



同濟大學

TONGJI UNIVERSITY

硕士学位论文

(专业学位)

车载人机交互零件的视觉检测系统设计

姓名：王彤

学号：1231622

所在院系：汽车学院

职业类型：工程硕士

专业领域：车辆工程

指导教师：罗峰

副指导教师：沈建于

二〇一七年一月



同濟大學
TONGJI UNIVERSITY

A dissertation submitted to
Tongji University in conformity with the requirements for
the degree of Master of Engineering

System Designing for Vision Inspection of Vehicle Human-Machine Interface Parts

Candidate: Wang Tong

Student Number: 1231622

School/Department: School of Automotive Studies

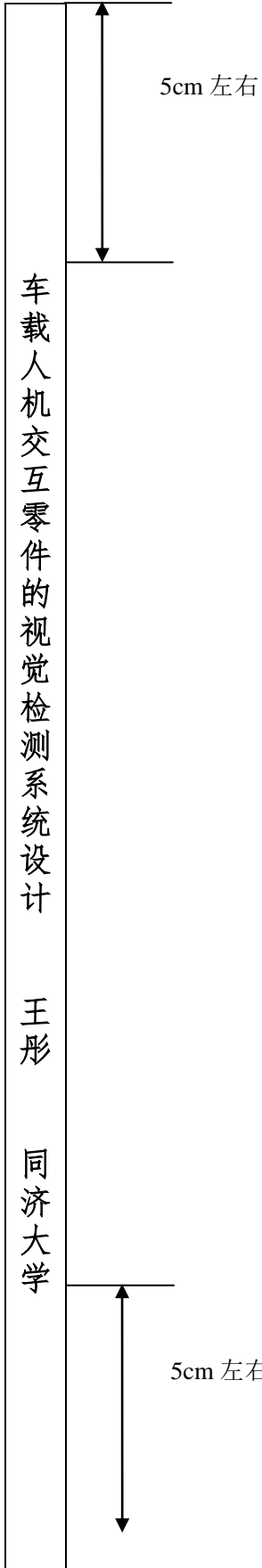
Discipline: Engineering Master

Major: Vehicle Engineering

Supervisor: Luo Feng

Vice Supervisor: Shen Jianyu

Jan, 2017



学位论文版权使用授权书

本人完全了解同济大学关于收集、保存、使用学位论文的规定，同意如下各项内容：按照学校要求提交学位论文的印刷本和电子版本；学校有权保留学位论文的印刷本和电子版，并采用影印、缩印、扫描、数字化或其它手段保存论文；学校有权提供目录检索以及提供本学位论文全文或者部分的阅览服务；学校有权按有关规定向国家有关部门或者机构送交论文的复印件和电子版；在不以赢利为目的的前提下，学校可以适当复制论文的部分或全部内容用于学术活动。

学位论文作者签名：

年 月 日

同济大学学位论文原创性声明

本人郑重声明：所提交的学位论文，是本人在导师指导下，进行研究工作所取得的成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或者没有公开发表的作品的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律责任由本人承担。

学位论文作者签名：

年 月 日

摘要

在汽车内饰件行业，目前针对车载人机交互面板的光刻信标的质量检测，在实际的生产环节中还是大多依赖于人工判断，效率较低而且准确度无法测量。本文基于计算机视觉设计了一种适合工厂自动化产线的在线视觉检测系统，替代人工检测。

通过对车载面板开关的工艺工程分析，确定了颜色、亮度和缺陷检测目标。对整个系统的软硬件设计和选型进行了论述。针对颜色检测提出一种可行的白平衡标定方案。针对亮度检测提出了一种使用极限样件进行参数传递的低成本的检测方案，使信标的颜色、亮度可以在精确的范围管控下检测。同时针对缺陷检测使用了一种基于深度置信网络和在线优化算法相结合的深度学习模型结构方案，将无法参数化的一些缺陷检测指标进行特征抽象建模并通过逐次累加单个样本数据对模型进行重构，对无法通过精确范围度量的缺陷进行机器自动检测。此系统能够克服由于使用传统机器视觉系统对缺陷检测误判率高的情况，最终实现和人眼的识别要求接近的测量效果。

通过对亮度检测的测量系统分析以及缺陷的实际准确率分析，最终验证此测量系统满足测试要求。最后对进一步优化和拓展的工作进行了简述。

关键词：机器视觉，深度学习，颜色亮度检测，缺陷检测，深度置信网络

ABSTRACT

In the automotive interior decoration industry, at present, the judgment of the quality of the optical indicator in the Human Machine Interface panel still depends on the manual judgment in the actual production process. The manual judgment is low efficiency and immeasurable. An online vision inspection system is designed based on computer vision technology instead of manual judgment.

Through the analysis of process for Vehicle Human-Machine Interface Parts, the detection target of color, brightness and defects is confirmed. The hardware and software design and selection of the whole system are discussed. This paper proposes a feasible method of white balance calibration for color detection and a low cost inspection method of transferring the parameter with the limit samples for brightness inspection so that the color and brightness can be detected under the exact limit control. Meanwhile this paper uses a deep learning structure based on deep belief network and online fine tune algorithm for defects inspection. It can be used to automatic detect the defects without accuracy parameter by modeling abstracted features for defects which cannot be parameterized and remodeling with iteration of single sample. This system overcomes the traditional machine vision system of high error rate of defects detected and finally achieve the similar effect with human eyes for measurement.

Through the measurement system analysis for brightness inspection and accuracy analysis for defects inspection, the vision inspection system can meet the measurement requirements. Finally the optimization and further studies are discussed.

Key Words: Computer vision, Deep learning, Color and brightness inspection, Defect inspection , DBN

目录

第 1 章引言.....	1
1.1 论文背景及意义.....	1
1.2 国内外研究现状.....	2
1.3 主要研究内容及拟关键技术.....	7
1.4 小结.....	7
第 2 章视觉检测系统设计.....	9
2.1 视觉检测需求分析.....	9
2.2 系统设计分析.....	13
2.3 系统设计.....	14
2.3.1 系统拓扑设计.....	14
2.3.2 系统整体设计.....	15
2.3.3 终端计算机设计选型.....	15
2.3.4 服务器设计选型.....	16
2.3.5 PLC 设计选型.....	16
2.3.6 相机与镜头设计选型.....	16
2.4 核心软件简介.....	18
2.4.1 机器视觉软件简介.....	18
2.4.2 TensorFlow.....	19
2.4.3 Python.....	20
2.5 小结.....	20
第 3 章机器视觉系统设计.....	23
3.1 检测过程.....	23
3.2 色彩空间.....	24
3.3 硬件标定.....	25
3.3.1 参数标定.....	25
3.3.2 标定板标定.....	26
3.3.3 白平衡标定.....	27
3.4 软件标定.....	28
3.5 小结.....	31
第 4 章缺陷检测系统模型设计.....	33
4.1 图像预处理过程.....	33
4.2 深度学习模型建模.....	37
4.2.1 RBM 限制波尔兹曼机.....	37
4.2.2 DBN 深度置信网络.....	43

4.2.3 深度学习建模过程.....	47
4.3 动态模型建模.....	48
4.3.1 在线优化算法.....	48
4.3.2 在线优化算法建模.....	49
4.4 小结.....	49
第5章视觉检测软件设计.....	51
5.1 基础软件平台架构.....	51
5.2 系统软件设计.....	51
5.2.1 系统总体软件设计.....	52
5.2.2 终端计算机软件设计.....	52
5.2.3 缺陷检测模型生成软件设计.....	53
5.2.4 服务器模型执行软件设计.....	54
5.2.5 服务器模型更新软件设计.....	55
5.3 小结.....	56
第6章视觉检测实验及结果分析.....	57
6.1 亮度检测评价过程及结果.....	57
6.2 参数调整验证.....	58
6.2.1 BATCH 调整验证.....	58
6.2.2 EPOCH 调整验证.....	59
6.2.3 LEARNING RATE 调整验证.....	60
6.2.4 在线优化算法评估验证.....	61
6.3 缺陷检测评价过程及结果.....	63
6.4 小结.....	64
第7章结论与展望总结.....	65
7.1 结论.....	65
7.2 展望.....	65
致谢.....	67
参考文献.....	68
个人简历、在读期间发表的学术论文与研究成果.....	71

第1章 引言

1.1 论文背景及意义

随着家用汽车的普及，家用轿车正在从以前单纯的运载工具转化为人们享受品质生活的一部分。人们对汽车内饰件的外观要求不断提高，因而驱使汽车内饰件中的车载人机交互面板开关的外观样式不断的优化改进，开关形式也从最初的机械式开关到现在的机械电子式开关。车载人机交互面板常用在各种开关或指示灯部分，用于接收驾乘人员的动作指令以及提示驾乘人员，当夜幕降临，汽车内部的人机交互面板开关即会点亮，指示驾乘人员车辆状态以及作为开关触控的引导指示。车载人机交互面板开关对于表面光刻质量的要求比较高，其光刻和装配质量直接关系到驾驶员和乘客的视觉观感，影响汽车的整体品质。在一些高档汽车上，对于车载人机交互面板的要求尤为严格，对发光信标的完整度，亮度，均匀度，颜色，暗点，单边尺寸等均有较高的检测要求。通常整车厂对于开关的信标的要求从在亮度方面做了严格的阈值定义，并且暗点毛边以及亮度的不均匀是不被允许的。而在实际的生产线上，有多个工艺过程影响着最终的信标品质。从光刻工序，LED(Light Emitting Diode)贴片工序，导光条装配工序再到总成装配工序中的任何一个工序出现偏差均会使最终的面板质量达不到设计要求。在生产现场最常见的检测方式是使用人眼判断的方式进行信标质量控制，对人员有特定的选择和培训要求，在经过严格培训后才可进行光刻信标的筛选。但是人有一定的不可控因素，当人员进行长时间重复检测后，会有一种疲劳和思维惯性的风险，使质量控制出现漏洞。再加上为此检测需要增加额外的人员投入，而且这种人员投入随着面板的复杂程度提高，生产成本会随之急剧提高，出现质量缺陷的风险也大大上升。

在工业领域，通常使用机器视觉技术来进行零件表面检测。近些年发展比较快速的视觉算法比如 Halcon 和 NIVision 均是典型的商用视觉系统的代表，这些视觉算法对于表面图像的定位、测量、滤波、颜色和亮度等均有对应的算法可以直接处理，速度也比较快。但算法多集中在特定条件的特征的检测，在实际的生产线检测应用领域，对于均匀度、暗点、尺寸和毛边的检测使用传统的机器视觉技术经常发生精度低和误判率高的问题，一次检测通过率较低，在一些极端条件下甚至低于50%。因此在生产过程中，当需要对车载人机交互零配件进行综合的表面检测时，无法完全脱离工人目视判断。在误判率较高的情况下，必须增加人工复检，大大增加了生产成本。传统机器视觉技术无法较好的达成目标的根本原因是由于对这些指

标的检测，首先会遇到参数难以量化的问题。由于阈值的设置均为固定参数，无法通过一些新的需求进行动态调整。其次，对于缺陷特征来讲，其具有不可预知性，也就是在于缺陷的发生前，无法获知其特定的缺陷状态，也就无法对其进行参数量化并将其检出，所有的检测方法均针对特定特征进行。再次，在线检测环境和要求不同于实验室环境，生产系统的检测需要实现快速，较为准确的找出由于工艺过程导致的光刻信标质量缺陷，而实验室的检测是在严格的条件下检验零件是否满足设计要求，这点上就是他们最大的差别，所以在生产现场是无法进行实验设备部署的。另外，从单品的生产成本考虑也无法满足要求。

如何才能设计一套系统，既可以实现亮度定量检查，又可以使传统的机器视觉系统完成复杂模糊的特征的判断，那么在生产过程中的无人化即可实现。而解决这一问题的关键，在于视觉系统具备同时亮度检测功能以及模糊特征的提取功能，那么引入机器视觉系统与人工神经网络相结合的方式进行处理是最好的解决方案。近来，深度学习理论的日趋成熟，其在各行各业的应用越来越广，它最大的优势在于可以通过抽象出特征值，来模拟人的思维模式，这是人工智能的一个基本的功能要求。在无人监督的或少量监督的情况下通过特征抽取与大规模的迭代运算实现自我学习^[1-5]。这种特性特别适合在一些需要人眼来识别的一些复杂场合，比如车载人机交互面板检测领域。相对于人眼的检测，摄像头在捕捉实际信标图像的稳定性要高很多，再配合深度学习算法即可实现人眼与人脑所完成的基本视觉感知能力。最终可以代替人眼进行产品表面的缺陷检测。而引入在线学习机制则是为了使深度学习的过程受控，加快原始模型对单个样品的检测结果趋向理想状态的过程。这样可以使深度学习模型实时插入单个偶然未知缺陷特征值，通过人为干预，达到符合人类大脑要求的算法模型，这也是在今后人工智能发展过程中的关键一环，使其更快的接近于人类思维模式，代替人的一些简单劳动。

将机器视觉技术结合深度学习理论的意义不只在在线缺陷检测有意义，对于所有的需要人来进行重复劳动的和进行感知判断的领域均有深远的意义，首先机器视觉技术可以感知和获取外界信息，就像人眼感知外围环境，从而构建自己的空间模型。人工智能又在此基础上将机器的感知加入了思维模式，使其可以自我完善，并可以理解人类对其的学习指导，使其向人员预想的方向发展，可代替一切人类感知和潜意识判断任务。

1.2 国内外研究现状

在光学检测领域比较常见的机器视觉商用软件主要有以下几种，Halcon 视觉处理软件，其由德国 MVtec 公司开发，包含了完善的标准的机器视觉算法库，常被

用于机器视觉集成开发领域，在国内外均有较多应用。Vision Pro 视觉处理软件，其由美国 Cognex 公司开发，重点倾向于模式识别，常搭配其硬件销售，在工业领域应用广泛。NI Vision，其由美国 NI 公司开发，提供完整的机器视觉算法库，并具有良好的调试环境，搭配 NI 的其他软件平台，可以实现机器视觉的快速实现，在测试领域应用较广。Open CV，其是一种开源的跨平台的机器视觉库，拥有丰富的视觉算法，并可以运行在 Linux, Windows, Mac OS 操作系统上，并提供了 Python, Matlab 等语言接口，在研究领域具有较广的用户群。

基于以上的机器视觉算法，是无法直接测量产品 LED 透射的亮度值的，因为数字量的值无法和亮度单位直接匹配在一起，国内外通常对亮度的测量均适用亮度计，比如 LumiCam1300 就是一款亮度色度计，其可以测量 0.0001 至 100000 cd/m² 的亮度，以及可以对色度进行分析。但由于其为专用的实验设备，二次集成非常困难，并且价格非常昂贵，不适合在生产现场使用。而使用机器视觉进行亮度检测的相关研究非常少，而本文基于低成本考虑，提出了一种以灰度值匹配亮度值的方式进行参数传递检测，可以有效对亮度值进行范围检测，大大降低了生产成本。

而对于颜色检测，由于相机对于白色的定义会因为环境光的变化而出现偏移，使拍摄的数字图片带有色温而无法反映正确的白色，尤其在待测品是白光 LED 的时候尤其重要。在一些相机系统中自带白平衡算法，在国内外的大量文献中也有关于自动白平衡的算法^[6]，本文基于 NI Vision 的白平衡功能，提出了一种在实际生产中可以实现简单标定的方法，以实现白平衡的在线标定。

对于物体表面缺陷的研究，在工业领域均有相关的研究。传统图像缺陷处理方式一般使用统计分析法、结构分析法和频谱分析法，并且在国内外的一些研究中对缺陷的动态阈值的设定也经常被提及。比如在无纺织表面缺陷检测中^[7]，扬州大学的邵鑫玉等在对于无纺布生产过程的缺陷进行了分析，提出了基于灰度均值的动态阈值缺陷检测算法。该方法对于固定缺陷样本的检测具有良好的适应性，但算法本身是基于对已知表面缺陷的检测，即通过提取一致背景中缺陷图像的区域，设置合理的预制，与标准图像结合的对比检测法。其待检测缺陷样本如下图 1.1，处理过程如图 1.2 所示：

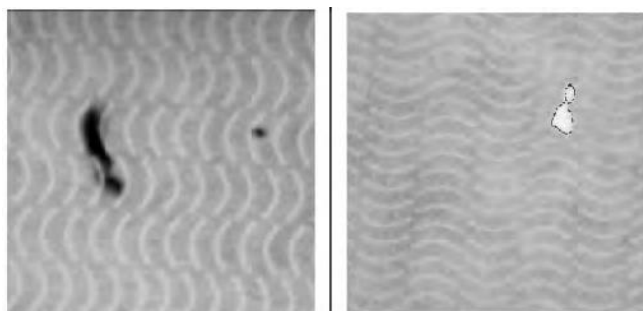


图 1.1 无纺布表面缺陷示意

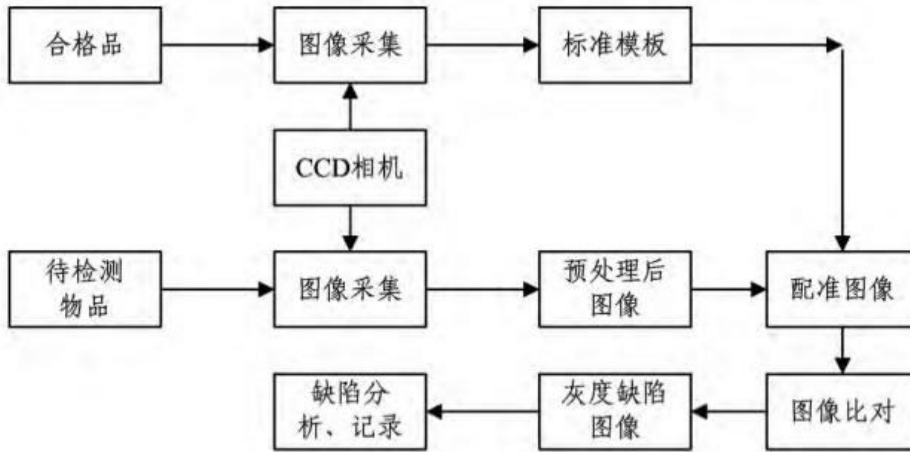


图 1.2 灰度均值的动态阈值检测过程

对于喷涂工业自动检测方面，天津大学的周卫国体使用了图像增强降噪和阈值分割，并使用统计方式进行了特征的抽取和分类。在实际的工业场合实现了一定的光滑面缺陷检测^[8]需求。其待检测缺陷样本如下图 1.3，处理过程如图 1.4 所示：

