

# 人脸检测&人流量统计分析

## 目录

### [Chapter 1: 前言](#)

### [Chapter 2: 人脸检测算法](#)

#### [1、人脸分析系统](#)

#### [2、人脸检测算法类型](#)

#### [3、Adaboost 人脸检测](#)

#### [4、人脸跟踪](#)

### [Chapter 3: 人流量统计算法](#)

## Chapter 1: 前言

I

首先说明，此报告把人脸检测和人流量统计放在一起分析是因为他们在目标检测阶段都通常使用基于分类器的方法。如人脸检测在目标检测阶段是检测人脸，而人流量统计是检测人头（该方式可以提供较高的统计精度）。

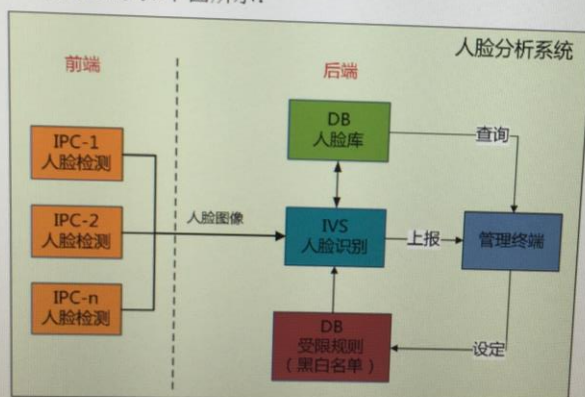
人脸检测一般来说比一般的智能功能更加复杂，在现有的理论体系内，其实现的方式也比较多，但是比较实用的或者说一般厂商会采用的应该都是基于分类器的人脸检测算法。后续会简单介绍各类检测算法，着重讨论基于级联分类器的算法。

人流量统计从其统计精度由低到高来看，可以分为三类：基于普通的周界算法、基于分类器的算法和基于双目立体视觉的算法，下文中会具体讨论每类算法实现及其优缺点

## Chapter 2: 人脸检测算法

### 1、人脸分析系统

人脸检测作为整个人脸分析过程的前处理，对后期的人脸识别的准确率有着比较大的影响，一般该过程在 IPC 端完成，以减轻后端人脸识别服务器的压力。通常一套完整的人脸分析系统如下图所示：



如图所示，对前端的

IPC 侧，我们只需关注人脸检测功能（对于嵌入式设备，这是不小的运算量）。而相对于人脸检测，人脸识别的运算量则是非常庞大的（和人脸库的规模成正比）。并且目前没有单一的人脸识别算法，可以在实际中达到很高的识别率，所以通常是多种识别算法加

权并用来提高识别率。在实际应用中，由于人脸库很庞大（如公安身份证照片库），人脸识别通常分布式的运行在一个集群上，以加快处理速度。

## 2、人脸检测算法类型

### A、基于知识的算法

这个方法核心是把人脸关于人脸的一些认知转化为一些可以让计算机识别的规则。比如人脸一般是椭圆型的，椭圆内部是对称的，存在对称的器官分布（如一对眼睛）。

这种基于知识的方法是自上而下的分析方式，其核心是如何将人类知识转化成为有效的计算机规则。困难是：如果规则划定得太细，那么可能有许多真的人脸无法通过规则的验证；如果规则划定得太宽泛，那么又可能有许多非人脸会被误判定为人脸。

这个方式是比较粗陋的，检测的性能很有限，也很难提高，但是运算量相对于其他算法是比较小的。

### B、模板匹配的算法

比较经典的模式识别算法，在预处理阶段，手动的预定义一个标准化的人脸图案模板，即做到尺度和灰度的归一化。然后在画面中滑动查找，计算预定义模板与采样模板的相关值，达到一定阈值即认为该区域为人脸。

该算法的改进形式，其会独立计算脸部轮廓、眼睛、鼻子和嘴部位的相关值后再统一的衡量采样模板是否属于人脸。

这类算法实现起来比较简单，但是效率并不高，检测性能也很有限。

### C、基于机器学习的算法

## 基于机器学习的算法有多种

如基于表象的机器学习，这类方法的模型是从一系列具有代表性的脸部的训练图像学习而来，再将学习而成的模型用于人脸检测，而不像基于模板匹配的算法，模板是由专家预先定义的，这是一种自下而上的算法。这类算法通常利用统计分析和机器学习的技术来寻找人脸图像的有关特性。学习而来的特性总结成分布模型或者判别函数，再利用这些分布模型或者判别函数来检测人脸。这类算法具有代表性的，如神经网络相关的算法。

