

DOI:10.3969/j.issn.1001-4551.2018.06.001

基于深度学习的气泡水平尺自动 矫正系统研究*

刘 尧, 朱善安*

(浙江大学 电气工程学院, 浙江 杭州 310027)

摘要:针对气泡水平尺大角度偏斜工件的自动矫正问题,对深度学习目标检测领域的模型进行了研究,设计了一种基于深度学习的自动矫正方案。在不同光照条件的气泡工件图片样本集上,采用k-means聚类方法,分析了不同anchor box数量对YOLO检测性能的影响,对比了YOLO与SSD两种模型的检测准确率和平均IOU;同时结合概率霍夫变换与最小二乘法,设计了两种气泡工件参考线边缘拟合方法,并对比了两种拟合方法计算工件偏斜角度的准确度;针对应用场景,采用Client/Server网络结构,服务端接收客户端采集的图像并将计算结果返回,客户端控制电机对偏斜工件进行了自动矫正。研究结果表明:该方案能对气泡大角度偏斜的工件进行检测,相比现有方案矫正的效率更高。

关键词:气泡水平尺;YOLO;SSD;概率霍夫变换;最小二乘法

中图分类号:TH73;TP274

文献标志码:A

文章编号:1001-4551(2018)06-0555-06

Automatic bubble level correct system based on deep learning

LIU Yao, ZHU Shan-an

(College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China)

Abstract: Aiming at the automatic correction of the bubble level with large angle of inclination, the models in the field of deep learning object detection were studied, and a method based on deep learning object detection models was designed. The YOLO and SSD models were trained on the data set of bubble level images with different light conditions. By adopting K-means clustering algorithm, the effect of different anchor box numbers on YOLO model's performance was analyzed. And two models' detection accuracy and average IOU were compared. Two edge fitting methods combining Progressive Probabilistic Hough transform with least square method were designed and two methods' accuracy was compared. Aiming at this application, Client/Server network structure was used. The images were sent to the server and calculated. The calculation results were sent back to the client to control motor to correct the bubble level. The results indicate that the proposed method can detect the bubble level with large angle of inclination, and is more efficient than the conventional methods.

Key words: bubble level; YOLO; SSD; progressive probabilistic hough transform; least square method

0 引 言

水平尺是一种检测或者测量水平和垂直度的测量工具。根据不同的工作原理,水平尺可以分为气泡水平尺、电子水平尺和激光水平尺等不同类型,使用最为广泛的是气泡水平尺。

气泡水平尺通过气泡与左右参考线的距离来判断是否水平。在气泡水平尺生产环节中,需要将水平尺放置在标准水平台上,根据气泡位于左右参考线之间的距离进行矫正。过去国内生产厂家大多由工人肉眼观察,判断气泡是否位于左右参考线正中央,再人工进行矫正。这种方案速度慢、主观性强而且误差大。

收稿日期:2017-10-30

基金项目:浙江东方精工水平尺自动检测校准平台软件开发项目(作者未提供)

作者简介:刘 尧(1993-),男,广西桂林人,主要从事图像处理、计算机视觉方面的研究。E-mail: 747243010@qq.com

通信联系人:朱善安,男,教授,博士生导师。E-mail: zsa@zju.edu.cn

随着技术的发展,机器视觉方法被应用到工业生产环节中。魏煜^[1-2]结合最优阈值变换和轮廓跟踪实现了水平尺偏移的高精度检测,盛伟^[3]结合 Canny 算子与加权最小二乘法设计了一种强鲁棒性的气泡端点定位方法,喻婷^[4]结合数学形态学和轮廓跟踪设计了一种快速准确的气泡偏移距离计算方法。

近年来,深度学习卷积神经网络在图像分类与识别上的研究取得了突破性进展。在目标检测领域中,YOLO(you only look once)^[5]和 SSD(single shot multi-box detector)^[6]是综合准确率与检测速度最好的两种方法。YOLO 检测效率达到了 45 fps,在 VOC2007 数据集上 mAP 达到了 63.4%。SSD 结合了 YOLO 的回归思想与 Faster R-CNN^[7]的 anchor boxes 机制,在 VOC2007 数据集上 mAP 达到了 74.3%,帧率达到了 59 fps。YOLO 与 SSD 两种模型都能达到实时检测的效率。