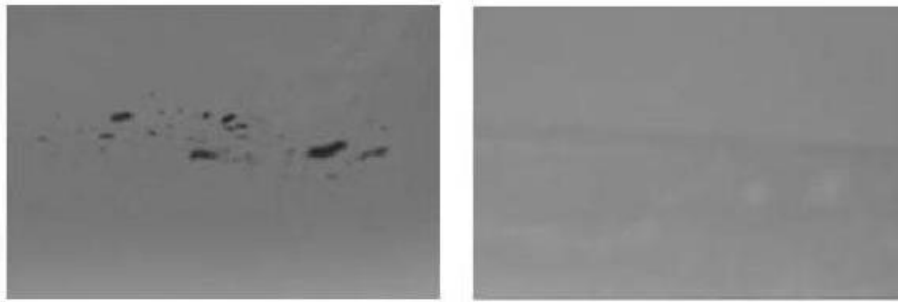


图 1.3 喷涂表面缺陷示意

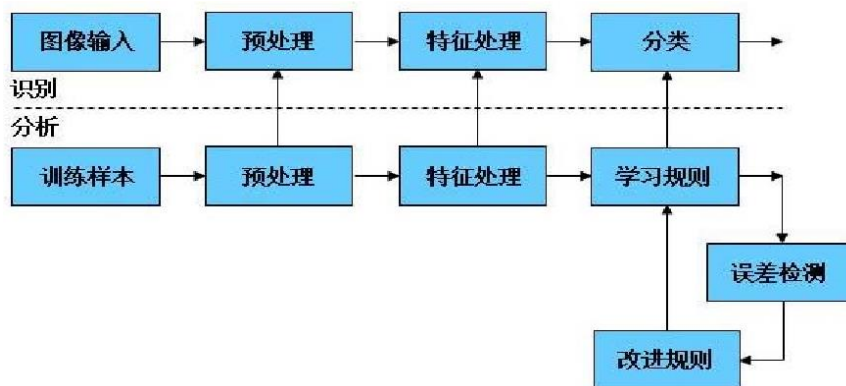


图 1.4 统计缺陷检测过程

而郑健峰发表的带钢表面缺陷检测方法研究中，提及了使用 BP（Back Propagation）神经网络进行特征值的提取，使缺陷检测的灵活性大大增强，使模型

的识别率提高。但是单纯的 BP 神经网络却有其自身的缺陷，BP 的自身分类器只能识别初始定义的几种缺陷模式，没有自我学习功能，新的缺陷模式无法被识别，需要对模型进行重建，这样会耗费大量的时间。并且在处理图像数据时容易陷入局部最小值，在复杂缺陷的工业领域无法推广应用^[9]。其待检测缺陷样本如下图 1.5，处理过程如图 1.6 所示：

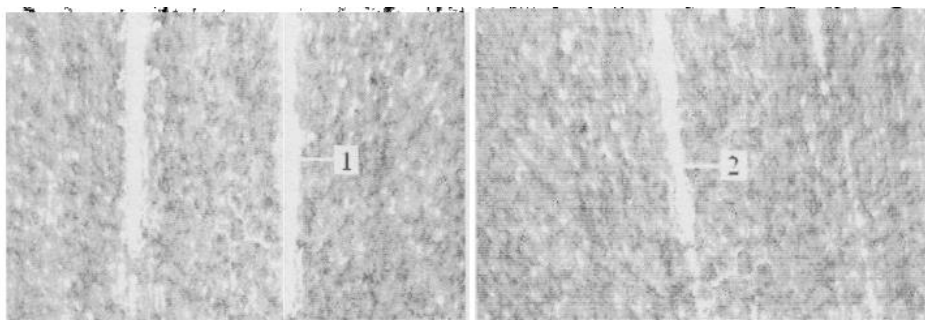


图 1.5 带钢表面缺陷示意

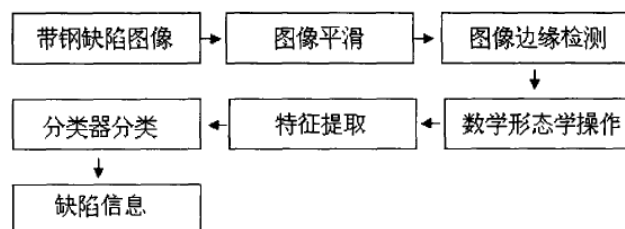


图 1.6 带钢缺陷处理过程

而在 2016 年燕山大学的张秀玲等人在针对带钢板形的缺陷检测研究中使用了 DBN(Deep Brief Network)模型，如图 1.7 所示。使其模型的识别率优于单纯的 BP 网络模型，但其同样具有无法在线调整模型的问题^[10]。

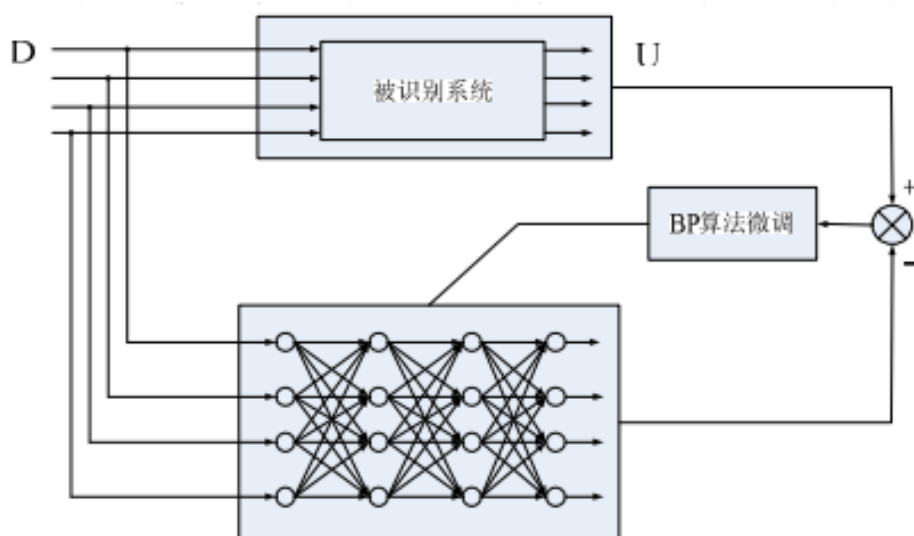


图 1.7 DBN 板形模式识别结构图

对于机器学习领域，国外的相关研究比较早。机器学习经历了几次起伏，伴随着 2006 年，Hinton 等人提出了深度置信网络^{[11][12]}，机器学习又掀起了一阵新的浪潮。Hinton 等人基于 MNIST database (Modified National Institute of Standards and Technology database)，使用 DBN 模型进行了手写字符的识别验证，字符如图 1.8 所示，提出了一种 DBN 混合神经网络的概念，如图 1.9 所示：

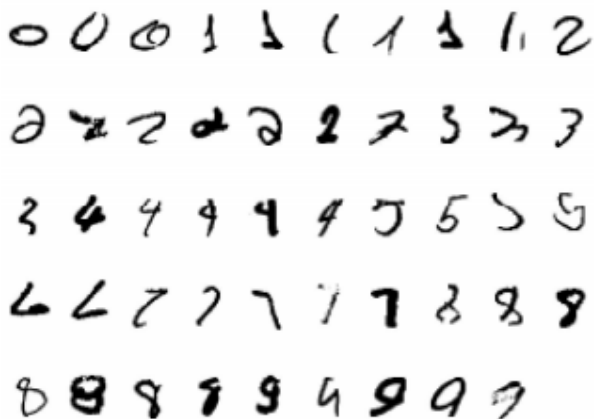


图 1.8 MNIST 手写数字样本

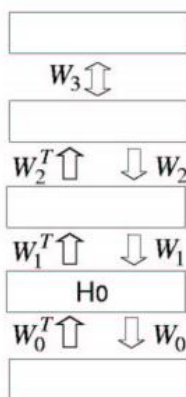


图 1.9 DBN 神经网络

随着这些年的发展，各大科技公司均在深度学习上投入了巨大的人力。比如 Google 的 TensorFlow 就是一款针对机器学习的平台软件，另外还有 Facebook 的 Torch 深度架构，Bonsai 的人工智能平台，以及微软，百度，阿里巴巴均推出了自己的人工智能系统。并且近年来，各种国际会议也对深度学习有了充足的关注度，比如 NIPS 会议讨论了关于无监督特征学习的深度学习结构，ICASSP 会议在 2013 年讨论了语音识别与深度学习的结合应用^[13]。由此可见，深度学习越来越引起人们的关注并对工业界产生深远的影响^[14]。

深度置信网络可以看作是多个限制波尔兹曼机的叠加，在特征抽取方面相比有监督的神经网络有很大的优势，广泛应用于音频，图像和文本识别领域。深度学习旨在使机器学习能够更加的接近最初的目标，人工智能。机器学习的发展大概分为

两个阶段，浅层学习阶段和深层学习阶段。在浅层学习中比较常见的是支持向量机 SVM (Support Vector Machines) [15]，浅层学习可以处理一些简单的信号，但在处理语音和图像的时候就受到了限制。到了 1974 年，反向传播 BP 网络提供了一种反向优化系统参数的方法，但在增加神经网络层数的时候，由于参数无法传递到更深的层数而使其容易陷入局部最优状态^{[16][17]}。直到使用了叠加 RBM 的 DBN 的提出，使参数调优过程可以影响深度学习的每一层的参数，从而使系统模型可以适应音频和图像的复杂处理需求^[18]。而在线优化算法对于模型参数的动态优化和学习率的优化提供了方法。2010 年，Duchi 等提出了自适应的全参数学习率策略 AadGrad^[19]，该方法为深度学习过程中每一个参数单独设计一个学习率，并利用梯度的平方和保证学习率的下降趋势，该方法首次提出全参数学习率策略，为深度学习模型的快速收敛提供了一个很好的解决思路。另一种在线学习算法 Follow-The-Regularized-Leader (FTRL) Proximal^[20]在 2011 年 Google 提出后，在广告领域发挥了巨大的作用，此算法基于 OGD(online gradient descent)，但其对稀疏性进行了优化。

1.3 主要研究内容及拟关键技术

基于实际检测条件和国内外的研究现状，本文的目标为设计一种结合传统机器视觉以及深度学习网络的车载内饰人机交互面板开关信标的视觉检测系统。针对此设计目标，本文的主要研究工作内容如下：

- 1) 对需求进一步分析，并对系统架构进行整体设计和选型。
- 2) 研究使用工业相机进行亮度和颜色检测的方法。
- 3) 研究使用深度学习模型和在线优化算法对信标表面缺陷的检测方法。
- 4) 按照检测功能需求和建模过程，设计软件系统。
- 5) 对测试样本进行实际执行实验，验证系统可靠性。
- 6) 总结系统设计和执行结果，发现不足，提出改进目标。

其中涉及到的关键技术是亮度范围标定研究，颜色检测标定研究，图像采集和处理技术研究，深度学习以及在线优化算法研究，NI Vision 和 Open CV 视觉算法库研究，TensorFlow 系统研究。

1.4 小结

本节主要描述了论文的课题来源以及国内外的研究现状，分析了系统设计的可行性以及现有技术的不足，并列出了本课题的研究工作重点以及需要研究的关键技

术，本文将沿主要研究工作内容的顺序进行分章节展开。

第 2 章 视觉检测系统设计

2.1 视觉检测需求分析

现有车载人机交互类零配件主要集中在中控台开关、排档开关、大灯开关、电动窗开关以及天窗开关等汽车内饰部位，如图 2.1 所示。这些人机交互面板开关有一个共同的特点，那就是表面具有大量的可透射出特定颜色亮度的信标，有色信标用于给驾驶人员提供清晰的灯光指示，颜色一般有白、红、橙、蓝、绿等。人机交互开关的设计通常采用 PCBA(Printed Circuit Board Assembly)表面贴片 LED 工艺，搭配树脂导光条，将 LED 发射出来的光导至表面材料的光刻处，呈现出均匀的不同颜色的信标效果。在现代汽车内饰件中大量采用了此类配件，他们的质量直接影响了汽车驾乘人员的感受。因此，对于所有信标透射出颜色亮度需要达到均匀一致的效果。



图 2.1 车载人机交互类零配件

本文基于人机交互面板开关中的单个天窗开关为例，进行视觉检测系统的设计。天窗开关的实际如下图 2.2 所示：



图 2.2 实际测量产品

根据天窗开关的设计方式，通常此类零配件所采取的工艺工程如图 2.3 所示。

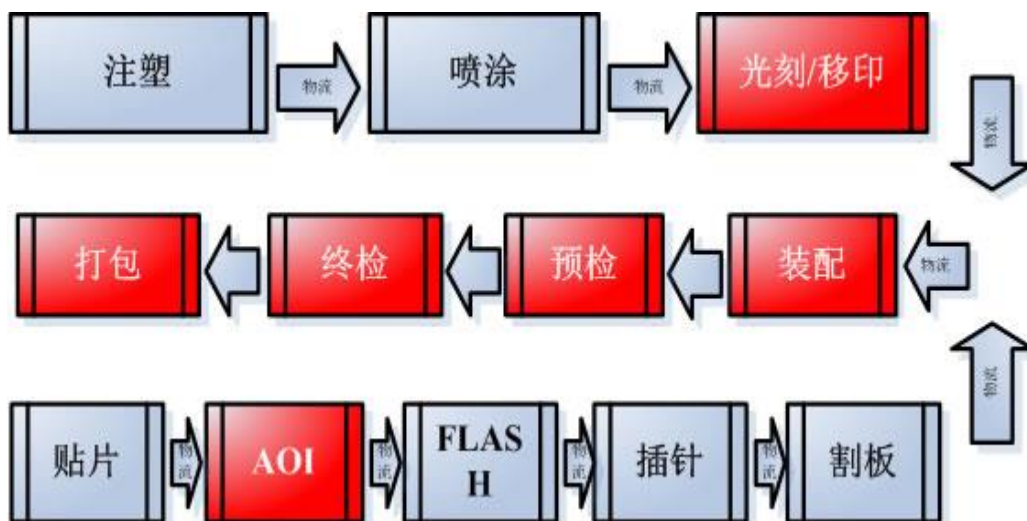


图 2.3 汽车人机交互面板零配件工艺过程

共分为三个过程，第一层过程为壳体表面件的工艺过程，包含零件注塑、表面喷涂以及光刻或移印的过程。零件注塑是将塑料粒子在注塑机种压塑成型；表面喷涂是将塑料表面涂上不同的涂料，使其呈现不同颜色和光泽效果；光刻移印则是在喷涂好的图层上进行加工处理，刻蚀或转印出不同效果的图标。第三层为电子工艺工程，分为贴片、AOI（Automatic Optic Inspection）检测、Flash 程序烧写、插针、割板。而第二层为最终的装配过程，分为装配、预检、终检、产品下线打包四个主要过程。在这些主要的工艺过程节点中，每一道工序均会影响最终的视觉效果，所以需要在各个控制节点对生产过程进行控制，涉及视觉检测的工艺过程主要集中在光刻移印检测、AOI 检测、装配、预检、终检、以及产品打包几个过程中。本设计主要解决信标的亮度、颜色判别和缺陷检测，属于预检工序。可以影响亮度颜色和缺陷的因素很多，比如喷涂与光刻质量和导光条的安装因素，在壳体表面件工艺过程中的光刻工位，一般会解决光刻质量方面的轮廓、尺寸、旋转、波浪、重影、污点、缺失、位置信息。做法有两种，一种是人工透明膜对比法，这种方法是使用一个透明的模板罩在光刻好的零件上，进行目视识别。另一种方法是使用传统的机器视觉进行定量检测，无论哪种方法均无法做到 100%的防错效果。部分缺陷面板件会流转至装配工艺工程中。而在装配过程中，导光条的安装以及注油工艺直接决定了最终的效果是否满足要求。因此在现有系统中，终检工位的操作工承担了产品的目视亮度颜色检测以及缺陷检测，如图 2.4 所示。



图 2.4 现有的人工目检工位

天窗开关信标的主要的异常模式如下：

1) 亮度偏暗或偏亮，如图 2.5。



图 2.5 亮度异常信标示意

2) 颜色使用错误，如图 2.6。



图 2.6 不同颜色信标示意

3) 暗点、缺角、亮度不均、条纹异常，如图 2.7。



图 2.7 缺陷信标示意

现有终检工位的缺陷检测是采用人眼进行识别的。识别方式分为两种，一种是使用暗盒，使待测件和参考样件在较暗的环境中同时进行上电操作，使用人眼对比两边的信标显示是否一致；另一种是将预检工位的图像保存下来，传输至终检工位由工人根据显示器上的图片进行缺陷识别。这两种方式对缺陷的识别标准均由操作工人经过培训后的经验作为判断依据，缺陷由现场质量人员根据曾经发生和预测会发生的缺陷模式，对工人进行培训。现有的培训模式是基于 NBS 色差板来进行的，如图 2.8 所示。



工人识别度培训板

图 2.8 工人训练色差板

但人工的培训模式具有一定的局限性，首先对于培训的效果来说，每个工人的人眼感受敏感度是不同的，对于不同缺陷模式的感觉也不尽相同。

再次，从系统稳定性角度来讲，人是会受精力、情绪的影响出现极大的不稳定性情况。这种不稳定性在 100% 的质量控制过程中是非常致命的，会导致一定量的不良件流出。所以在产线中需要使用机器视觉的方式对零件表面质量进行测量。

在对亮度、颜色和缺陷这三个目标检测项中，通过传统的机器视觉技术可以把亮度和颜色检测出来。亮度这个参数可以通过 CCD (Charge-coupled Device) 的反馈灰度值进行测量，颜色可以通过输出的 HSI (Hue Saturation Intensity) 的颜色分量进行测量，但缺陷因其复杂性却无法通过简单的机器视觉算法实现，仅可通过人眼的方式进行判断。如何使用机器视觉来进行缺陷检测，那就必须引入深度学习算法进行对未知缺陷的特征分析。车载面板开关的表面光刻线条简单，为均匀宽度刻蚀

纹理，除刻蚀纹理外，表面为黑色喷涂材质，基本可以按照黑色背景进行滤除。在检测样本固定以后，对于正常面板信标基本可以确定图像的一致特征。无缺陷的样本在灰阶图下基本无差异，而有缺陷的产品在进行图像处理，可以呈现细微的局部差异，而这些差异可以作为特征值，适合使用深度学习的一种深度置信网络算法（DBN）进行特征提取，在 DBN 的最上层还可以加入传统的带标签的神经网络来对缺陷件和正常件进行区分。另外针对使用场合上，需要在装配流水线上进行部署，所以对于新的缺陷以及参数的优化无法将模型进行离线重建，所以在线优化算法可以被引入来实现单一样本参数优化。

基于此，本次设计的视觉检测系统目标是使用机器视觉自动检测代替人工目检，以提高生产效率和产品质量。

2.2 系统设计分析

整体系统执行架构设计和所用技术如下图 2.9 所示。

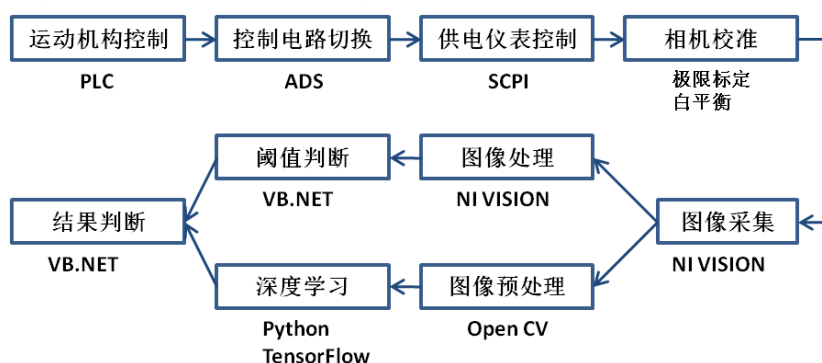


图 2.9 系统执行架构

整体系统包含控制系统，机器视觉系统以及深度学习系统三大块。

1) 控制系统

控制系统的作用是实现待测品自动到达测量工位，并对其进行自动上电，使其满足视觉测量条件。因此，需要进行运动机构控制，电气电路切换控制以及供电仪表控制。其中涉及的技术包含可编程逻辑控制器 PLC(Programmable Logic Controller)设计，切换矩阵的先进设计系统 ADS (Advanced Design System) 设计以及供电仪表控制的可编程仪器标准命令 SCPI(Standard Commands for Programmable Instruments)设计。