其他如基于支持向量机 SVM、HMM 等人脸检测算法我们不再一一描述。重点关注现在业界使用较为广泛和成熟的基于 Adaboost 分类器的算法。

## 3、Adaboost 人脸检测

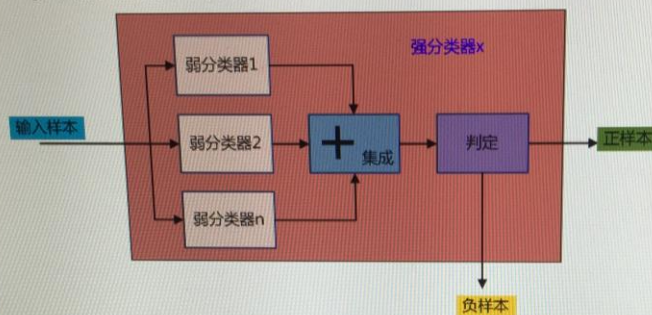
### A、分类器架构

在讨论 Adaboost 之前我们先简单说下 Boosting 相关的算法。其核心原理只要有足够的数据，弱学习算法就能通过集成的方式生成任意高精度强学习方法。打比方随机猜测一个是或否的问题，将会有 50% 的正确率。如果一个假设能够稍微地提高猜测正确的概率（如 52%），那么这个假设就是弱学习算法，得到这个算法的过程称为弱学习。使用半自动化的方法为多个任务构造弱学习算法（构造过程需要数量巨大的假设集合，通常这个假设集合是基于简单的规则组合和对样本集合的评估而生成的），如果一个假设能够显著地提高猜测正确的概率，那么这个假设就称为强学习。

Adaboost（Adaptive Boosting）是 Boosting 算法的改进，该方法根据弱学习的结果反

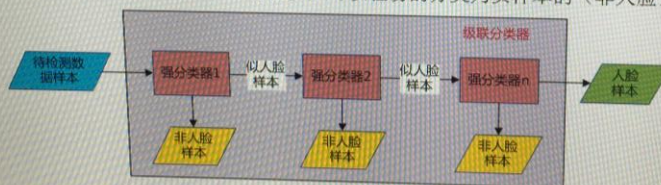


自适应地调整假设的错误率，所以 Adaboost 不需要预先知道假设的错误率下限（所以不需要任何关于弱学习器性能的先验知识），能够更加自动化的进行训练。



如上图所表示的，Boosting 的核心概念就是把许多弱分类器叠加形成强分类器（理论基础来自 Kearns 和 Valiant 的证明，不深入探索）。

实际中，一个强分类器中可能会包含有超过 200 个弱分类器，简单的平推弱分类器执行，会导致运行效率较低，因为所有的样本必须逐个通过所有的分类器。但是实际中输入分类器的大量样本，有很多是可以轻易的分类为负样本的（非人脸）。



如上图所示级联分类器就是对前面问题的解决，将多个强分类器通过级联的方式前后连接，每个

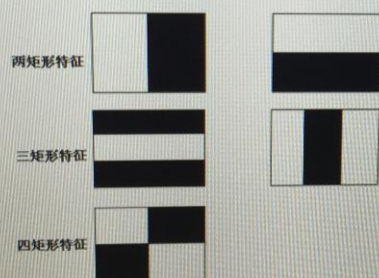
强分类器由多个弱分类器组成，且每一级强分类器都比前一级复杂。每一级分类器可以让几乎所有的正例通过，同时可以滤除大部分负例。这样每一级的待检测样本就比前一级少了一些，随着逐级深入，会排除大量的非检测目标，从而可大大提高检测速度(而且从结构上来说前面简单的强分类器处理的样本较多，后面复杂的强分类器处理的样本较少，也是符合减小运算量、加速检测的思想的)。

通常一个级联用的强分类器包含 20 个左右的弱分类器，然后在将 10 个强分类器级联起来，就构成了一个级联强分类器。而这样结构的级联分类器和单独的一个强分类器（包含 200 个弱分类器）的性能差不多，但是检测速度上却有很大的提升。