本文针对气泡水平尺大角度偏斜工件的自动矫正问题,对 YOLO 与 SSD 两种深度学习目标检测模型进行研究,设计一种基于深度学习的气泡水平尺自动矫正方案,采用 C/S 网络结构,客户端采集图像并发送至服务端,服务端使用深度学习目标检测模型对图像进行检测,提取左、右参考线所在的图像区域,结合概率霍夫变换与最小二乘法拟合直线,计算倾斜角度结果返回客户端,客户端控制电机将大倾斜角度矫正到水平的位置。

1 自动矫正

自动矫正是指在标准水平台上,通过计算机控制电机旋转气泡工件,将气泡置于左、右参考线的正中间。

工业摄像机采集到的气泡工件图像如图(1~3)所示。



图1 合格的气泡图像



图2 小偏斜角度的气泡图像

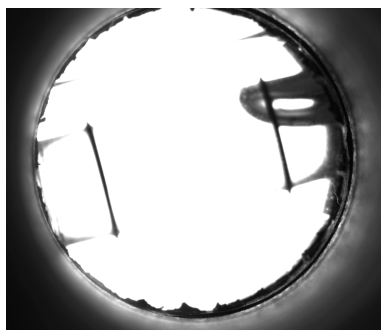


图3 大偏斜角度的气泡图像

其中,图1是矫正合格的图像,气泡位于左、右参考线的正中间。图2与图3是待矫正的气泡图像,图2气泡工件偏斜度较小,气泡位于左、右参考线之间但不居中,图3偏斜度较大。文献[4]中的方案能够在图2所示的情形下,快速准确地计算出气泡的偏移距离。但对图3的情形,其算法不能一步到位,而是采用逐步调整法,控制电机以一定步长逐步调整使气泡位于左右参考线之间,再进行分析处理。

逐步法每次调整都要对图像重新进行采样分析,如果步长设置小需要较长的调整时间,步长设置过大又可能出现震荡。而气泡任意时刻偏斜角度的信息包含在左、右参考线上,工件偏斜角度与参考线的角度呈互余的关系,以参考线为目标,对图像进行检测,提取目标并计算参考线的倾斜角度,就可以一步到位地对大偏斜角度的工件进行矫正。

2 目标检测

本文中要检测的气泡水平尺的平行参考线在矫正过程中只有倾斜角度的变化,因而对不同倾斜角度的图像进行采集作为模型训练样本。同时为了使模型具有鲁棒性,样本集中加入了不同光照条件下的图像。

本文采集了1 000张不同倾斜角度不同光照的气泡图片作为样本集,并对样本集中的参考线进行标定,随机选取其中的800张样本作为训练集,剩余200张作为测试集。额外采集100张图片作为验证集。

2.1 YOLO 模型

YOLO 使用整张图像作为输入,将目标检测问题转化成回归问题,在网络输出层回归位置信息和类别信息,极大地提高了检测效率。YOLOv1 将输入图像划分为 $S * S$ 个网格,每个网格预测 B 个目标外接框。在输出端,每个外接框输出置信度和预测位置信息 (x,y,w,h) ,置信度计算公式如下:

$$Conf = Pr(Object) * IOU_{pred}^{truth} \tag{1}$$

式中:Pr(Object)—当前外接框预测目标存在的精确度;IOU—衡量目标位置预测精确度的指标。

假设预测目标外接矩形框为 A ,标定的真实目标外接框为 B ,IOU 定义为 AB 交集与并集的面积比,表达式如下:

$$IOU_A^B = (A \cap B)/(A \cup B) \tag{2}$$

同时,每个网格输出类别信息(C 个类别),最终网络输出 $S * S * (B * 5 + C)$ 维向量。网络模型由 24 个级联卷积层加上两个全连接层构成,网络需要对目标是否存在以及目标的位置与类别进行回归。

损失函数为:

$$\begin{aligned} &\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \\ &\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] + \\ &\sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{noobj} (C_i - \hat{C}_i)^2 + \\ &\sum_{i=0}^{S^2} I_{ij}^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2 \end{aligned} \tag{3}$$