2) 机器视觉系统

机器视觉系统的作用是进行图像采集，并对亮度和颜色进行判别。对机器视觉系统的设计，需要进行相机校准，图像采集以及图像处理和阈值判断。所涉及和需

要研究的技术包含极限参数匹配，白平衡校准，基于美国国家仪器公司的 NI Vision 视觉算法开发以及基于微软的 Visual Basic.NET 程序开发。

3) 深度学习系统

深度学习系统的作用是对待测品图像进行特征提取以及对待测品的缺陷模式进行识别。对于深度学习系统的设计，需要进行图像预处理和深度学习建模。涉及和需要研究的主要技术包含 Open CV 视觉系统开发，基于 TensorFlow 的 Python 程序开发。

2.3 系统设计

2.3.1 系统拓扑设计

根据此影像测试需求，需要构建现场的设备环境，网络拓扑图如图 2.10 所示。分为两部分，交换机左侧为测量终端，交换机右侧为模型计算服务器，这个系统采用星型拓扑。在系统最前端为工业相机与暗箱夹具，通过相机将采集到的图片传输给终端电脑，再由终端电脑将原始图片发送至模型服务器进行运算。

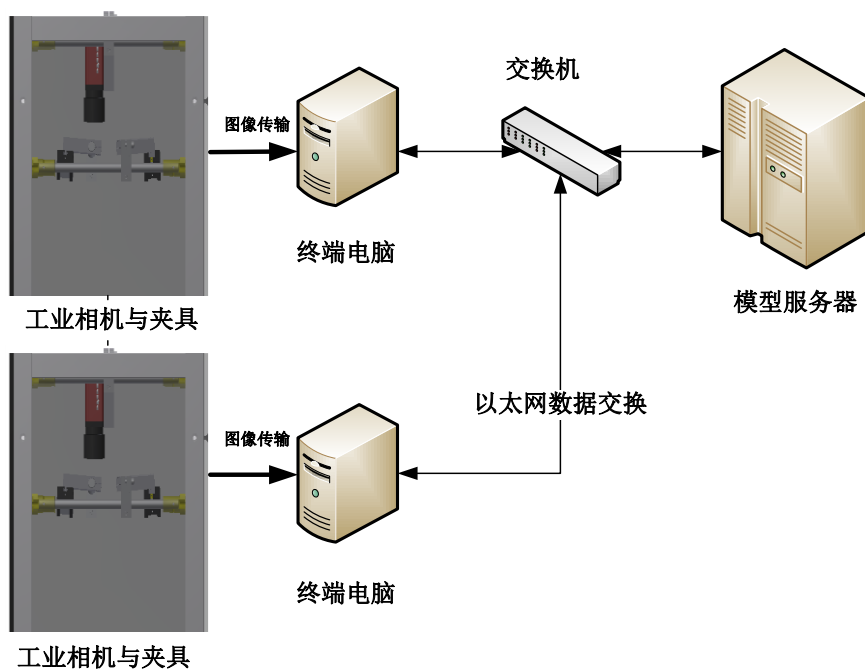


图 2.10 系统拓扑架构

机器视觉系统的设计实现是在终端电脑上完成的，而深度学习系统的设计实现是在模型服务器完成的，这两部分将在第三章节展开研究。

2.3.2 系统整体设计

系统的整体设计如下图 2.11 所示，整个系统包含以下几个部分。

- 1) 终端计算机，主要负责视觉测量部分的颜色和亮度测试。
- 2) 服务器，主要负责缺陷检测，与终端计算机通过交换机相连。
- 3) PLC，主要负责整个系统的机械运动部件的执行和信号反馈。
- 4) 相机系统，主要包含工业相机、镜头、光源以及适配器，负责图像采集。
- 5) 电源和切换模组，主要负责待测品的供电以及电路切换。

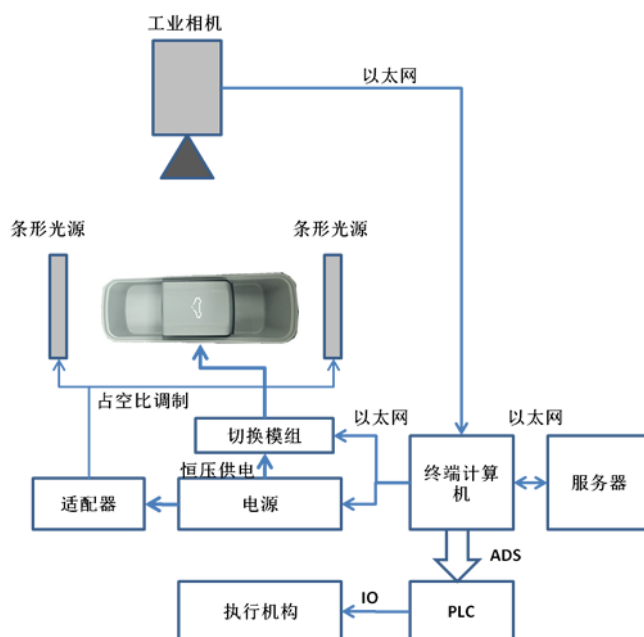


图 2.11 系统整体设计

2.3.3 终端计算机设计选型

对于终端系统，使用研华工业计算机 UNO-3382G 作为整个系统的核心部分，搭配第 4 代 Intel i7 CPU，并具有两个千兆以太网接口，其作用是负责控制电源、切换模组的供电起停，给待测产品和辅助光源供电以及接收工业相机的图像进行亮度判断，同时将采集的图像发送至服务器。系统采用 KEYSIGHT E3644 作为系统供电，这是一款最大输出功率为 80 瓦的单路直流电源，可以稳定输出 12V DC 来点亮产品，以及驱动光源适配器和切换模组。适配器的作用是通过输入电源得 PWM 调制，对条形光源上的 LED 阵列进行调光。切换模组使用继电器排组进行测试电路的切换，它的作用是使单路电源的输出扩展至多路待测产品的引脚上去，以实现不同信标的测试，本文使用的样本为单个单色的天窗开关，所以对于切换模组仅使用开路、闭路即可。条形光源的作用是可以对产品在未点亮之前进行图标匹

配，识别其是否为正确的生产序列产品，以及对信标进行初步的定位和筛查。而在此系统中最关键的工业相机，其作用是拍摄产品信标图片。工业相机采用 Manta G-504C 工业相机，其具有 500 万像素彩色 2/3 CCD Sony ICX655 传感器，分辨率 2452*2056，兼容 GigE 以太网传输接口，搭配 Computar M3514-MP 镜头，安装距离 400mm。采集终端服务器采用 ADLINK MXC6201D 嵌入式工业计算机，i7 处理器，4G 内存，500G SATA 机械式硬盘，Windows 7 操作系统。软件系统采用 Visual Studio 2015 的 VB.NET 软件编程环境，安装 NI Vision Acquisition Software 以及 NI Vision Development Module Run-Time License。

2.3.4 服务器设计选型

而对于服务器系统的设计是此系统的关键部分，服务器作用是承担整个网络传输回来的图片的判断作用，学习模型在此设备上运行，所以对机器的性能有所要求，因为 TensorFlow 平台中的一些算法需要使用 GPU 运算，所以为了保证学习和判断效率，需要在普通工业服务器的基础上增加一张显卡以提高 GPU 运算能力。具体配置如下：Advantech IPC-610H 标准机箱，500W 电源，PCE7128G2 CPU 板搭配 INTEL E3-1275V3 型 3.5GHz，四核 64bit CPU，16G 内存，2TB 硬盘，Leadtek Nvidia Quadro K620 显卡，显存容量 2GB，显存位宽 128bit。对于交换系统设计，采用 CISCO SG200-26 千兆以太网交换机，其具备非常稳定的全双工数据通讯功能。且数据吞吐量能到达每秒 1000Mbit 的双向传输速率，非常适合图像文件的传输。

2.3.5 PLC 设计选型

整体系统的机械动作执行全部由 PLC 负责控制，选用具有 1GHz Intel CPU 的 Beckhoff CX1020 系列 PLC 模块，搭载 Windows CE 操作系统以及 TwinCAT2 实时控制软件。

2.3.6 相机与镜头设计选型

对于相机的选型首先需要考虑的是测量精度，通常情况下，待测图像区域在 100mm*60mm 的范围内，信标单边宽度为 0.5mm，单边宽度上最少需要有 10 个像素点来表示才能比较精确的反映实际图形。因此，选用较大像素的相机是首先需要考虑的。其次，由于本方案中需要对颜色进行识别判断，所以需要彩色 CCD 相机。再次对于通讯采用千兆以太网接口，所以选用支持 GigE 的相机种类。

实际的选型过程比较复杂，由于相机和镜头是搭配使用的，因此在选型中在相

机的基本确定选择范围后，使用以下步骤进行最终确认。如下图 2.12 所示。

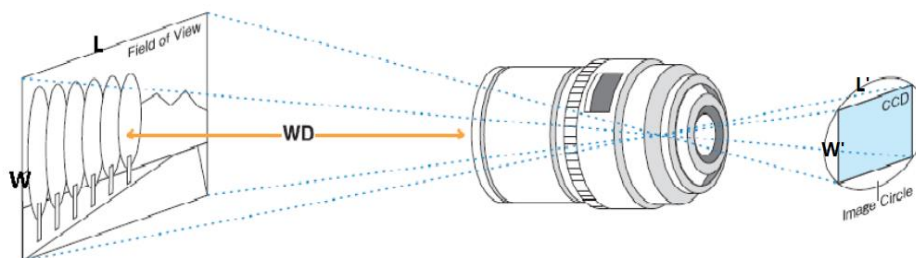


图 2.12 镜头成像原理图

假设 f 为镜头焦距， L 、 W 分别表示视野长度和宽度， L' 、 W' 分别表示 CCD 的长度和宽度。 WD 表示工作距离。则计算过程如下。

1) 视野长宽判定：根据实际输入，判定视野长宽关系，如果输入 $L < W$ ，则在程序中将 L 、 W 交换数值。

2) 设定 CCD 尺寸：程序根据选择的相机型号，设定对应的 CCD 尺寸 L' 、 W' 以及像素数。例如 Manta G504C，则程序设定 $L' = 8.8\text{mm}$ ， $W' = 6.6\text{mm}$ ， L' 上像素数 2452， W' 上像素数 2056。

3) 计算长宽比：如果视野长宽比 L/W 大于 CCD 长宽比 L'/W' ，则程序以 $f = L' * \frac{WD}{L}$ 计算；否则程序以 $f = W' * \frac{WD}{W}$ 计算。

4) 计算理论焦距：根据第 c 步选择的结果进行计算，例如 $L=100$ ， $W=60$ ， $L'=8.8\text{mm}$ ， $W' = 6.6\text{mm}$ ， $WD=400\text{mm}$ ，则计算得到理论焦距 35.2mm。

5) 推荐镜头型号：由于镜头型号是固定的，不能是任意焦距，目前主要使用的有 8，12，25，35，50，75 这些定焦镜头。因此程序会根据计算的理论焦距，推荐最接近的可购买的镜头型号，上述例子推荐 M3514-MP。

6) 计算实际视野：镜头型号选定以后，这时候视野范围已经变化，要反过来求使用推荐镜头的视野范围 $L = L' * \frac{WD}{f}$ ； $W = W' * \frac{WD}{f}$ 。上述例子，则实际视野范围 $L=100.57\text{mm}$ ， $W=75.4\text{mm}$ ，这与最初的输入 100mm*60mm 是有区别的。

7) 计算实际精度：由于需要的是选定镜头之后的实际检测精度，因此精度需要根据实际视野计算。精度等于 L/CCD 长度上像素数。上述例子，精度 $=100.57\text{mm}/2452\text{Pixel}=0.04\text{mm}/\text{Pixel}$ 。

使用软件对上述过程进行编程，在此平台上分别选择两款常用的工业相机 AVT GUPPY PRO F125C 和 AVT MANTA G504C 进行对比，结果如下图 2.13：



图 2.13 相机和镜头选型软件界面

由图中可以看出，只有选择 AVT MANTA G504C 搭配 COMPUTAR M3514-MP 可以满足测试精度^[21]。

2.4 核心软件简介

2.4.1 机器视觉软件简介

此次设计一共涉及两款机器视觉处理软件，一款是 NI Vision，另一款是 Open CV，分别用在终端计算机和服务服务器上，执行图像处理的功能。

NI Vision 是美国国家仪器公司针对机器视觉领域推出的一款可以二次嵌入开发的一个视觉软件包。包含三个主要功能模块，视觉应用辅助功能，图像处理以及机器视觉。视觉应用辅助功能中主要包含文件管理、图像采集、图像显示、图像校准以及图像访问。而图像处理部分则主要包含分析、滤波和图像处理函数，如二值化、腐蚀和数值变换等功能函数。机器视觉模块则可以处理区域选择，坐标变换，几何测量等功能。NI Vision 配置方便快捷，功能库模块丰富且实现简单，因此适合在终端计算机平台上快速生成测试序列。

Open CV 是一款开放式的机器视觉功能库集合，全称为 Open Source Computer Vision Library，是一个开源发行的跨平台的计算机视觉库，可以运行在 Linux、Windows、Android 和 Mac OS 操作系统上。其具有 C、C#、Python、JAVA 等多种语言库接口，提供大量的机器视觉功能函数，具备图像采集、图像滤波、数值变换、图像分析、图像处理以及图像传递和保存功能。适合在服务器端使用 Python 程序调取 Open CV 功能库进行图像的预处理工作。

2.4.2 TensorFlow

TensorFlow 是由 Google 开发的一种用于机器学习运算的开源算法库。其采用的是数据流图的方式，可以使软件工程师非常迅速的部署机器学习算法和模型。数据流图非常容易识别，如下图 2.14 所示：

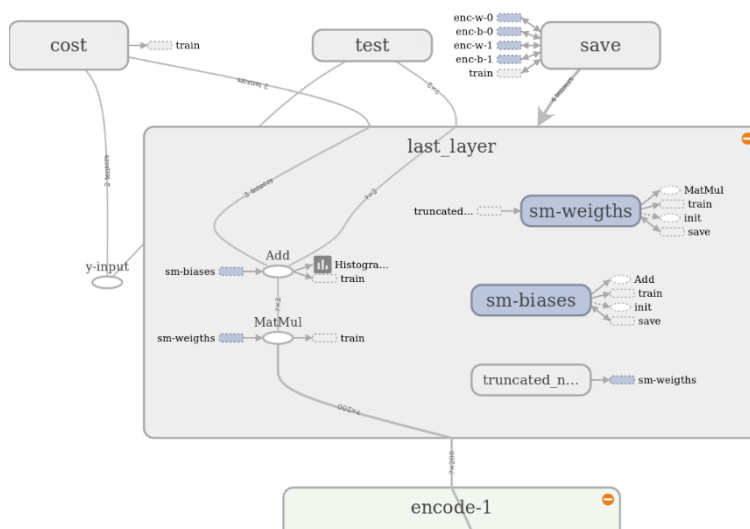


图 2.14 TensorFlow 流图示意

节点 (Nodes) 在图中表示具体的数学操作，图中的线条 (edges) 则表示在节点之间相互联系的数据数组，即张量 (tensor)。TensorFlow 支持多种平台，可以在计算机、移动终端和大型云服务器上运行，可以灵活的配置多 CPU 和 GPU 协同运行而不需要修改代码。在配置好所有的节点以后，所有的节点算法执行将被分配至不同的计算机资源中去进行异步并行运算。TensorFlow 主要提供了神经网络库，但其也具备作为其他数学运算的承载平台，使用简单的数据流图来构建数学模型。TensorFlow 提供 Python, C++ 接口，可以允许用户在 TensorFlow 的架构下进行二次开发，编写上层 Python 应用库，同时可以使用 C++ 对底层库进行添加。TensorFlow 提供了大量的神经网络的算法库，可以使开发者将一个想法快速的实现，大大缩短了从概念到应用的时间，例如，TensorFlow 直接集成了在线优化算法 SGD、ADAGRAD、FTRL，使整个验证在线学习的过程非常简单。TensorFlow 具有自动求微分的能力，因此在机器学习中需要大量求梯度的过程将会非常容易实现。你只需要建立模型结构，将模型结构和目标函数结合在一起，并添加相应的输入端数据，Tensorflow 将自动完成所有计算。计算某个变量相对于其他变量的导数可以通过扩展你的图来完成的，这样可以使用户监控程序过程，了解程序执行中的细节变化。TensorFlow 具有多个管理器，可以实现各种资源的分配调用。TensorFlow 的界面可

以是 C++，也可以使用 Python 的编写界面，当代码构建完毕后，其可以将所有的脚本代码进行整理，并以流程图的方式展现出来。总之，作为一款 2017 年刚刚发布 1.0 正式版本的神经网络平台软件来说，其最大的特点就是简单、图形化、跨平台。

2.4.3 Python

Python 是一种脚本语言，其在 20 世纪 90 年代初被推出，主要用在系统管理任务和 Web 编程领域。Python 的设计理念就是简单易用，语法非常容易理解，并且融合了 C 的一些内容。其特点是简洁、易读以及可扩展性。大量的开源科学计算软件都提供了 Python 的软件接口，比如开源的计算机视觉算法库 Open CV。Python 还具有专用计算扩展库的软件接口，例如 NumPy、SciPy 以及 Matplotlib，它们为 Python 提供了快速数组处理、数值运算以及绘图功能。大量的算法接口支持使其非常适合对工程技术，数据实验，图表分析以及科学计算算法开发。与 MATLAB 相比，Python 有如下优势：

1) MATLAB 是一款商用软件，价格较高。而 Python 属于开放软件，完全免费，大量开源的科学计算库都提供了 Python 的软件接口。安装和使用 Python 以及相应的算法库均为免费的。

2) 与 MATLAB 相比，Python 是一种计算机编程语言，更加灵活，可以让用户自定义各种简单易懂的代码，方便后续维护。

3) MATLAB 是一款完整功能的专注于科学和工程计算的专用软件，而 Python 除了具有大部分 MATLAB 的功能以外，还可以进行网络交互，界面设计等扩展功能，可以灵活的实现各种高级任务。

当然，Python 的劣势就是需要编写代码。而 MATLAB 相对更加简单。

Python 在执行过程如下：首先会将.py 文件中的源代码编译成 Python 的 byte code（字节码），其次再由 Python Virtual Machine（Python 虚拟机）来执行这些编译好的 byte code。使用虚拟机来执行编译好的代码，这种方式 and Java 和 .NET 中的类似，但其抽象程度更高，更加灵活。另外，Python 本身是跨平台的，可以支持 Linux、Windows 以及 Mac OS，并且可以直接使用命令模式进行交互执行。这点上是 .NET 所无法比拟的。