### B、分类器基于的特征

前面讨论了人脸检测分类器的总架构，再来具体讨论，其组成分类基础：特征。

此处我们考虑采用了 Haar\_Like 特征，即图像的矩形特征。下图为一般会用到的矩形特征模板：



通常两矩形特征反映的是边缘特征，三矩形特征反映的是线性特征而四矩形



特征反映的是特定方向特征。



上图中两个矩形特征，表示出人脸的某些特征。比如中间一幅表示眼睛区域的颜色比脸颊区域的颜色深，右边一幅表示鼻梁两侧比鼻梁的颜色要深。同样，其他目标，如眼睛等，也可以用一些矩形特征来表示。使用特征比单纯地使用像素点具有很大的优越性，其能够编码特定区域的信息，且相对于基于像素的系统，其运算量会小很多。

实际中必须通过特定算法选择合适的矩形特征，并将其组合成强分类器才能有效的检测人脸。

每个特征模板都有其对应的特征值，特征值定义为：白色矩形像素和减去黑色矩形像素和。为了加速计算通常采用基于积分图的方式来计算特征值。

#### B、弱分类器和强分类器的训练

训练弱分类器即对与某一矩形特征模板，计算所有样本的该特征的特征值，然后寻找最优的分类阈值的过程（即基于该特征的弱分类器对所有训练样本的分类误差最低），计算公式不在此详述。

训练强分类器时，先确定当前强分类器中需要  $T$  个弱分类器。然后以迭代的方式进行弱分类器的挑选，迭代过程中，每次改变样本的权重（对于分类错误的样本，加大其对应的权重；而对于分类正确的样本，降低其权重，这样分错的样本就被突显出来，从而得到一个新的样本分布），选择错误率最小的最优弱分类器。

总的来说样本的训练是个很庞大的计算过程，以  $20 \times 20$  的归一化样本为例，其包含近 80000 个矩形特征。分析可知，影响训练计算量的两个因素：1、样本数量 2、矩形特征数量。样本数量是不该减小的，因为会影响分类器的性能。而固定大小的样本中的矩形特征数量则可以按实际情况分析，予以裁剪（通常最终也只选取了 200 个最优特征）。

对于减少矩形特征数量，主要有两个途径：1、在特征生成阶段人为剔除。如完全不必考虑宽度为 1-2 像素的矩形特征，因为用这样的特征计算特征值随机性太大，不太可能寻找到合适的分离阈值。此外根据实际训练的图像可以看到，训练集中的人脸特征都集中在画面中部，可以适当减少样本画面边缘的矩形特征。2、在训练过程中动态地减少矩形特征。如对于在训练中，阈值的选取对错误率降低成效不大的那些矩形特征（即对人脸和非人脸区分能力很低的），可以设定一个错误率范围（逐轮改变），在每轮训练中按照这个范围，对矩形特征进行剔除。

## 4、人脸跟踪

人脸检测阶段进行人脸跟踪是必要的，这是为了不传输重复的人脸给后端处理。而且很重要的一点：从人脸进入画面到离开画面，人脸的姿态在不断的改变，这个过程中对一张人脸进行跟踪，然后建立一定的评价机制，选择最优姿态的人脸，传递给后端进行识别（提高识别率）。此处跟踪算法一般不必太复杂，通用的周界相关跟踪算法即可。因为人脸检测场景下，目标人脸的呈现一般比较清晰，目标也比较大，跟踪不会太困难

(易于周界), 此处不做深入讨论, 详情参考周界分析报告。

## Chapter 3: 人流量统计算法

人流量统计算法根据其统计精度的由高到底一般可以分为三类: 1、基于周界类似的算法 2、基于分类器的算法、3、基于双目立体视觉的算法。除双目立体视觉外, 其他两类算法都在其他章节或报告中有详细分析, 所以该章节会比较简略的描述下相关算法。

### 1、基于周界算法的人流量统计

这类人流量统计的实现, 完全是基于周界中的绊线检测算法, 或由其优化而来。其实现是基于现有的周界功能。缺点很明显, 精度不高, 通常周界只需要考虑是否有车、物绊到警戒线, 而不关注是否有几个人或物同时绊到警戒线(不处理几个人融合的或遮挡的情况), 而这恰恰影响了人流量统计的精度。该类人流量统计方式通常只适合在人流量较低的场景下勉强使用(即一般情况下同一时间内不会有多个人一起触发绊线计算)。