式中: (x,y,w,h) 部分—目标位置和尺寸的预测; C 部分—当前外接目标框是否包含目标的置信度预测; $p(c)$ 部分—目标的类别预测。

YOLOv2^[8] 将原来的网络结构进行了改进,移除了全连接层,借鉴了 Faster R-CNN 中的 anchorboxes 方法,使用 anchor boxes 来预测最终的目标外接框。anchor boxes 是一种目标框搜索方案,由一组中心点相同、不同宽高比的矩形构成,anchor boxes 示意图如图 4 所示。

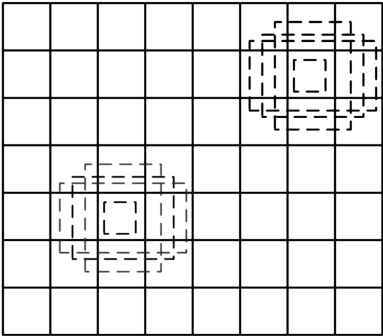


图4 anchor boxes 示意图

图 4 中:虚线方格为预设 anchor boxes。

Faster R-CNN 中,anchor boxes 的个数与宽高维度都是手动设定的先验框,anchor boxes 的设计与最终的检测精度直接相关。但不同样本集中,目标形状大小各不相同,对特定目标的检测依照手动设定的方法有时并不能取得很好的效果。YOLO v2 中采用 k-means 聚类的方法,对于样本集标注的目标框真实宽高数据进行聚类,以聚类类别数 k 为最终 anchor boxes 候选框的个数,以每一类别的中心对应的宽高作为该 anchor box 的宽高,在数据集上进行性能分析。本文使用气泡样本集对应的目标标注信息作为聚类样本,anchor box 数取 1 到 6,分别迭代训练 10 000 次后,不同模型在验证集上得到的结果如表 1 所示。

表 1 不同 anchor boxes 数测试结果

anchor boxes 数量/个	AvgIOU	误检数/个
1	0.687	2
2	0.705	0
3	0.717	3
4	0.726	0
5	0.729	0
6	0.730	0

从表 1 中可以看出:随着 anchor boxes 数量的增大,验证集上的平均 IOU 越高,增量越来越少,anchor boxes 数量取值为 5 之后几乎不再增长,这与文献[8]中呈现的规律相仿;并且 anchor boxes 数量设定越高,待选目标区域越多,计算时间越长。综合精度、误检数和计算时间,本文最终设定 anchorboxes 数为 5。

2.2 SSD 模型

SSD 是全卷积结构,同样采用 anchor boxes 方法,不同之处在于 SSD 使用了不同维度的特征图,高维特征图中的一个点对应了原始图像的一块区域,低层特征图的感受野较小,高层的感受野较大。不同特征图的感受视野不一样。SSD 在不同尺度特征图提取了 8 732 个 anchor boxes 信息,最后对所有的 anchor boxes 做非极大值抑制输出候选结果。SSD 的损失函数为:

$$L(x,c,l,g) = \frac{1}{N} (L_{conf}(x,c) + \alpha L_{loc}(x,l,g)) \tag{4}$$

损失函数包括了置信度损失与位置损失两部分,其中置信度部分的损失函数为:

$$L_{conf}(x,c) = - \sum_{i \in Pos} x_{ij}^p \log(\text{softmax}(c_i^p)) - \sum_{i \in Neg} \log(\text{softmax}(c_i^0)) \tag{5}$$

位置部分的损失函数为:

$$L_{\text{loc}}(x, l, g) = \sum_{i \in \text{Pos}} \sum_{m \in \{cx, cy, w, h\}} x_{ij}^k \text{Smooth}_{L1}(l_i^m - \hat{g}_j^m) \quad (6)$$

$$\hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx}) / d_i^w \quad (7)$$

$$\hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy}) / d_i^h \quad (8)$$

$$\hat{g}_j^w = \log\left(\frac{g_j^w}{d_i^w}\right) \quad (9)$$

$$\hat{g}_j^h = \log\left(\frac{g_j^h}{d_i^h}\right) \quad (10)$$

式中: $\hat{g}$ —回归得到的目标位置尺寸 (x, y, w, h) 四维信息相对 anchor boxes 的补偿; g —anchor boxes 的位置与尺寸; d —先验标注的目标位置与尺寸信息。