2.5 小结

本章节结合产品设计方式以及生产工艺工程，对实际生产过程中的产品质量控制所需要的检测项目进行了分析，并在系统设计方面得出结论，需要考虑进行三个

检测项，分别为颜色检测、亮度检测和缺陷检测，做并尝试分析了引入深度学习来进行零件表面缺陷检测的可行性。依据需求对系统进行了设计，并对硬件选型和关键软件进行了详细介绍。

第3章 机器视觉系统设计

针对人机交互面板的颜色和亮度检测来讲，其亮度区域主要集中在喷涂表面的光刻区域，因此，对待测品的亮度颜色检测只需要将光刻信标本身作为分析区域即可，无需对整个待测品的表面进行分析。由于相机所感应的为 RGB (Red, Green, Blue) 三分量，不方便进行颜色数据处理，因此，需要进行 HSI 色彩空间变换。而在相机使用之前，需要对其进行硬件标定才可适用，以消除畸变、对比度、白平衡等因素的影响，否则会出现色差等系统偏差，影响颜色的测量值。同时针对亮度在车载面板开关领域中主要关注的是光强 Intensity，但其物理学单位是 cd / m^2 (坎德拉每平方米)，这并不是工业相机所具有的功能，而且通过 HSI 转换后的 Intensity 值虽然反应了亮度值，但其单位是一个数字量，无法与设计要求直接匹配，所以需要对其进行软件标定，使其满足亮度检测要求。

3.1 检测过程

整体的亮度颜色的检测过程如下图 3.1 所示：



图 3.1 亮度颜色检测过程

如图所示，这个过程分为五个部分。硬件标定，主要针对相机的硬件状态进行标定，包含相机 x, y, z 三方向位置，镜头光圈，焦距调节位置，颜色失真状态标定；软件标定，主要针对亮度参数匹配进行软件范围值得标定；点亮产品，作用是使待测品在放光的条件下进行测量；测量，调用软件功能块进行图像处理以及亮度颜色测量；结果输出，将结果反馈给程序主界面，并上传数据库与整线控制系统，将对应的产品按不良品进行质量管控。

而其中针对亮度和颜色的测量，主要的过程如下图 3.2 所示：

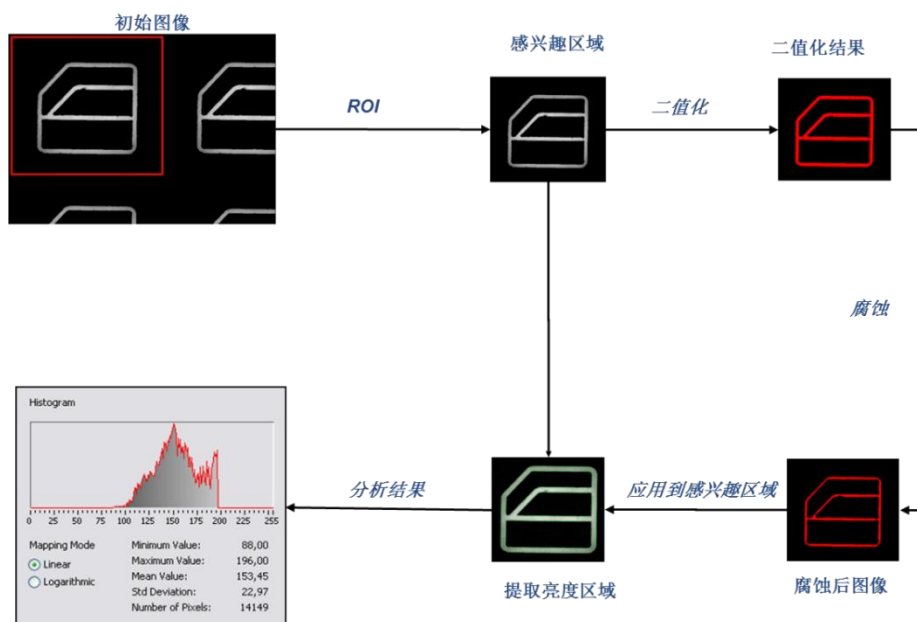


图 3.2 亮度颜色检测过程

整体过程包含：采集图片；进行颜色模型转换，将 RGB 转换为 HSI，提取 H 为颜色信息，按照范围设定进行判断；提取 I 分量作为计算亮度信息的灰度图原图；进行信标定位，以灰度图的信标状态为模板，用基本灰度模式匹配的方式获取感兴趣信标区域 ROI；进行图片二值化，二值化过程需要设置阈值参数，参数的设置需要满足足够多的像素数量用于计算亮度值，判定依据是下极限样件；为得到清晰的图像边界，对图片腐蚀处理；再次提取信标；得到信标的测量值方图，则可以得出信标亮度均值，标准差及像素个数等信息；根据得出的亮度均值与上下灰度极限值做对比，得出判断结果。

3.2 色彩空间

颜色和亮度检测必须对色彩空间进行分析，因为相机采集来的图像通常只要 RGB 三个分量，并不具有亮度属性，所以需要将 RGB 三分量转化到色彩空间进行分析，常见的色彩空间有 LAB, HSL, HSI 色彩空间^[22]，而与人类感知最为接近的是 HSI 色彩空间，因为本文中涉及检测的车载面板信标是人机交互的一个指示灯光，是供人眼进行识别的，因此，采用与人类感知接近的 HSI 色彩空间是最符合检测要求的。

人眼可以分辨一个物体颜色的三种变化，分别是色调、饱和度、亮度，而 HSI 的三分量正好与之对应。

1) H: Hue 色调，其与光的波长有关，而不同波长的光呈现出来的是不同的颜色，比如红色、蓝色、黄色，当然也可以表示为暖色调和冷色调。

2) S: Saturation 饱和度，其表示颜色的纯度，单一波长颜色的饱和度最高，颜色最鲜艳，而加入白光，则会使饱和度下降。

3) I: Intensity 亮度，其表示图像的亮度和灰度。是人眼敏感度最高的一个分量。

HSI 锥形色彩空间模型如图 3.3 所示：

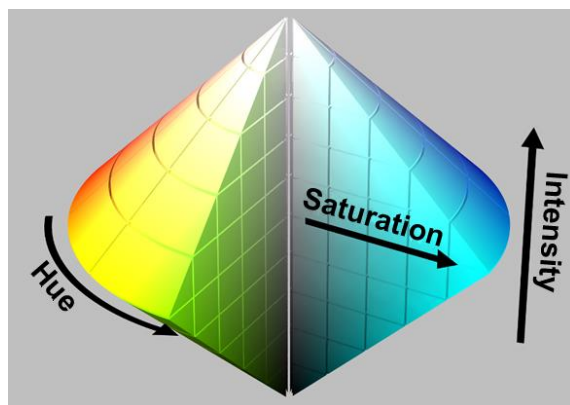


图 3.3 HSI 色彩空间锥形图

通过将 RGB 转化为 HSI，简化了计算机的处理图像数据的难度，而 RGB 与 HSI 之间可以通过公式直接转换的。本文采用的是一种常用转化算法，公式如下：

$$H = \begin{cases} \cos^{-1} \left(\frac{(R-G)+(R-B)}{2\sqrt{(R-G)^2+(R-B)(G-B)}} \right), & G \geq B \\ 2\pi - \cos^{-1} \left(\frac{(R-G)+(R-B)}{2\sqrt{(R-G)^2+(R-B)(G-B)}} \right), & G < B \end{cases} \quad (3.1)$$

$$S = 1 - \frac{3\min(R, G, B)}{R+G+B} \quad (3.2)$$

$$I = \frac{R+G+B}{3} \quad (3.3)$$

以上的转化均是针对单个像素点的，整个图形的转化即为依次转化各个像素点得出最终的 HSI 像素矩阵。

3.3 硬件标定

相机系统在运行之前需要做标定才可进行测量。本方案中主要涉及以下三类标定，分别为参数标定、标定板标定和白平衡标定。

3.3.1 参数标定

根据极限样件调整曝光时间及光圈状态以及相机感光度，保证最亮及最暗的情况下都能在影像系统最佳状态下工作。参数对相机的影响如下图 3.4 所示：

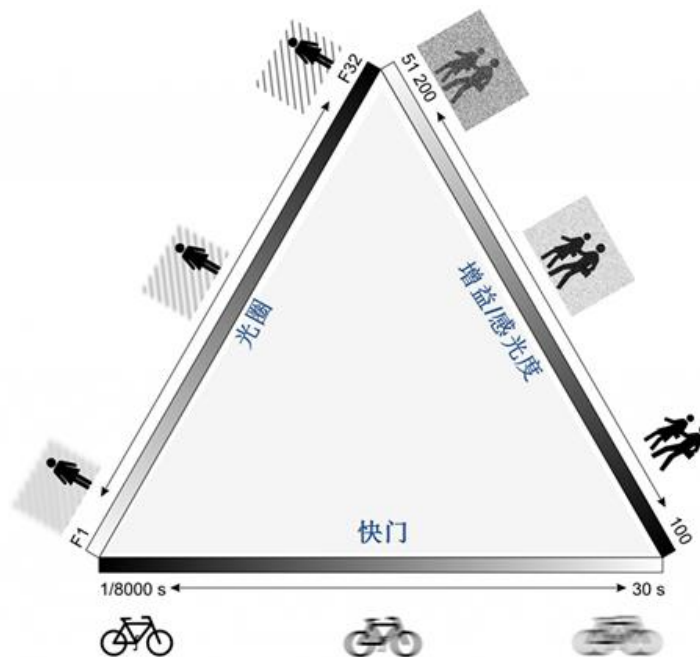


图 3.4 相机主要参数效果示意图

快门即曝光时间，可以看到，随着曝光时间的增加，对于动态物体会出现比较明显的拖影，由于在快门时间内物体映射到 CCD 区域的位置有明显变化造成的。而光圈则会影响景深，即相机感应的轴向区域范围，值越大，则景深越深，值越小，则景深越浅。增益和感光度会影响成像的质量，当感光度太大时，背景噪点会大量增加，会提高图片处理的难度，感光度太小，则图片偏暗。因此设置合适的对后续的图像处理极为重要。

3.3.2 标定板标定

主要是针对由第一步标定的初始参数下的相机特性进行进一步校准。标定块如图 3.5 所示。标定块可以实现对比度、明暗度、颜色、焦距、几何尺寸的标定。



图 3.5 相机标定块

3.3.3 白平衡标定

使用标准白光，或者是客户指定的白色定义对影像系统的白色基准进行标定，确保后续对白色 LED 信标颜色检测的准确度。如图所示，当相机未进行白平衡校准时，会出现颜色值的偏移，例如白色的信标会偏红色，如图 3.6。

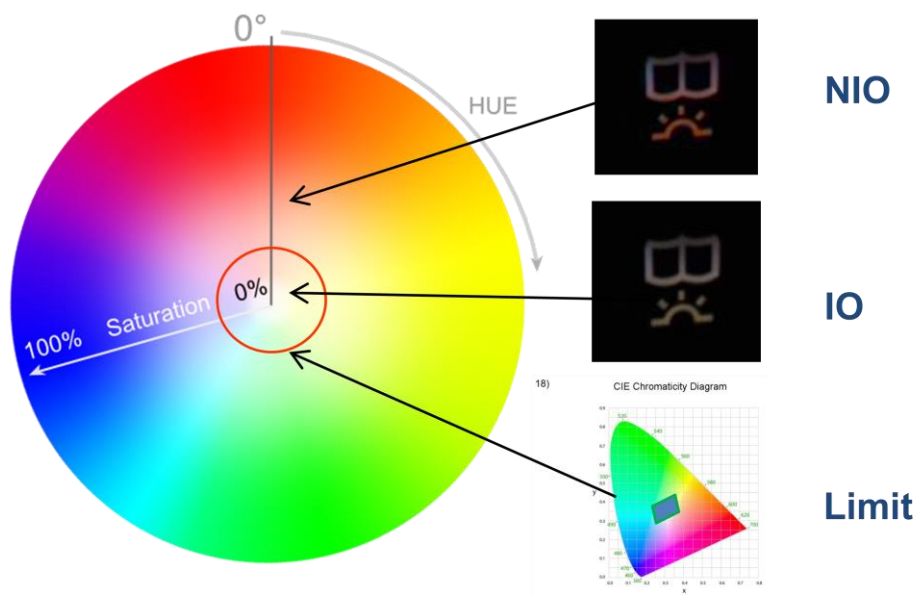


图 3.6 白平衡不良图片对比图

白平衡需要做的是将偏移的颜色区域调整到正确的色彩空间区域。原理如下图 3.7 所示：

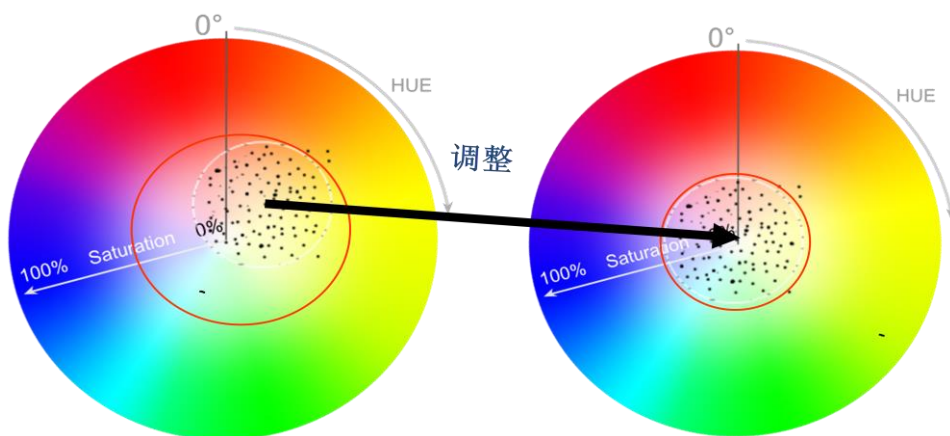


图 3.7 白平衡调整原理示意

在实际的标定中本方案使用的是一款平板电脑，首先需要对屏幕本身在光学实验室中使用专业仪器进行标定^[23]。平板如图 3.8 所示：



图 3.8 光学标定工具的预标定

对相机的白平衡标定过程如下：

- 1) 确定客户白光色坐标参数。
- 2) 对光学标定平板进行实验室预标定。
- 3) 使用客户色坐标对光学标定工具进行设置。
- 4) 使用白平衡标定软件进行色彩预估，如图 3.9。

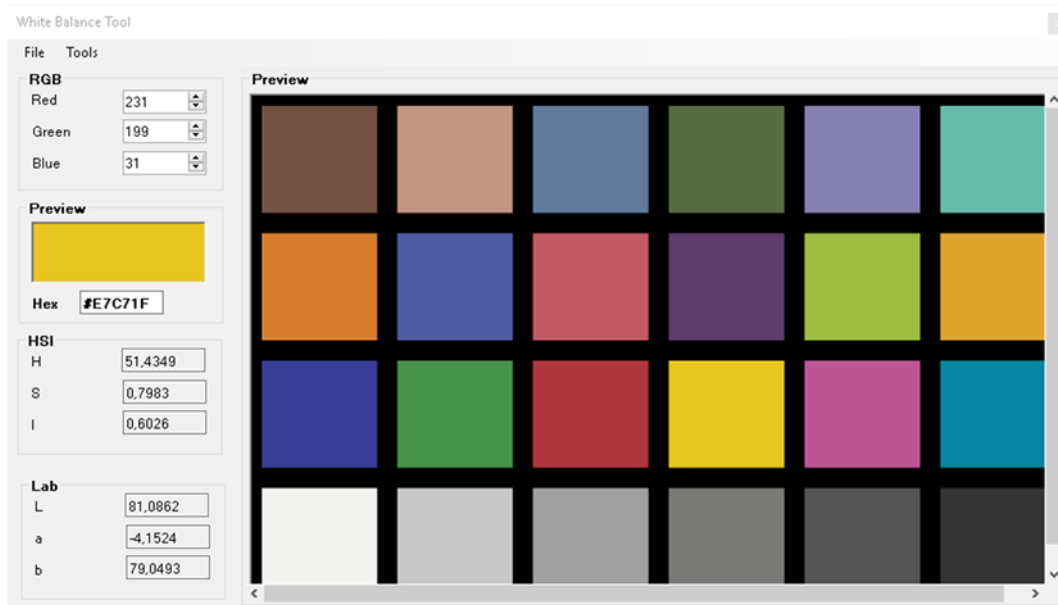


图 3.9 白平衡标定软件界面

- 5) 将参数传递给相机。

经过白平衡标定,对于客户色坐标白色的样件来说,色调和饱和度均降低为 0,而对亮度的几乎没有影响。

3.4 软件标定

极限样件是针对亮度的范围设定的,由于在整张图像的亮度 I 取出后,对信标区域的亮度值即可以发光区域的平均亮度来表示。但由于 I 值是一种灰度表示,其取值从 0 到 255,而亮度是的单位表示为 cd/m^2 ,坎德拉每平方米,这两个度量单

位是无法直接进行映射的，那么如何找到一种方法进行单位传递，是解决此问题的关键。

对于 CCD 相机来讲，不同区域的 CCD 晶元对于亮度 I 值的响应是不一样的，当使用同一个标准光源进行测量的时候，在空间区域的取 9 个区域点，如图 3.10 所示：

1n	4n	7n
2n	5n	8n
3n	6n	9n

图表 3.10 相机校准区域定义

在同样是 20 的阈值设置下，得出差异较大的不同的亮度值，且响应曲线并不重合，假设 x 轴表示不同等级的亮度， y 轴表示不同的灰度等级。如图 3.11 所示：

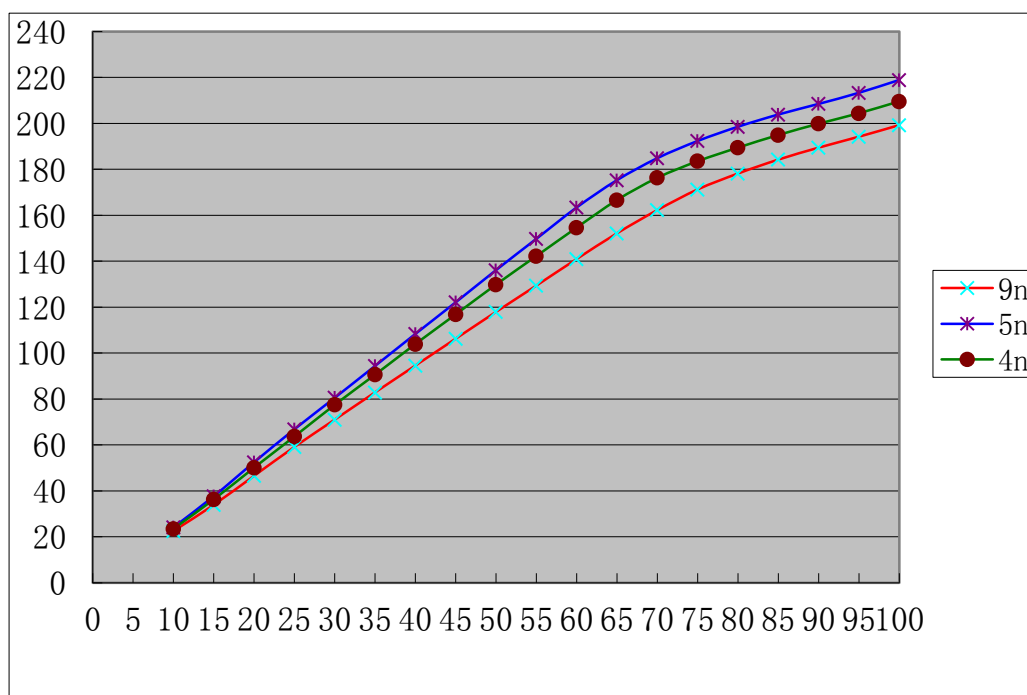


图 3.11 不同位置的亮度-灰度关系曲线

这里面的影响因素主要有 CCD 的均匀度，镜头的影响，以及标准光源的方向误差，从上图还可以看到另外一个 CCD 的感应特性，在 160 灰度值以下基本上保持为一阶线性关系，从 160 至 255 的灰度范围出现了曲率。

