation[C]// International Conference on Multimedia Computing and Systems. IEEE, 2014:1101-1106.
- [14] 李文娜.基于Haar特征级联强分类器和肤色模型的人脸检测[J].辽宁石油化工大学学报,2010,30(3):61-64.
- [15] KHAMMARI A, NASHASHIBI F, ABRAMSON Y, et al. Vehicle detection combining gradient analysis and AdaBoost classification [C]. Intelligent Transportation Systems. IEEE, 2015: 66-71.
- [16] 田仙仙,鲍泓,徐成.一种改进HOG特征的行人检测算法[J].计算机科学,2014,41(9):320-324.
- [17] VIOLA P, JONES M. Rapid Object Detection using a Boosted Cascade of Simple Features[C]// IEEE Computer Society Conference on Computer Vision & Pattern Recognition. 2003,41(1):112-117.
- [18] FREUND Y, SCHAPIRE R E. A decision-theoretic generalization of on-line learning and an application to boosting [J]. Journal of Computer and System Sciences, 1997, 55(1): 119-139.
- [19] RISTANI E, SOLERA F, ZOU R S, et al. Performance Measures and a Data Set for Multi-Target, Multi-Camera Tracking[J]. Discrete Dynamics in Nature, 2016(2):710-719.
- [20] HTIKE K K. Efficient determination of the number of weak learners in AdaBoost[J]. Journal of Experimental & Theoretical Artificial Intelligence, 2017, 29(5): 967-982.

A Pedestrian Model Recognition Detection Algorithm Based on Foreground Extraction

YANG Zhongming¹, LI Zilong², HU Yinwen², HUANG Han², CAI Zhaoquan³

(1.Guangdong Polytechnic of Science and Technology, Zhuhai 519090, China; 2.South China University of Technology, Guangzhou 510006, China; 3.Huizhou University, Huizhou 516007, China)

Abstract: In this paper, the algorithm principles of Gaussian mixture model, HOG+SVM classifier and Haar+Adaboost classifier were explored. A pedestrian detection algorithm based on foreground extraction and pattern recognition was proposed. The background modelling was executed by using Gaussian mixture model and then the moving object was entracted by using foreground modeling methods. The pedestrian detection based on the moving objects and face recognition on the objects were execnted. By this, the misjudgment problems was solved based on background modeling methods and efficiency problems based on statistical learning methods. The experimental results showed that the new algorithm could greatly reduce the missed detection rate compared to using the pattern recognition algorithm alone, and it performed well in terms of running speed and detection rate.

Key words: background modeling; pedestrian detection; face detection; AdaBoost; model recognition