2.3 对比分析

本文在相同的样本集上训练 YOLO 与 SSD 模型, 并在验证集上对比了 YOLO 与 SSD 检测结果的 IOU, 结果如图 5 所示。

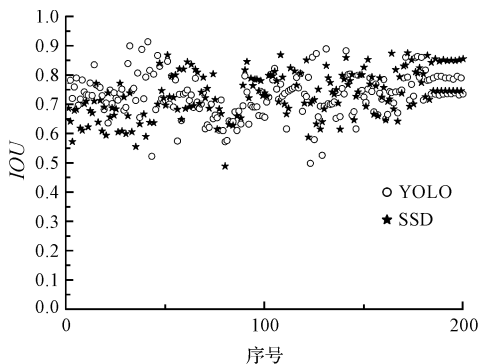


图 5 YOLO 与 SSD 在验证集上 IOU 对比

经统计, YOLO 检测一张图像的平均时间为 0.023 5 s, SSD 为 0.021 7 s, 均能达到实时检测的要求。验证集一共是 100 张图片 200 个目标, YOLO 模型漏检数为 0, 平均 IOU 达到了 0.729。SSD 模型漏检数为 16, 平均 IOU 为 0.676。综合检测精度与效率, 本文最终选择 YOLO 模型, 检测效果如图(6,7)所示。



图 6 YOLO 的检测效果(光照条件良好)

由此可以看到: 在光线不佳的条件下, 模型也能准确地检测出工件的左、右参考线。

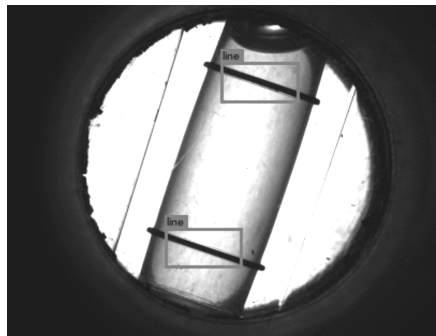


图 7 YOLO 的检测效果(光照条件不佳)

3 倾斜角度计算

由于参考线与矫正倾角是互余的关系, 研究人员能通过前景图像中参考线的倾斜角度折算出工件的倾斜角度。本研究采用 Canny 算子对获得的前景图像(工件的左右参考线)区域进行边缘提取, 如图 8 所示。

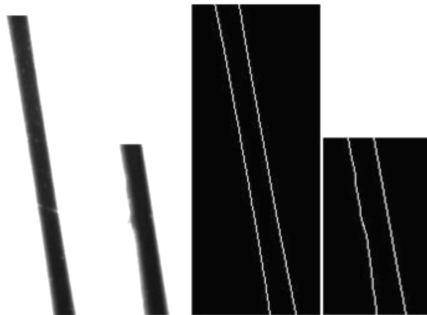


图 8 提取的前景图像与其边缘图像

获得边缘图像后, 需要对边缘直线方程进行拟合, 常用的图像直线拟合方法有最小二乘法和霍夫变换。

(1) 最小二乘法是最常见的线性回归方法之一, 通过最小化误差平方和能获得亚像素级别的回归直线。最小二乘法速度快, 但对噪声敏感, 拟合数据点集的选取直接影响拟合精度。

(2) 霍夫变换是一种参数转换算法, 将图像的直角坐标系 $(x-y)$ 空间) 映射到极坐标参数空间 $(\theta-\rho)$ 空间), 再对极坐标空间下的直线参数进行累计投票, 最终确定直线方程。坐标转换方程为:

$$x \cos \theta + y \sin \theta = \rho \quad (11)$$

式中: θ —直线法线与 x 轴的倾斜角度 $(0 \leq \theta \leq 180^\circ)$; ρ —原点到直线的距离。

霍夫变换以一定分辨率将极坐标参数 θ 与 ρ 离散化为若干区间, 每个区间设置一个计数器, 对每个数据点 (x_i, y_i) , 遍历离散的 θ_j , 求得相应的 ρ_j , 相应区间的计数器增加 1。最终计数器值大于设定阈值参数的便是获得的直线参数对 $\theta-\rho$ 。霍夫变换参数分辨率影响了计算时间, 过度细分的参数空间可能无法获得足够

的计数造成检测失败^[9]。改进的概率霍夫变换^[10]采用随机抽取的方式,同样采取计数器,当相应参数对计数值大于设定阈值时,便将这条直线拟合出来,有效提高了标准霍夫变换的耗时问题。