而对于不同阈值的设置下， I 响应曲线如下图 3.12 所示：

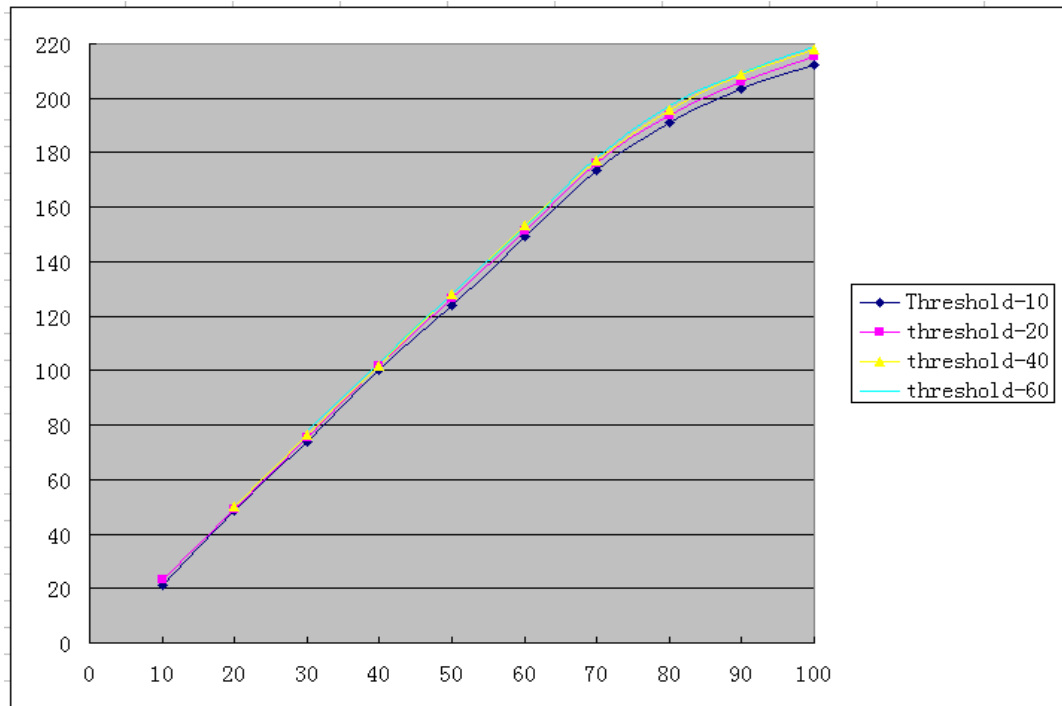


图 3.12 不同阈值的亮度-灰度关系曲线

同样可以看到，阈值的设置并不影响 CCD 的亮度相应曲线在 160 以下保持 1 阶线性，而在 160 以上出现曲率的特性。

由于此特性，无法通过一个标准光源对整个 CCD 的亮度响应曲线进行预估，且 CCD 对不同的颜色的敏感度也是不一样的，所以，最好的办法就是使用和产品同样形状和颜色的样件对特定区域进行测量，以得到参考值。

假设待测品映射到 CCD 的区域在 $4n$ 位置，当阈值设置为 20 的响应曲线如下图所示，取亮度等级下极限值 25，上极限为 57，则其对应的灰度值下极限为 70，上极限为 148，可以看出整个响应曲线的导数始终为正，没有拐点出现，即可说明当亮度持续增大的情况下，灰度值一定是持续增大的，反之亦然，如图 3.13。

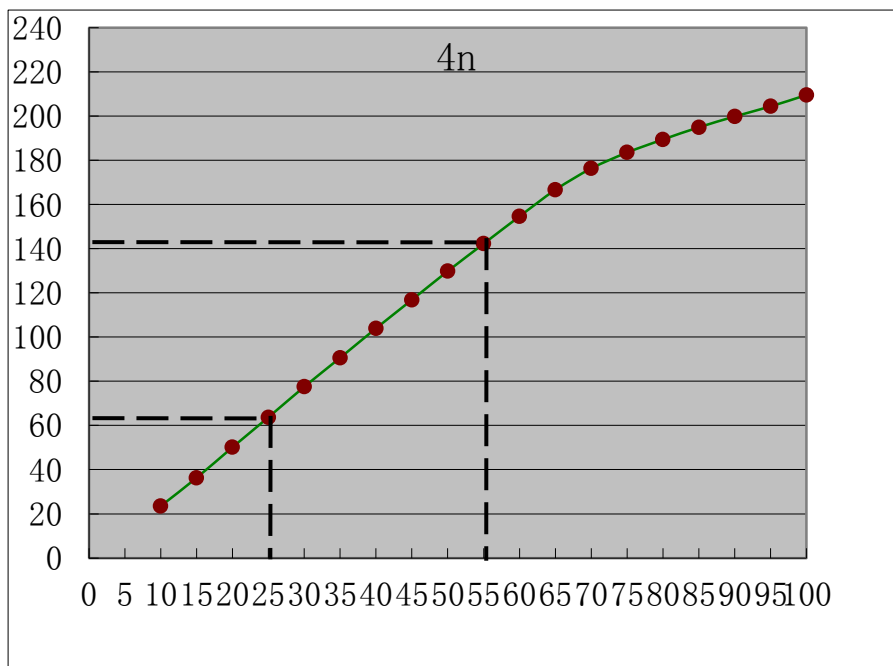


图 3.13 亮度与灰度的极限关系

因此取不同的亮度值的两个样件，这两个样件基于产品设计亮度规范进行制作上下极限样件。将上下极限样件放置在产线视觉系统的固定位置，即可得到各自对应的上下灰度极限，则此上下灰度极限即可作为后续的亮度参数设置来测量实际的产品亮度。

3.5 小结

本章对信标的亮度和颜色进行了系统分析和设计，给出了实际的检测过程。介绍了 HSI 色彩空间模型，以及给出了选择此模型的原因，提出了一种通过极限样件在生产现场实现亮度测试的可能，并给出了详细的亮度极限样件实现原理以及相机系统标定的工程模型。

第 4 章 缺陷检测系统模型设计

由于对缺陷检测方面无法通过简单的阈值设置进行处理，所以缺陷检测不能仅通过传统的视觉处理进行缺陷件的筛查。在进行基于深度学习的缺陷检测的模型设计时，为使模型更有利于计算机识别，在进行深度模型建模之前需要对图像进行预处理。

4.1 图像预处理过程

图像的采样和预处理是样本能够进行深度学习的前提条件。在所有过程中最前端的采样过程，对整个系统的整体稳定性起到至关重要的作用。其中比较关键的参数是图像的分辨率。由于在工业应用环境中，不同面积大小的产品需要使用同一套相机系统来进行测量，所以对相机的来说，在无法实现动态调整位置与焦距的前提下，必须使用足够大的 CCD 像素点阵才能满足产品上微小信标的检测精度要求。所以对于此固定安装方式的系统来讲，需要使用 500W 像素 CCD 相机 2452*2096。

本文以一个天窗信标为例，信标长边尺寸 10mm，单边宽度 0.5mm。步骤如下：

1) 采样后使用 ROI (Region of Interest) 进行粗定位，选取 256*256 正方形区域框对图片进行定位切割，形成 256*256 的正方形原始图片，这样可以保证信标单边宽度不少于 10 个像素点，测量精度大于 0.05mm。

2) 在整个图像处理过程中，由于图形本身具有的特性为 RGB 原图，即每个像素点中均包含 RGB 三个分量，这样是不匹配 RBM 的样本输入要求的，因此需要对 RGB 图像进行灰阶转换，使其变为维度较低的灰阶值矩阵，由于处理的目标信标为单一颜色样本，因此对其进行灰阶转换不会影响最终的缺陷检测结果。灰阶转换的一个经典公式如公式 (4.1)：

$$\text{Gray} = R \times 0.299 + G \times 0.587 + B \times 0.114 \quad (4.1)$$

在 OPENCV 中使用 GRB2GRAY 函数对样本图像的所有像素点做灰度值转换^[47]。处理完成的图片如图 4.1 所示。

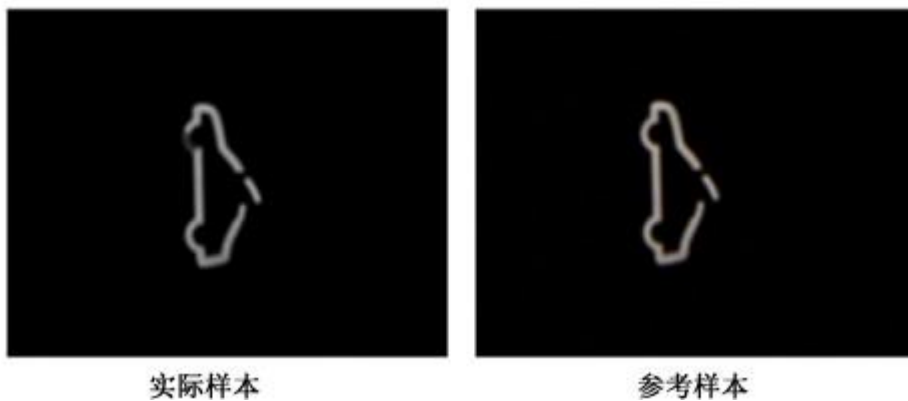


图 4.1 灰度处理的实际样本与参考样本

3) 图像定位

图形定位使用 PATTAN MATCH^{[24][25][26]}功能，将实际样本的中心位置与参考样本的中心位置对其重合。实际执行效果如下图 4.2 所示：

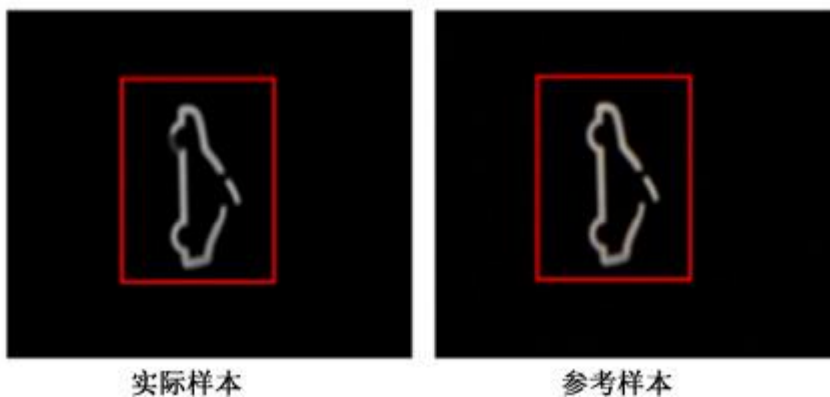


图 4.2 图形定位

4) 像素填充

由于中心定位会对样本的边界有所调整，会出现空白坐标区域，所以对于空缺的部分需要进行像素填充，是其形成一个完整的 256*256 像素矩阵空间。填充好的图片如下图 4.3 所示：



图 4.3 像素填充后的图片

5) 残差处理

对实际样本做分析，可以看出对于实际样本与参考样本在差异方面还是比较细微的，实际样本的汽车样式的前轮部分有偏暗缺陷发生，如果使用原始图像做深度学习，会出现收敛缓慢的情况，特征区分不明显，为避免此情况发生，需要对使用实际样本的像素矩阵与参考样本的像素矩阵相减，得出一个残差二维矩阵，假设样本图像为 a ，实际样本的图像为 b ，则计算公式(4.2):

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} - \begin{bmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{m1} & \cdots & b_{mn} \end{bmatrix} = \begin{bmatrix} |a_{11} - b_{11}| & \cdots & |a_{1n} - b_{1n}| \\ \vdots & \ddots & \vdots \\ |a_{m1} - b_{m1}| & \cdots & |a_{mn} - b_{mn}| \end{bmatrix} \quad (4.2)$$

残差图像如图 4.4 所示:



图 4.4 残差图片

6) 腐蚀处理

为突出残差处理过后的实际图像，消除图像噪点，需要对残差图像进行腐蚀处理，进一步突出差异，使用 `cvErode` 函数进行处理，腐蚀过后缺陷亮斑的边缘更加清晰，消除毛边和一些过渡灰阶的影响^[27]。实际效果如图 4.5 所示:



图 4.5 腐蚀后效果

7) Gabor 滤波:

腐蚀处理后的图片已经和周边的黑色背景清晰的隔离开，但是区别还不够明显，为了使最终的数据在深度学习网络中更容易区分出特征，需要对腐蚀后的图片进行 Gabor 滤波处理，使像素矩阵的二值化特征更加明显，更有利于特征提取。

Gabor 滤波是由傅里叶变换引申而来的，在图像数字化处理方面，Gabor 滤波与人类的视觉系统中简单细胞在受刺激时的响应极为相似，其在提取目标物体的特征信息非常具有优势，且在频率域分析方面具有良好的特性。Gabor 滤波对图像的边缘敏感，但对于图像的光照变化不敏感，所以对图片亮度的变化有极好的适应性。因此对于一些纹理也可以很好的区分。Gabor 滤波函数^[28]是一种线性滤波器，通常用于边缘提取。在空间域中，二维的 Gabor 器可以看做是由正弦平面波进行调制的高斯核函数。也可以理解为使用一个正弦平面波乘以高斯函数，其中的乘法具有卷积特性。Gabor 滤波器是由实部和虚部组成的，实部和虚部相互正交。

Gabor 滤波器的复数表示公式如下：

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp(i(2\pi \frac{x'}{\lambda} + \psi)) \quad (4.3)$$

实部表示公式如下：

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos(2\pi \frac{x'}{\lambda} + \psi) \quad (4.4)$$

虚部表示公式如下：

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \sin(2\pi \frac{x'}{\lambda} + \psi) \quad (4.5)$$

其中 λ , θ , ψ , σ , γ 是五个设置参数。 λ 表示波长，它是以像素为单位指定其实际值，一般大于等于我，但不大于图像尺寸的五分之一； θ 表示方向，此参数表示 Gabor 函数平行条纹的方向，取值范围从 0 到 360 度； ψ 表示相位偏移，取值范围从-180 度至 180 度。 σ 表示高斯因子的标准差，其只随带宽变化，假设 b 表示带宽，则 b , σ , λ 的关系公式如下：

$$\frac{\sigma}{\lambda} = \frac{1}{\pi} \sqrt{\frac{\ln 2}{2}} \times \frac{2^{b+1}}{2^b - 1} \quad (4.6)$$

带宽一般为正实数，通常设置为 1，则 $\sigma = 0.56 \lambda$ 。 b 越小， σ 越大，则 Gabor 滤波后的图形的中德显示形状越大，条纹数量越多。 γ 表示长宽比，其关系到 Gabor 滤波后图形的纵横比，，可以理解为椭圆率，当 $\gamma=1$ 时，为正圆形，当 $\gamma<1$ 时，形状会沿平行条纹的方向延伸。一般设置为 0.5。

本文使用实部表示公式进行 Gabor 滤波处理^[24]，并将最终的参数设置为 $\lambda=5$ ， $\theta=0$ ， $\psi=0$ ， $b=1$ ， $\gamma=0.5$ ，则处理过的图形如图 4.6 所示：



图 4.6 腐蚀后效果

8) 归一化

Gabor 滤波之后即可进行归一化处理，这也是将样本转化为深度学习模型的最后一步，归一化是深度学习的一项基础工作，由于不同的评价体系通常具有不同的评价指标和单位，不同量纲的参数是无法进行数据分析的，因为必须将数据变化至 0 到 1 以内，从而可以使不同量纲的数据放入一个模型中进行综合评估。归一化方法一般有以下两种：

a) min-max 标准化法，这种方法又成为离差标准化，其是对原始数据的线性变换，可以将数据的转换值映射到 0 和 1 之间。函数如下：

$$y = (x - \min) / (\max - \min) \quad (4.7)$$

其中，max 表示整体样本中的最大值，min 表示样本中的最小值，在图像的像素矩阵中，可能出现的 max 为 255，可能出现的 min 为 0。

b) Z-score 标准化方法

这种方法使用数据集合的均值 μ 和标准差 σ 进行数据的归一化，经过处理的数据将符合标准正态分布，即均值为 0，标准差为 1。函数如下：

$$y = (x - \mu) / \sigma \quad (4.8)$$

本文使用较为简单的 min-max 标准化法进行处理。处理后的图像如图所示：



图 4.7 归一化后效果

归一化后，由于对比度较小，从图像样本来看，几乎无法分辨，但信息均储存在像素矩阵中。

4.2 深度学习模型建模

4.2.1 RBM 限制波尔兹曼机

在深度学习的，当前应用较多的 RBM 算法是从 BM 算法衍生而来。BM 算法全称为 Boltzmann machine，中文简称波尔兹曼机^[29]。它是 Hinton 等人在 Sejnowski

在 20 世纪 80 年代中期提出的一种随机递归神经网络。波尔兹曼机是早期几种可以进行对数据进行学习并可以解决复杂问题的人工神经网络之一，其概率分布类似于波尔兹曼分布，所以命名为波尔兹曼机。在这种神经网络中，状态输出由二值化神经元表示，每个神经元仅输出两种状态，1 或者 0。状态 1 表示接通，状态 2 表示断开。所以波尔兹曼机也是一种对称耦合的随机反馈型二值化神经网络。

就 BM 网络来讲，每个神经节点均可能具有二值化状态中的一个，假设其网络结构使用向量表示 $S = (s_1, s_2, \dots, s_i, \dots, s_N)$ ， s_i 表示向量中第 i 个神经元节点的状态， s_j 表示向两中第 j 个神经元节点的状态， N 为节点总数。那么 s_i 属于 $\{0, 1\}$ 的网络能量函数为如下公式表示：

$$E = -\sum_{i=1}^N \sum_{j=1, i \neq j}^N w_{ij} s_i s_j + \sum_{i=1}^N \theta_i s_i \quad (4.9)$$