### 2、基于分类器的人流量统计

相比于基于周界的人流量统计方式, 分类器的优势在于其能逐个单一的分辨行人目标。当然实际中不会以整个行人作为训练样本, 为了提高精度, 根据安装位置的不同,

我们会选用不同的训练样本。如在出口正上方安装时, 一般选取头部俯视图像最为训练对象进行识别; 而在正对入口进行检测时, 一般选用人脸部位图像进行训练。这两种典型的安装方式相比, 前者俯视安装具有较高的统计精度, 因为该视角下的头顶部位图像一般不会发生遮挡的情况; 而后者由于行人的前后重叠, 脸部较容易出现遮挡, 导致漏计。不过前者的缺陷是无法拍摄到行人的详细特征(即无法发挥典型的监控设备的作用), 只能作为人流量统计设备存在。详见下图示例: 俯视和正对入口



这类算法的分类器实现, 可以参考人脸检测的 Adaboost 分类器(参见前一章节)。当然相对于基于周界的算法, 这类算法的明显缺陷是运算量会更大; 此外需要额外的训练阶段来实现更好的头部或脸部检测。

### 3、基于双目视觉的人流量统计

相比于前面两种算法该类算法的最明显优势是, 其具有业界最高的人流量统计精度(不容易受环境光照变化、场景变动等影响, 如可以安装在公交车门口等部位), 而且不需要进行类似分类器那样的训练过程。不过缺点在于: 其必须俯视安装, 此外依赖于特定的硬件(必须具有双 Sensor 输入), 通常也无法作为通用的 IPC 所使用。

下面具体说明下其相关实现原理。

相对于单目摄像机, 双目摄像机可以通过相应的模型计算出画面中某点的视差信息,



然后根据所有点的视差信息所生产的视差图像来识别头部目标，进行跟踪和计数。该类算法的核心内容在于视差图像的生成。

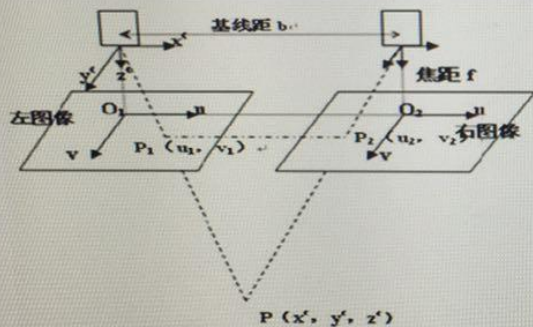


左右视图的一副

视差图像

二值化

关于视差图像的生成我们先看下理想的平视双目立体成像原理图：



上图中  $P$  点为空间一点， $P_1/P_2$  为  $P$  点分别在两个摄像机上的成像位置。由三角几何关系的相关公式可以计算出  $u_1$  和  $u_2$  的坐标值，而  $P$  点视差的定义就是  $(u_1 - u_2)$ 。

当然现实的世界中不可能存在理想的平视双目结构，通常要经过标定来得到两摄像机之间的仿射参数（一般只有平移和选择），后期可以通过该仿射参数把两个摄像机拍摄的图像对投影到一个公共的图像平面上（这个平面相当于理想的平视双目结构的成像

平面）。

现在的情况相当于我们已经知道了  $P_1$  但是不知道  $P_2$  在哪里；所以我们需要找到  $P_2$  的位置，即可计算视差。理论上对应于  $P_1$  的点  $P_2$  只存在一个。这里可以使用类似于 PTZ 跟踪中的图像配准算法，首先需要对校正后的立体图像对进行匹配。由于经过校正后，两幅图像中的对应点在图像的同一行中，因此在点匹配时只需要在相应的行中寻找对应匹配点（匹配算法此处不详细讨论）。以此类推到每一点，即得到了左右视图图像的视差图像。视差图像的二值化来得到靠近摄像机的区域（亮区），在此处的应用中一般为头部。后面的具体跟踪过程与其他算法类似，不做累述。

总结以上几种算法，人流量统计功能的实现可以依照实际应用的统计精度需求和芯片硬件条件，来选择。

附：查看了我司现有平台和可能使用的平台，Hisi3516 和 AMBA S2 都支持双 Sensor 输入，不过两颗芯片都是只有一路支持 ISP，另外一路需要外接。而实际中，双目立体视觉相关算法中，对于图像的一致性还是有比较高的要求的(如图像配准过程)，所以目前 SOC 的方案似乎还无法满足要求。