由于参考线本身并不是标准的直线,在设备安装过程中,图像采集摄像机与工件存在细微的偏角,加上光照的影响,获取到的参考线可能有弯折,如图8中右边参考线的边缘。本文结合概率霍夫变换和最小二乘法,首先用概率霍夫变换去除噪声,然后对保留的点集进行最小二乘直线拟合。对一个工件能提取两条参考线的4条边缘线,本文设计了两种工件倾斜角度的拟合方法:

方法一:对4条边缘点集,采用概率霍夫变换获得潜在直线所在区域,以此排除掉参考线畸变的部分点集,之后对区域内的点集做最小二乘直线拟合计算边缘倾角,最后取4条边缘倾斜角度的平均值作为整个工件的倾斜角度。

方法二:考虑工件参考线的粗细,获取两条边缘线每一行的中点作为待拟合直线,采用概率霍夫变换与最小二乘进行拟合,以两条中心线拟合结果的均值作

为整个工件的倾斜角度。

4 实验对比

图像像素点以矩阵形式存储,其坐标都是实整数。除了0°,45°,90°,135°倾角的直线能够精确地拟合(均方误差和为0),其他角度的边缘直线都有一定的测量误差,故以人工测量图像中直线的倾斜角作为标准来评判没有意义。本文的目的是尽可能地将工件一步到位矫正到水平位置,而矫正角度由输出给伺服电机的脉冲数数量决定,因此,从水平合格位置起,给定脉冲数量,将工件旋转一定角度,通过YOLO模型提取参考线区域,并对参考线边缘点集进行拟合,折算成要矫正的脉冲数,以此为标准对两种工件倾斜角度拟合方法进行对比分析。

本研究中,伺服电机10⁶个脉冲对应360°,每个脉冲矫正的角度为3.6°×10⁻⁴。经过测算,气泡从合格位置到与任意一条参考线相触的脉冲数为1650。分别将工件从合格位置处顺时针和逆时针旋转10000,20000,30000,40000,50000和60000个脉冲,获得的实验结果如表2所示。

表2 不同折算方法结果对比(单位:脉冲数)

实际脉冲数(顺时针)	方法一	方法二	实际脉冲数(逆时针)	方法一	方法二
-10 000	-6 768	-10 190	10 000	10 674	10 594
-20 000	-16 434	-18 858	20 000	20 151	21 096
-30 000	-25 528	-29 306	30 000	29 765	29 758
-40 000	-38 690	-41 050	40 000	41 407	41 602
-50 000	-50 457	-49 799	50 000	48 463	49 747
-60 000	-60 097	-58 914	60 000	60 896	60 874

由表2中数据可以看出:方法一对逆时针方向计算比较准确,最大误差为1573个脉冲,而在顺时针方向最大误差为4472个脉冲数,大于气泡与参考线相碰的脉冲数,不能将气泡矫正到参考线之间;方法二在两个方向上的最大误差为1602个脉冲,小于1650,能够将工件从大角度偏斜矫正到气泡位于左右参考线之间的位置。因而本文选择方法二作为计算工件倾斜角度的方法。

深度学习网络模型计算速度与硬件性能相关,综合硬件成本与实际生产场景,本文采用C/S网络结构,主要流程为:客户端采集图像并将数据发送至服务端,服务端计算最终结果返回至客户端,客户端控制电机对工件进行矫正。本文将该方案与文献[4]的方法进行了效率对比,测试矫正10个工件的平均时间如表3所示(测试平台:客户端 windows7,i3-3.7 GHz;服务端:Ubuntu16.04,i7-4.20 GHz, GeForce GTX 1080)。

表3 不同倾斜角度下矫正平均时间对比(单位:s)

原始偏斜脉冲数	原方案矫正时间	本方案矫正时间	原始偏斜脉冲数	原方案矫正时间	本方案矫正时间
10 000	12.46	9.67	40 000	14.95	10.09
20 000	13.52	9.86	50 000	15.71	10.32
30 000	14.27	9.95	60 000	16.32	10.58

(下转第602页)

本文引用格式:

刘尧,朱善安. 基于深度学习的气泡水平尺自动矫正系统研究[J]. 机电工程,2018,35(6):555-559,602.
LIU Yao, ZHU Shan-an. Automatic bubble level correct system based on deep learning[J]. Journal of Mechanical & Electrical Engineering, 2018,35(6):555-559,602.
《机电工程》杂志: <http://www.meem.com.cn>