上述公式中 w_{ij} 为神经元节点 i, j 之间的连接权值， θ_i 是神经元 i 的偏置，因此可以得出神经元节点 i 在产生 0 或者 1 的能量差值为如下公式：

$$\Delta E_i = E_{s_i=0} - E_{s_i=1} = \sum_{j=1}^N w_{ij} s_j - \theta_i \quad (4.10)$$

引入激活函数，则，当 $s_i = 1$ 的概率为如下公式：

$$P_{s_i=1} = \frac{1}{1 + e^{-\Delta E_i/T}} \quad (4.11)$$

上式中 $P_{s_i=1}$ 表示为神经元节点 i 输出状态取 1 的概率值。则对于二值化网络来讲，神经元节点 i 输出状态为 0 的概率使用 1 减去输出为 1 的概率即可得出，公式如下：

$$P_{s_i=0} = 1 - P_{s_i=1} = \frac{e^{-\Delta E_i/T}}{1 + e^{-\Delta E_i/T}} \quad (4.12)$$

从上述公式可以看出对于 $P_{s_i=1}$ 的概率，能量差值 ΔE_i 越大，神经元状态取 1 的概率越大；相反能量差值越小，神经元状态取 0 的概率越大。而温度 T 的变化可改变概率曲线的曲率。如下图 4.8 所示：

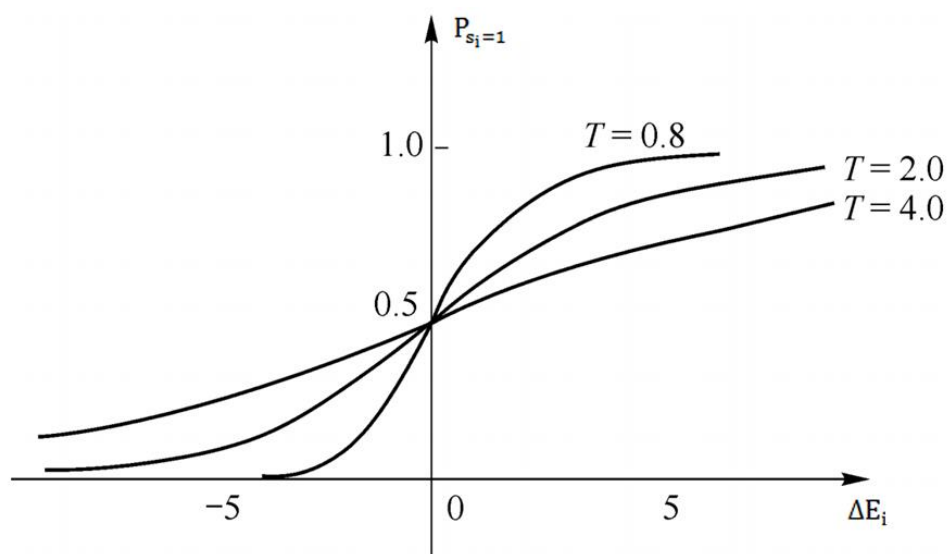


图 4.8 激活函数曲线

从上图中可以看出，当温度 T 比较高时，概率表现出的曲线变化趋于平缓，对于同一输入得到的状态为 0 或 1 的概率值差别不大，整个系统处于活跃状态，系统稳定性变差。而温度低时，曲线的斜率变大，对于同一输入状态为 1 或 0 的概率差别变大；当 $T = 0$ 时，概率函数退化为符号函数，神经元的输出状态将变为固定状态。

对于波尔兹曼机来讲，随着整个网络状态的演变，从概率的意义上网络的能量是朝着减小的方向变化的。然而，虽然网络能量的总趋势是朝着减小的方向变化，但在有些神经元状态可能会按照小概率事件进行反向的取值，从而使网络能量增加。正因为此，才是波尔兹曼机具备了从局部极小值的低谷中爬出去的能力，这种可能性在 **BM** 中是确实存在的，那是由于采用了神经元状态按概率随机取值的工作方式，波尔兹曼具有不断跳出位置较高的低谷来进行寻找位置更低的新低谷的能力。这种运行方式简称为搜索机制，即网络在运行过程中不断地搜索更低的能量极小值，直到达到能量的全局最小。从模拟退火的原理可以看出，温度 T 不断下降可使得网络跳出局部底部的能力由强减弱，这正是保证 **BM** 机能成功搜索到能量全局最小的有效措施。

将神经元的状态为 1 和状态为 0 的概率相除，则如下公式所示：

$$\frac{P_{s_i=0}}{P_{s_i=1}} = e^{-\Delta E/T} = e^{-(E_0-E_1)/T} = \frac{e^{-E_0/T}}{e^{-E_1/T}} \quad (4.13)$$

将上式推广到网络中任意两个状态 α 和 β ，则两个全局状态 α 和 β 出现的概率与对应的能量之间满足如下公式：

$$\frac{P_\alpha}{P_\beta} = e^{-(E_\alpha-E_\beta)/T} \quad (4.14)$$

上式就是著名的波尔兹曼的分布。从公式中可以看出：**BM** 机处于某一状态的概率主要取决于此状态下的能量，能量越低概率越大；**BM** 机处于某一状态的概率还取决于温度参数 T ，温度越高，不同状态出现的概率越近，网络能量较容易跳出局部极小而搜索全局最小，温度越低，不同状态出现的概率差别越大，网络能量较不容易改变，从而可以使得网络搜索收敛。这正是采用模拟退火方法搜索全局最小的原因所在。

BM 机的结构具有可见节点和隐节点。一般可见节点包括输入与输出节点，隐节点成为不可见节点，训练时输入输出节点收集训练样本，而隐节点主要起辅助作用，用来实现输入输出之间的联系，使得训练集能在可见单元再现。**BM** 机连接形式可用如下图 4.9 所示的有向图表示：

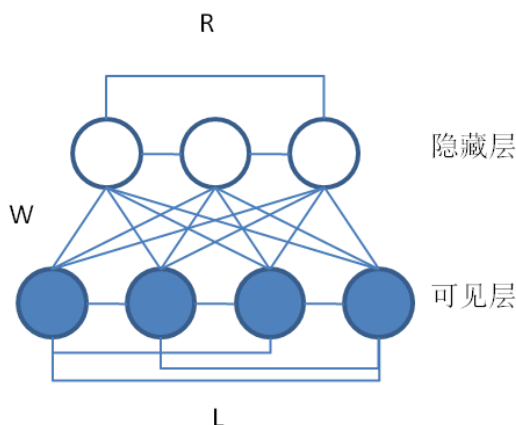


图 4.9 BM 结构

假设隐藏层有 m 个隐藏神经元节点， $h = (h_1, h_2, \dots, h_m)$ ，可见层有 n 个可见神经元节点， $v = (v_1, v_2, \dots, v_n)$ ，则波尔兹曼机的能量函数为如下公式：

$$E(v, h) = - \sum_{i=1}^n \sum_{j=1}^m w_{ij} v_i h_j - \frac{1}{2} \sum_{i=1, k \neq i}^n v_i l_{ik} v_k - \frac{1}{2} \sum_{j=1, k \neq j}^m h_j r_{jk} h_k - \sum_{i=1}^n b_i v_i - \sum_{j=1}^m c_j h_j \quad (4.15)$$

上式中 $\{W, L, R, B, A\}$ 均是系统参数， W 表示可见层节点到隐藏层节点的对称连接权值， L 表示可见层节点间的连接权值， R 表示隐藏层节点间的连接权值， B, C 表示可见层节点和隐藏层节点的偏置，通过改变 L, R 的不同设置，可以得到一般 **BM**，半 **RBM** 和 **RBM** 三种不同的拓扑结构。

RBM 限制波尔兹曼机是在波尔兹曼机的基础上衍生出来的，在实际的机器学习过程中，部分随机变量是不可见的，所以需要将 **BM** 中神经元节点分为两部分，分别为可视节点 v ，以及隐藏节点 h ，那么对于可视节点之间是没有什么连接，同样在隐藏节点之间也不存在任何连接，可是节点 v 和隐藏节点 h 之间使用权值 w 进

行连接。由此形成的 BM 即为限制波尔兹曼机 RBM。如图 4.10。

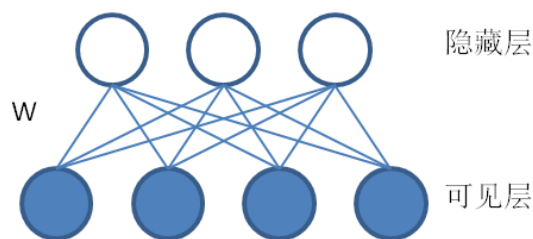


图 4.10 RBM 结构

由于 L, R 均为零, 所以 RBM 能量函数为:

$$E(\mathbf{v}, \mathbf{h}) = -\sum_{i=1}^n \sum_{j=1}^m w_{ij} v_i h_j - \sum_{i=1}^n b_i v_i - \sum_{j=1}^m c_j h_j \quad (4.16)$$

因此, RBM 系统随机变量的概率分布为:

$$P(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} e^{\sum_{i=1}^n \sum_{j=1}^m w_{ij} v_i h_j - \sum_{i=1}^n b_i v_i - \sum_{j=1}^m c_j h_j} \quad (4.17)$$

其中 Z 为归一化因子, 也称为配分函数, Z 的表示公式如下:

$$Z = \sum_{\mathbf{v}, \mathbf{h}} e^{\sum_{i=1}^n \sum_{j=1}^m w_{ij} v_i h_j - \sum_{i=1}^n b_i v_i - \sum_{j=1}^m c_j h_j} \quad (4.18)$$

同样考虑激活函数:

$$\text{sigmoid}(x) = \frac{1}{1+e^{-x}} \quad (4.19)$$

当可见层节点给定值后, 各个隐藏层神经节点的激活状态是相互独立的, 则在 \mathbf{v} 已知条件下 \mathbf{h} 取激活状态的条件概率为:

$$P(h_j = 1|\mathbf{v}) = \text{sigmoid}(c_j + \sum_{i=1}^n v_i w_{ij}) \quad (4.20)$$

由于 RBM 网络在可见层和隐藏层间是对称结构, 所以在 \mathbf{h} 层节点状态给定的情况下, \mathbf{v} 各节点取激活状态, 即概率为 1 的条件概率为:

$$P(v_i = 1|\mathbf{h}) = \text{sigmoid}(b_i + \sum_{j=1}^m h_j w_{ij}) \quad (4.21)$$

这仅仅表示了已知可见层求隐藏层的概率, 以及已知隐藏层求可见层的概率, 但在实际的取值中, 如何让计算机决定是取 0 还是 1 呢, 这需要程序产生一个随机数, 如果它小于条件概率, 则对应的神经元节点取 1, 反之取 0。

整个 RBM 的学习过程就是在寻找使概率最大的一组参数中, 这里需要引入 Kullback-Leibler 差异的概念, 简称 KL, 也称为相关熵, Relative Entropy, 它是用来估算在一个事件空间中的两个概率分布的差异, 假设真实的样本事件空间 Ω 的概率分布为 $P(x)$, 而 $Q(x)$ 为使用采样编码得出的拟合概率分布, 则使用 $D(P||Q)$ 表示 KL 距离, 计算公式如下:

$$D(P||Q) = \sum_{\Omega} P(x) \ln \frac{P(x)}{Q(x)} = \sum_{\Omega} P(x) \ln P(x) - \sum_{\Omega} P(x) \ln Q(x) \quad (4.22)$$

从上式中可以看出, 当真实概率分布与采样拟合概率分布相等时, 即 $P(x)=Q(x)$, 则相关熵为 0, 此时的拟合概率分布极为真实概率分布, 不过就目前的技术而言, 这是不可能发生的, 仅可以无限接近, 因此需要使 D 尽量小, 以使拟合概率分布更加真实, 分析公式可知, 只要是 $\sum_{\Omega} \ln Q(x)$ 最大即可, 在实际的应用中一般取 $\frac{1}{N} \sum_{\Omega} \ln Q(x)$, N 表示样本个数, 使其最大的参数求解就是极大似然估计。

接下来可以使用梯度下降法来求解参数。假设一个样本空间 S , 已知可见层数据集 v , 则使用观测数据集来估计的概率分布的极大似然函数为以下公式:

$$\ln \prod_S P(v) = \sum_S \ln \sum_h P(v, h) = \sum_S \left\{ \ln \sum_h e^{-E(v, h)} - \ln \sum_{x, h} e^{-E(x, h)} \right\} \quad (4.23)$$

那么似然函数关于 w_{ij} , b_i , c_j 的梯度如下公式。推导过程不再描述。

$$\frac{\partial \ln P(v)}{\partial w_{ij}} = v_i P(h_j = 1|v) - \sum_v P(v) P(h_j = 1|v) v_i \quad (4.24)$$

$$\frac{\partial \ln P(v)}{\partial b_i} = v_i - \sum_v P(v) v_i \quad (4.25)$$

$$\frac{\partial \ln P(v)}{\partial c_j} = P(h_j = 1|v) - \sum_v P(v) P(h_j = 1|v) \quad (4.26)$$

由上述三个公式可以看出, 第二项均含有 \sum_v , 这就意味着需要对 v 进行遍历, 这样的计算量非常巨大, 几乎是不可能的。因此, 需要引入 Markov 采样来计算, 处理过程如下: 针对每个训练样本 X , 具有 N 个元素, k 表示第 k 个样本元素, 样本元素均使用一种抽样方式抽取对应样本集合 Y , 且 Y 符合 RBM 网络所表示的 Gibbs 分布, 则可以使用新的样本集合 Y 去估算第二项了。则公式更新如下:

$$\Delta w = \frac{\partial \ln P(v)}{\partial w_{ij}} = v_i P(h_j = 1|v) - \frac{1}{N} \sum_{k=1}^N P(h_j = 1|v_{yk}) v_{ykj} \quad (4.27)$$

$$\Delta b = \frac{\partial \ln P(v)}{\partial b_i} = v_i - \frac{1}{N} \sum_{k=1}^N v_{ykj} \quad (4.28)$$

$$\Delta c = \frac{\partial \ln P(v)}{\partial c_j} = P(h_j = 1|v) - \frac{1}{N} \sum_{k=1}^N P(h_j = 1|v_{yk}) \quad (4.29)$$

这里还需要引入两个概念, 来求得 v_i , 和多项式的第二项。一个是 Gibbs 采样, 另外一个 CD-K 算法^{[12][30]}, 即对比散度算法。

Gibbs 采样, 是一种基于 MCMC 策略的抽样方法, 对于一个 N 维的随机向量样本, $X = (x_1, x_2, \dots, x_N)$, 无法获知其概率分布 $P(X)$, 但可以计算出 X 的其他分量是第 i 个分量 x_i 的条件概率分布 $P(x_i|x_{i-})$, 因此从 X 的任何一个样本状态 $X = (x_1(0), x_2(0), \dots, x_N(0))$ 为起始, 进行迭代抽样计算, 随着抽样次数 n 的增加, 随机向量样本 $X = (x_1(n), x_2(n), \dots, x_N(n))$ 的概率分布将以 n 的几何级数的收敛于 X 的联合概率分布 $P(X)$, 如下图 4.11 所示:

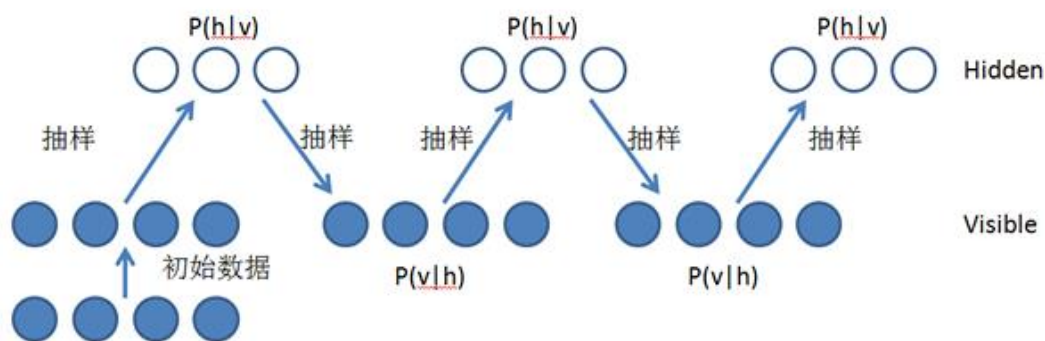


图 4.11 Gibbs 采样过程

在初始数据输入 RBM 后，所有的抽样过程会在可见层和隐藏层间交替进行。则可以得到如下抽样过程：

$$h(0) = P(h|v(0))$$

$$v(1) = P(v|h(0))$$

$$h(1) = P(h|v(1))$$

$$v(2) = P(v|h(1))$$

.....

$$v(k+1) = P(v|h(k))$$

再依据公式(4.20)(4.21)即可代入(4.27)(4.28)(4.29)中计算梯度值，得出更新参数。不过此方法在特征数较多，且可见层神经元节点比较多的是后，在采样步数比较大的情况下，计算比较缓慢，因此需要引入另一个简化算法 CD-K 算法。

CD-K 算法最主要的作用是给出了一个思想，那就是只要较少的 k 步抽样，就可以得到足够好的近似值了。最极端情况下，可以将 k 设置为 1，则使用一个来回的抽样即可。因此在对样本的 k 取为 20 的时候，更新参数模型如下，其中 ϵ 表示学习率：

$$w \leftarrow w + \epsilon(h(0)v(0) - h(20)v(20)) \quad (4.30)$$

$$b \leftarrow b + \epsilon(v(0) - v(20)) \quad (4.31)$$

$$c \leftarrow c + \epsilon(h(0) - h(20)) \quad (4.32)$$

4.2.2 DBN 深度置信网络

深度置信网络 DBN (Deep belief Networks) 可以看作是多个 RBM 的叠加。对于单个 RBM 来讲，仅有一个可见层和一个隐藏层。将多个 RBM 叠加在一起，使下层的 RBM 的隐藏层当作上层 RBM 的可见层，依次向上，理论上可以叠加的层数没有限制，但过多的层数影响计算时间，且有相关论文^[31]指出，随着 RBM 的堆叠层数的增加，对于特定样本，当达到一定数量的层数后，一般层时小于 10，即

会出现互相关系数趋于稳定的情况，互相关系数不再变化，说明进一步增加 RBM 的层数对于深度学习模型的准确率不会再有优化的趋势。

DBN 模型是一种半监督学习模型，其基本结构如下图 4.12 所示：

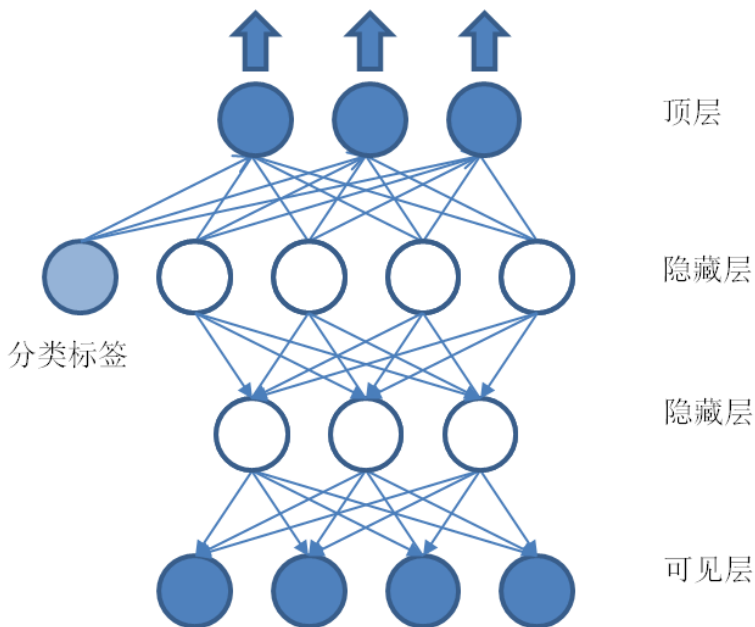


图 4.12 DBN 结构

自下而上，分为可见层一个，隐藏层若干，和一个顶层。靠近可见层的两层隐藏层是有方向性的，在进行特征提取的自下而上的过程中属于无监督学习方式，先计算第一层 RBM 的权值和偏置，计算方法和 4.2.1 章节中所述的 RBM 算法一致。在计算第二层 RBM 的时候，使用第一层 RBM 的隐藏层作为可见层，同样可以计算出第二层 RBM 的权值和偏置。而在顶层与最上层隐藏层组成了一个联合记忆区，且在最上层的隐藏层是可以加入分类标签，进行有监督的学习，对特征值进行归类，再通过 BP 反向传播算法更新下层的权值^[32]。同时由于顶层的联想记忆的存在，所以可以实现最终的样本分类识别。这种 DBN 结构要优于单纯的 BP 神经网络，因为单纯的 BP 神经网络算法是有监督的学习过程有自身的一些缺陷，比如，收敛较慢，容易陷入局部最小。

BP 算法的基本原理如下^[33]：

假定一个多层级联的深度网络如下图 4.13 所示：

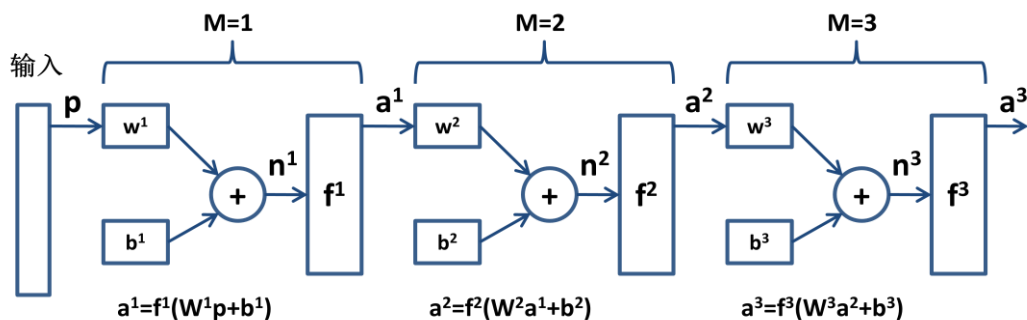


图 4.13 多层级联深度网络

假设 p_q 是多层的网络样本输入, t_q 是样本输入所对应的标签, M 表示层数, a^{m+1} 表示最后一层的输出, W 表示权值矩阵, b 表示偏置, f 为传输函数, f 采用逻辑回归函数形式, 则前向传播公式如下。

$$a^{m+1} = f^{m+1}(W^{m+1}a^m + b^{m+1}), \quad m = 0, 1, \dots, M-1 \quad (4.33)$$

下面来讨论反向传播, 假设 n 为传输函数前的输入, 他是参数的显性函数, f 是 n 的显性函数, 则 f 为参数的隐性函数, 假设 q^{m-1} 表示上一层的节点数。

$$n_i^m = \sum_{j=1}^{q^{m-1}} w_{ij}^m a_j^{m-1} + b_i^m \quad (4.34)$$

因此

$$\frac{\partial n_i^m}{\partial w_{ij}^m} = a_j^{m-1}, \quad \frac{\partial n_i^m}{\partial b_i^m} = 1 \quad (4.35)$$

定义 \hat{F} 为近似均方误差, 则下式表示 \hat{F} 对 m 层输入的第 i 个元素变化的敏感性:

$$s_i^m = \frac{\partial \hat{F}}{\partial n_i^m} \quad (4.36)$$

则:

$$\frac{\partial \hat{F}}{\partial w_{i,j}^m} = s_i^m a_j^{m-1} \quad (4.37)$$

$$\frac{\partial \hat{F}}{\partial b_i^m} = s_i^m \quad (4.38)$$

定义:

$$\hat{F}^m(n^m) = \begin{bmatrix} \hat{f}^m(n_1^m) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \hat{f}^m(n_{q^m}^m) \end{bmatrix} \quad (4.39)$$

则根据链式推导：

$$s^m = \hat{F}^m(n^m)(W^{m+1})^T s^{m+1}, \quad m = M - 1, \dots, 2, 1 \quad (4.40)$$

因为如链式所示，将以下式方式传递，因此使用 BP 网络将可以对整个网络进行反向微调。

$$s^M \rightarrow s^{M-1} \rightarrow \dots \rightarrow s^2 \rightarrow s^1 \quad (4.41)$$

而最后一层的起始点 s^M ，可以用如下式子表示， t 表示目标输出， a 表示实际输出。

$$s^M = -2\hat{F}^M(n^M)(t - a) \quad (4.42)$$

则最终使用近似的最速下降法更新权值和偏置， α 表示学习率， k 表示第迭代次数。

$$W^m(k+1) = W^m(k) - \alpha s^m (a^{m-1})^T \quad (4.43)$$

$$b^m(k+1) = b^m(k) - \alpha s^m \quad (4.44)$$

通常，DBN 训练分为两步：

第 1 步：单独的无监督的训练每一层 RBM，确保最初输入的信息可以保留更多的信息特征。

第 2 步：在 DBN 的顶层可以使用分类器网络^[34]，可以将前一层 RBM 的隐层数据作为输入值，并进行分类器学习，使其成为一种有监督的学习过程。从第一步可以获知，单独训练的 RBM 层的权值仅表示本层的连接权值，可以达到局部最优，但其不是整个 DBN 网络的连接权值，因此，反向传播网络还需要将误差信息从上向下传播至每一层 RBM，微调整个 DBN 网络。

可以看出，单纯的使用 RBM 叠加因其无监督是不能发挥判定作用的，而使用 BP 网络也会因随机初始化权值参数而容易陷入局部最优和训练时间长的缺点^[35-40]。所以需要两者结合起来才可发挥最优效果。

在进行下面两层 RBM 的时候，为了使模型可以更加快速的收敛，可以对样本进行单独额外的预训练，并将生产的权值作为 DBN 网络的初始值^{[31][41][42]}。

4.2.3 深度学习建模过程

核心模型采用深度置信网络模型，由于深度置信网络是由多个限制波尔兹曼机和一个分类器网络组成的，所以对其中的标签分类维度以及 RBM 层数均需要分别进行定义输入。初始参数中标签分类维度为 DBN 的必备输入参数，其标记着对于模型本身输出的性能判别，对于机器学习，分类器需要的是连续的数据，而在本次任务中，对于产品的判断为 PASS 和 FAIL 两种状态，即为 0 和 1 两种状态，但对于 DBN 模型的分器来讲，是不能接受的，尤其在判断状态增多的情况下，数据会产生随机性，无法保持有序状态，所以需要引入一种稀疏编码方式 One-Hot Encoding 独热编码^[43]。独热编码又称一位编码，原理是使用，其方法是使用 N 位状态寄存器来对 N 个状态进行编码，每个状态都有独立的寄存器位，并且在任意时候，其中只有一位有效。

例如：自然状态码为：000，001，010，011，100，101

独热编码为：000001，000010，000100，001000，010000，100000

可以这样理解，对于每一个特征，如果它有 m 个可能值，那么经过独热编码后，就变成了 m 个二元特征。并且，这些特征互斥，每次只有一个激活。因此，数据会变成稀疏的。这样做的好处是解决了分类器不好处理属性数据的问题，并在一定程度上也起到了扩充特征的作用

本文采用的是二分类，分别使用 01、10 表示两个分类，即合格判断和不合格判断。

从图像处理阶段生成的归一化后的 256*256 二值化图像，选取 4000 张作为训练样本，这 4000 张图片包含已知的所有缺陷模式样本以及正常样本。首先对这 4000 张图片进行人工判断，将正常图片与缺陷样本进行二分类，再次，定义 DBN 模型的层数，以及各层 RBM 的隐层和显层的节点数，形成 65536，500，200，2 的层级结构。设置学习 BATCH 数量，学习率，以及迭代次数。将样本输入 DBN 模型中。

第一步，进行两层 RBM 的额外预训练，得到两个预训练权值 $w1'$ 、 $w2'$ 。

第二步，开始 DBN 训练过程，将第一步中的 $w1'$ 、 $w2'$ 作为第一层和第二层 RBM 的初始权值。并进行多层 RBM 训练。

第三步，将样本的分类标签在顶层 BP 网络中进行特征分类，并利用 BP 算法计算网络的代价函数，采用梯度下降优化算法自上而下逐层优化 RBM 权值，最终使整个 DBN 网络得到优化。 $W1$ 、 $W2$ 、 $W3$ 即为优化过的 DBN 网络权值^[44]。

对于本文实际讨论的样本的模型建模如图 4.14 所示。

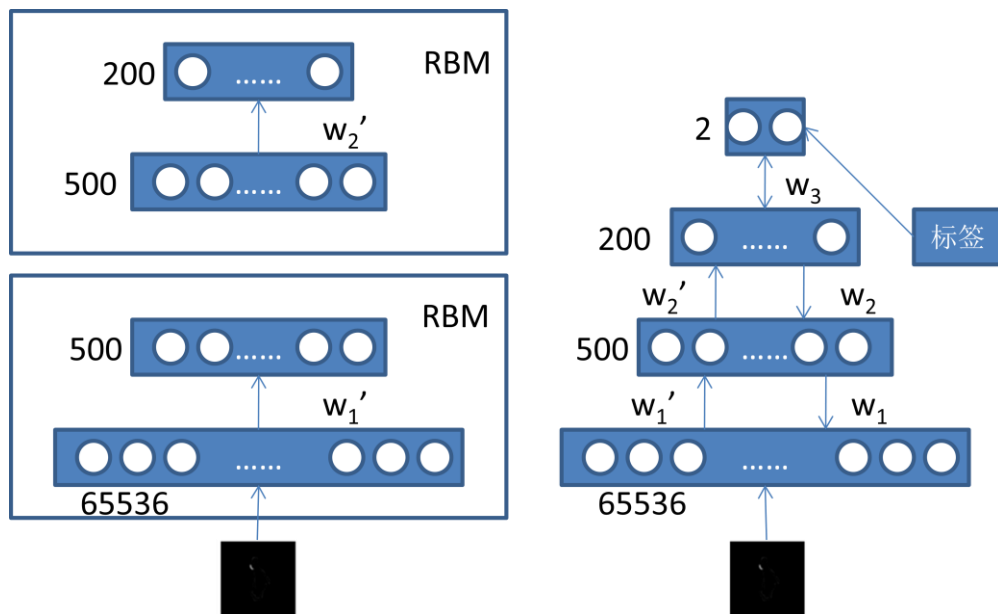


图 4.14 模型结构

4.3 动态模型建模

4.3.1 在线优化算法

如果需要实现模型的动态改变，一定需要引入在线优化算法。在神经网络训练中的难点在于超参数问题，神经网络五大超参数为学习率、权值初始化、网络层数、单层神经元数和正则惩罚项，选择合适的参数需要进行大量的实际验证。在线优化算法设计中关键的一点是使用合适的学习率策略快速更新模型参数，使其快速和准确的拟合模型真实分布。由于在模型生成后，对于后续的模式更新每次仅输入一个单一样本，如果学习率设置不合适，就会产生损失函数剧烈波动致使模型失效或者模型更新不明显，无法保证准确率在平稳的条件下向优化的方向发展，使模型的在线优化无法完成。由于 DBN 本身使用标准 BP 反向传播算法，其中的学习率是一个定值，学习率无法自适应改变，因此不能适应深度神经网络的需求，因此设计一个良好的学习率策略的在线优化算法对模型的稳定性和收敛速度均有帮助。

本文对涉及到运算结果准确度和损失率做了相关的实验对比，并且使用三种调优算法对本样本模型进行了对比。这三种调优算法分别为 SGD，ADAGRAD，FTRL^{[19][45][46]}。假设参数 x 每次调整的步长为 Δx_t ，即 $\Delta x_t = x_{t+1} - x_t$ ， η 为学习率， g_t 为 x 在 t 个样本时刻的梯度值，则各在线优化算法的公式如下：

1) SGD 算法英文名称 Stochastic Gradient Descent，随机梯度下降，每次可以扫描一个样本而使算法更新， η_t 表示 η 随样本时刻 t 改变，其算法如下：即可实现在线梯度下降。

$$x_{t+1} = x_t - \eta_t * g_t \quad (4.45)$$

2) ADAGRAD 算法主要是针对不同参数可以使用不同的更新速率进行了优化, 比如有些参数已经不需要大幅度调整, 而有的参数由于样本太少而需要大幅度调整的时候, 可以使用 Adagrad 对学习率按分配进行调整。对于 Adagrad 算法中的每个参数, 随着更新总距离的增加, 学习率会逐步变小的。 ϵ 为防止分母为 0, 公式如下:

$$x_{t+1} = x_t - \frac{\eta}{\sqrt{\sum_{s=1}^t g_s^2 + \epsilon}} * g_t \quad (4.46)$$

3) FTRL 算法是 Google 在研究大规模数据集以及在线数据流学习中提出来的, 它可以对每次迭代的学习率均进行优化, 增加了稀疏性, 并可以允许每个维度的学习率不同。 σ 表示学习率, 其与 η 存在固定关系, λ_1, λ_2 为两个正则系数, 分别组成两个正则项, 从而增加稀疏性, 其公式如下:

$$x_{t+1} = \operatorname{argmin}_x \left(\sum_{s=1}^t g_s x + \frac{1}{2} \sum_{s=1}^t \sigma_s \|x - x_s\|_2^2 + \lambda_1 x_1 + \lambda_2 \|x\|_2^2 \right) \quad (4.47)$$

4.3.2 在线优化算法建模

对于上述的三种现在优化算法, TensorFlow 将此功能集成在了其框架下, 因此, 优化算法的建模过程就大大简化, 只要使用 `tf.train.saver()` 来保存模型和恢复模型参数即可在 Python 编程环境下不断的更新模型参数, 不断地生成 Checkpoint, 使 t 参数不断累加递增, 以达到持续优化的结果。

对以上三种算法在 TensorFlow 中使用损失函数进行优化验证, 使用 `Softmax_cross_entropy` 进行评估, 此函数包含两个部分, 分别是 `Softmax` 和 `Cross entropy`, `Softmax` 的作用是计算概率分布, 而 `Cross entropy` 是用来估算拟合模型和实际模型之间的区别, 是交叉熵的概念, 假设有 m 组的已知样本, $(x^{(i)}, y^{(i)})$ 为第 i 组样本数据本体及其分类标签, $x^{(i)} = (1, x_1^{(i)}, x_2^{(i)}, \dots, x_p^{(i)})^T$, 模型参数 $\theta = (\theta_0, \theta_1, \theta_2, \dots, \theta_p)^T$, $y^{(i)}$ 在配合 `Softmax` 中可以设置为多分类模式, 本文讨论的分类仅为两分类, 所以 $y^{(i)}$ 去两分类器中的任何一个, 取 0 或者 1。假设函数的定义为

$$h_{\theta}(x^{(i)}) = \frac{1}{1 + e^{-\theta^T x^{(i)}}} \quad (4.48)$$

则交叉熵函数公式为:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)})) \quad (4.49)$$

4.4 小结

本章节分析了本方案使用的深度学习的算法，对 **RBM**，**DBN** 进行了原理分析，并介绍了三种在线优化算法。并对整个模型建立的过程进行了描述，阐述了对实际的工程样本进行了图像处理，描述了参照 **DBN** 模型的参数需求进行了二分类标识添加，样本分组，初始参数定义，以及现在优化算法的设定，以达到可以在 **TensorFlow** 中进行深度学习模型运算的目标。

第 5 章 视觉检测软件设计

5.1 基础软件平台架构

终端电脑的软件系统结构图如下：采用 Windows 7 操作系统，上层安装 VB.NET 编程平台，使用 Ni Vision Acquisition 软件接口库。实现亮度和颜色检测的功能，以及传输采集图片至服务器。



图 5.1 终端软件平台架构

服务器软件系统如下图所示，采用 Linux ubuntu 系统，上层安装 Cudnn 和 Cuda 驱动库，使用 Python 调试环境，并搭配使用 Open CV、Numpy、Tensorflow 功能库，来构建整个核心运算服务器的软件系统^{[47][48]}。

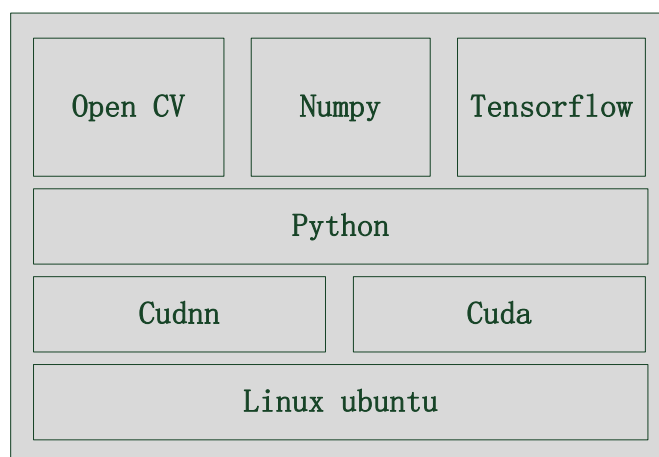
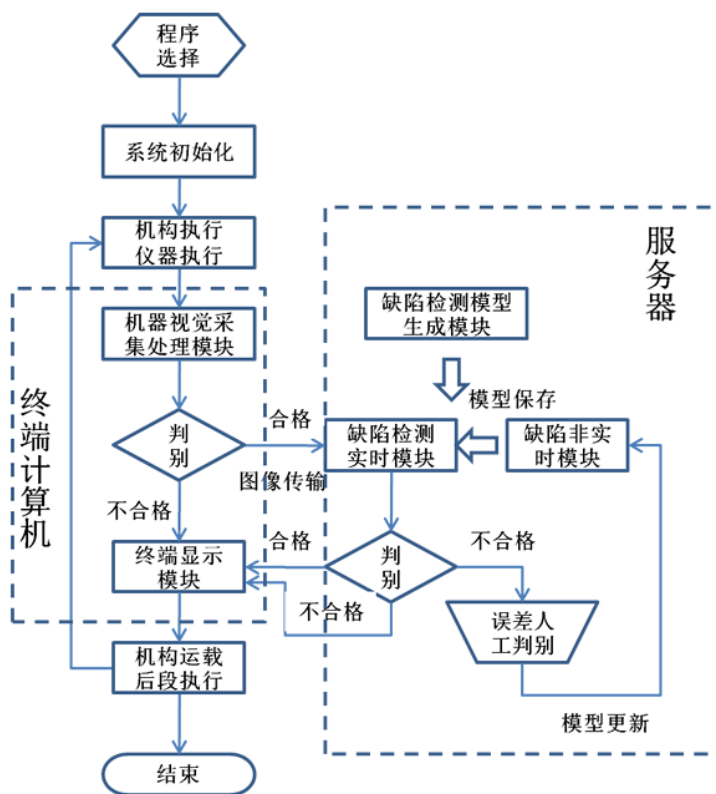


图 5.2 服务器软件平台架构

5.2 系统软件设计

5.2.1 系统总体软件设计

系统软件架构如下图所示，在选择程序和系统初始化后，整体系统将进入 PLC 执行过程，而后终端计算机启动机器视觉采集，并进行颜色和亮度测量，如果测量合格，即将图片传输至服务器进行缺陷检测。缺陷服务器需要预先进行模型生成，在预先生成的深度学习算法模型中对传输过来的图片进行处理判断，如果结果正确，则将结果传递回终端电脑进行显示，如果结果不良，则需要执行人工干预的深度学习算法模型的更新过程，将缺陷模型按照新的纠错样本进行重训练。最后执行后段机械动作，完成整个软件循环。如图 5.3 所示。



5.2.2 终端计算机软件设计

终端软件架构如下图所示，其功能是进行机器视觉亮度和颜色的判断执行。在系统初始化相机参数后，即执行图像采集，进行 RGB 至 HSI 转换，ROI 定位后，执行分量 HS 的提取与颜色判断，并同时执行 I 分量的提取、二值化、腐蚀过程后进行亮度判断，最终将两个判断取与，如果合格，则将图片传递至服务器，等待进行缺陷检测，如果不合格，则直接进行结果显示。如图 5.4 所示。

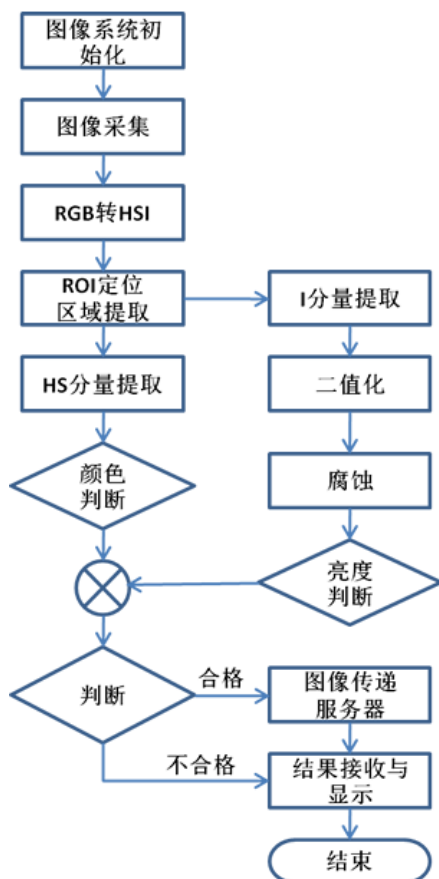


图 5.4 终端计算机软件架构

5.2.3 缺陷检测模型生成软件设计

缺陷检测模型生成软件部分主要实现的功能是 DBN 模型建立起来，如图 5.5 所示。软件设计在 Python 环境下进行，使用 TensorFlow 和 Open CV 算法库，起始需要将参数输入，并将学习样本执行随机化过程后按照给定参数进行分组，再进行图像预处理。同时按照输入参数同时训练所设定层数的 RBM 预训练，将预训练的参数传递和预处理好的图像同时传递给 DBN 网络，执行 RBM 第一层运算后，依次执行第二层 RBM 运算，分类器使用批次标签进行分类，损失函数计算，BP 或其他优化算法反向调优，通过迭代，最终执行模型保存。另外图中的 Dropout 节点是对神经网络单元，在深度学习网络的训练中，按一定的概率将其从网络中暂时丢弃，从而减少计算时间，并防止过拟合。因为，Dropout 功能相当于增加了样本集合数量，使稀疏性变大，可以更好的区分样本特征^[49]。

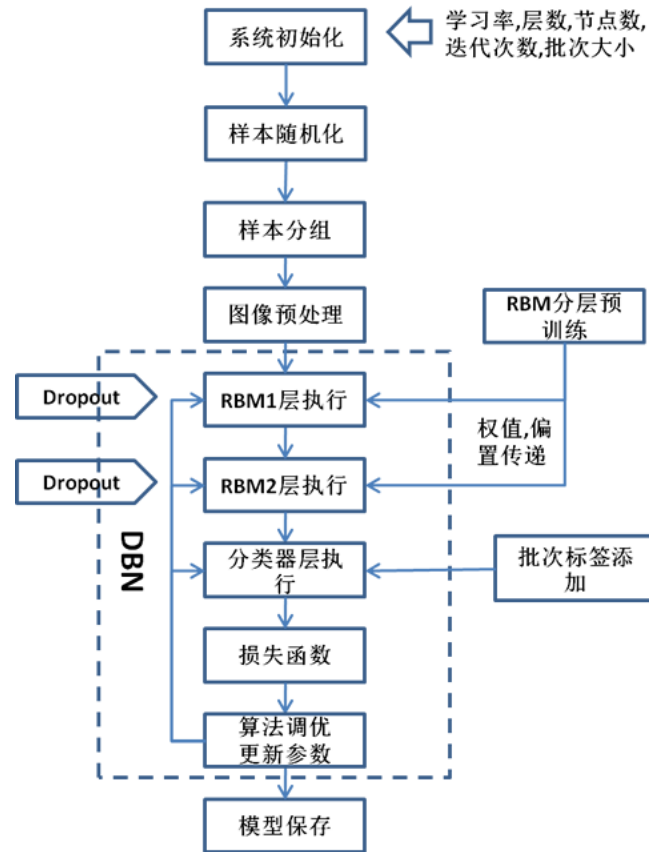


图 5.5 缺陷检测模型生成软件架构

5.2.4 服务器模型执行软件设计

模型执行过程是正常的在线运行模式。首先需要加载说要使用的深度模型，开始等待接收图像数据，进行图像预处理，将图像数据输入至训练好的 DBN 模型中，得出判断结果，如果合格，则将结果传回终端，如果不合格，则需要存下差异结果图像供后续在线优化使用。如图 5.6 所示。

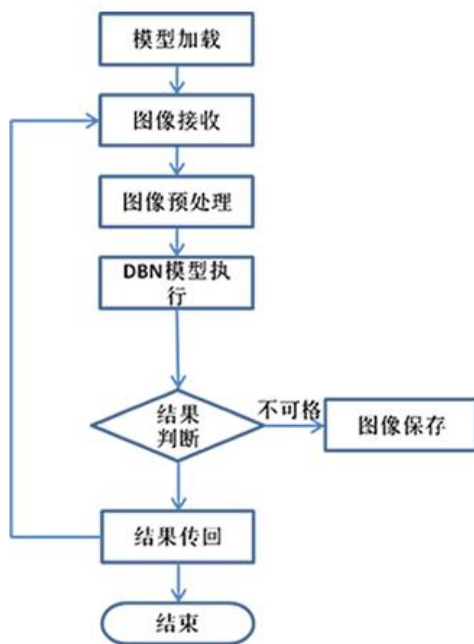


图 5.6 服务器执行软件架构

5.2.5 服务器模型更新软件设计

服务器模型更新软件的作用是将缺陷判断不合格的保存图片进行人工介入判断,将结果进行比对,按照随机单样本的方式对模型进行更新。首先进行模型加载,而后将接收到的图像进行人工判断,并添加人工判断的标签,同时对图像进行预处理。然后和在模型生成过程的 DBN 执行过程基本一样,仅仅使用不同的调优算法,最终执行模型更新。如图 5.7 所示:

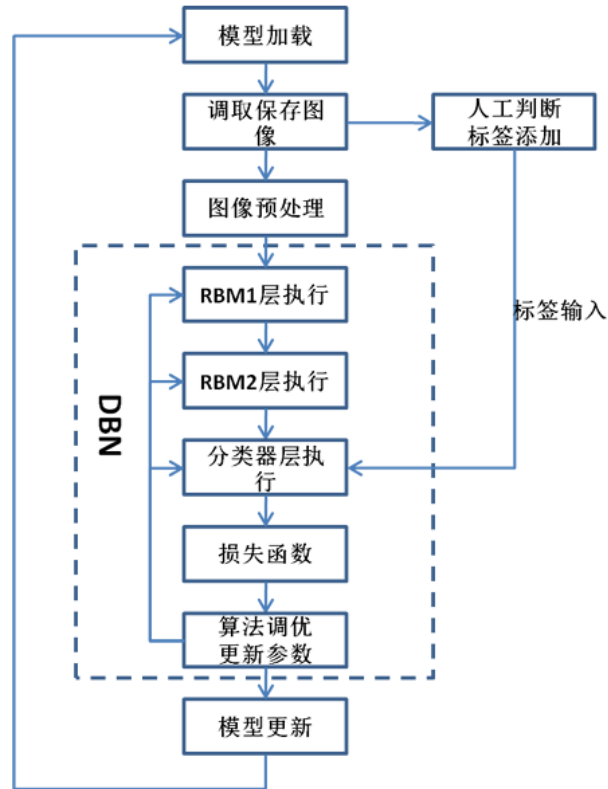


图 5.7 终端计算机软件架构

5.3 小结

本章节描述了整个系统的软件架构，给出了系统基础软件框图。并分别描述了系统总体软件流程，终端计算机的视觉软件流程，缺陷检测模型生成软件流程，缺陷检测的服务器执行软件流程以及服务器模型更新软件流程。上述流程涵盖了整个系统的关键软件流程设计。

第6章 视觉检测实验及结果分析

整体设计制造完成的系统如下图 6.1 所示：后续的视觉检测和结果评估将在此系统中进行。



图 6.1 系统外观图

后续实验过程分为两部分对此系统进行验证，第一部分将验证亮度的检测系统稳定性，第二部分将进行缺陷检测的系统可靠性。在对第二部分的缺陷检测系统的验证中，还细分为参数匹配验证和缺陷检测执行和模型更新执行验证。颜色检测由于仅涉及区分测量，所以不再进行试验分析。

6.1 亮度检测评价过程及结果

对亮度检测采用质量检测方法 MSA（Measurement System Analysis）^[50]进行系统重复性精度的分析。过程如下：

- 1) 选取 10 件从实验室测得的合格样品。如下图 6.2 所示：



图 6.2 MSA 验证用样件

- 2) 准备三个不同的操作人员，对 10 个样本进行 10 轮测试，记录测试结果。
- 3) 将数据代入 MSA 计算公式，进行重复性和再现性 %R&R 统计计算。

4) 得出结果，与系统期望值做对比，评估系统的稳定性。

假设操作人员以 A、B、C，每个操作人员操作 3 次，表实验数据如下表所示：纵列表示测试的样本序列，横排表示不同人员的不同次数，例如 A2 表示第一个操作人员的第二轮测试。

表 6.1 亮度 MSA 数据记录

	A1	A2	A3	B1	B2	B3	C1	C2	C3
1	131.727	130.189	130.731	131.346	131.384	132.231	132.502	131.302	131.393
2	130.764	132.238	131.900	131.864	132.472	132.854	132.527	131.853	132.189
3	135.333	135.128	135.180	135.310	135.260	135.950	136.053	135.339	135.749
4	135.748	136.159	136.164	135.483	136.932	137.079	136.830	136.669	136.186
5	92.785	92.886	93.167	93.183	93.640	94.010	93.495	93.995	93.719
6	129.021	128.377	129.134	128.991	129.045	130.358	130.121	129.992	129.447
7	141.045	140.094	140.944	141.274	141.422	141.989	141.195	142.018	141.568
8	129.017	128.266	128.899	130.209	129.478	131.265	129.719	130.101	129.474
9	116.819	117.043	117.683	117.888	117.983	118.223	118.694	118.804	118.018
10	140.829	140.585	141.720	140.785	141.705	140.969	141.871	141.680	141.224

使用 MSA 统计工具，计算出本文设计系统的重复性和再现性%R&R=4.91%，在 10%以下，属于可接受范围。说明系统设计稳定。

6.2 参数调整验证

在机器学习中，调参是非常重要的，一个好的参数设置会直接影响模型的准确度以及损失函数，不同的样本对应的参数也是不相同的，不能一概而论。下面将基于各种不同的参数条件下的模型反应，寻找最优参数。

6.2.1 BATCH 调整验证

BATCH 参数表示每批次的样本数，一个较大的 Batch，可以充分利用矩阵、线

性代数库来进行计算的加速, Batch 越小, 则加速效果可能越不明显。Batch 太大了, 权重的更新就会不那么频繁, 导致优化过程太漫长。所以 Batch Size 选多少, 要根据数据集规模、设备计算能力去选。从图 6.4、图 6.5 中可以看出, 当 BATCH 设置为 10 的时候, 准确率和损失函数最小。

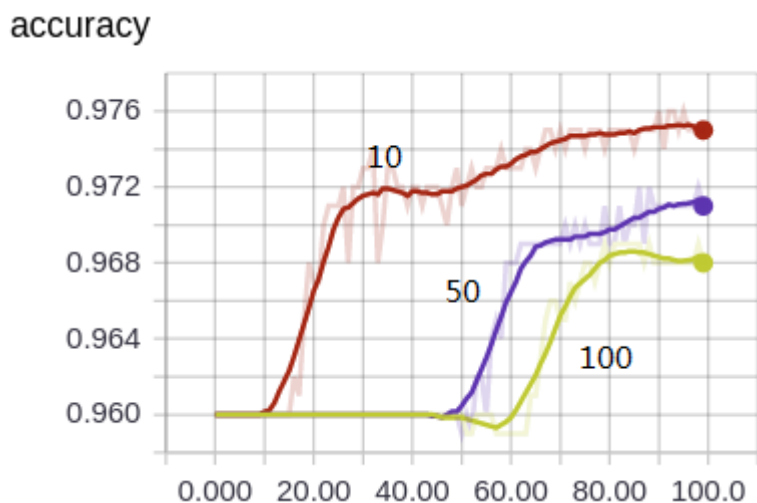


图 6.4 不同 BATCH 参数的准确率曲线对比

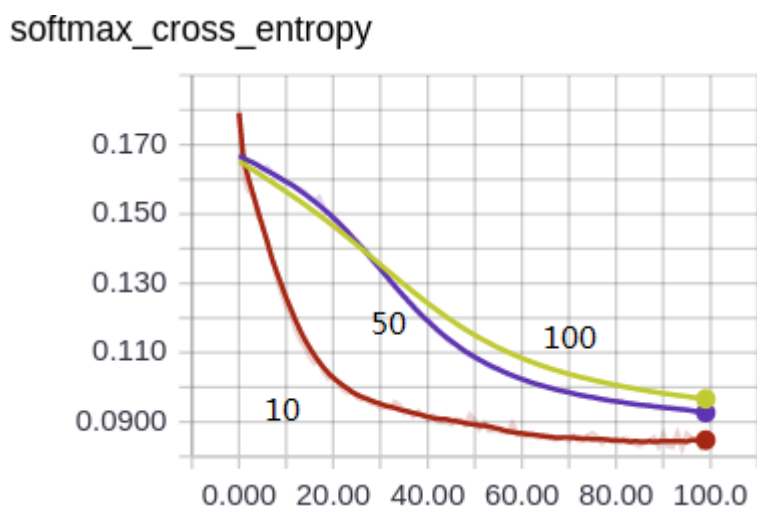


图 6.5 不同 BATCH 参数的损失函数曲线对比

6.2.2 EPOCH 调整验证

EPOCH 表示迭代次数, 一次迭代即表示对所有训练数据的一轮遍历, 在每一个迭代结束时, 计算 Validation Data 的 Accuracy, 当 Accuracy 不再提高时, 就停止训练。这样做还能防止过拟合。从图中可以看出, 当迭代次数设置为 100 时, 最为合适。

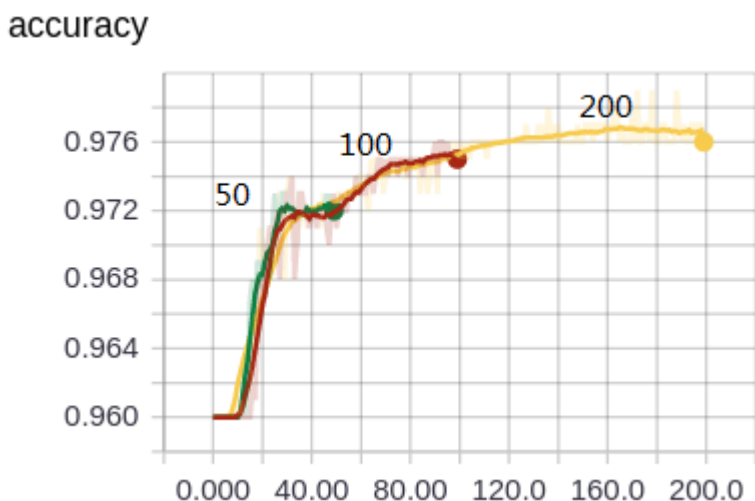


图 6.6 不同 EPOCH 参数的准确率曲线对比

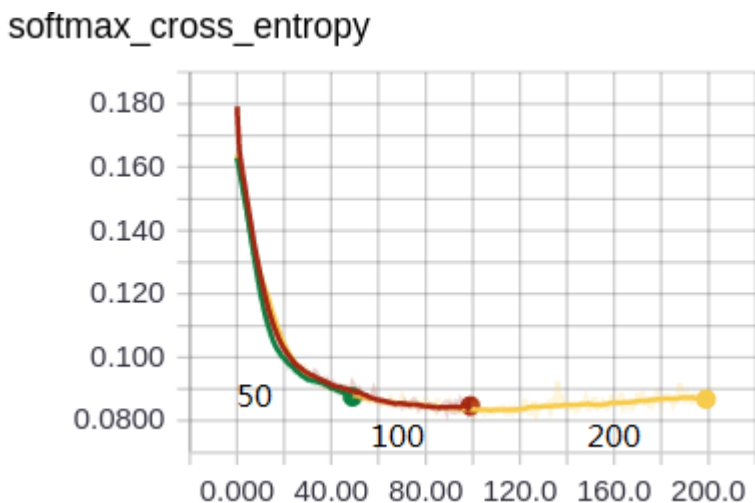


图 6.7 不同 EPOCH 参数的损失函数曲线对比

6.2.3 LEARNING RATE 调整验证

Learning Rate 表示学习率，如果学习速率太小，则会使收敛过慢，如果学习速率太大，则会导致 Cost 函数振荡，最终无法找到梯度极小值结果。

只能通过尝试的方式确定。先把学习速率设置为 0.01，然后观察 Training Cost 的走向，如果 Cost 在减小，逐步地调大学习速率，试试 0.1，1.0...如果 Cost 在增大，那就得减小学习速率，试试 0.001，0.0001...经过一番尝试之后，大概确定学习速率的合适的值。从图中可知，基于此样本模型，学习率设置为 0.01 比较合适。

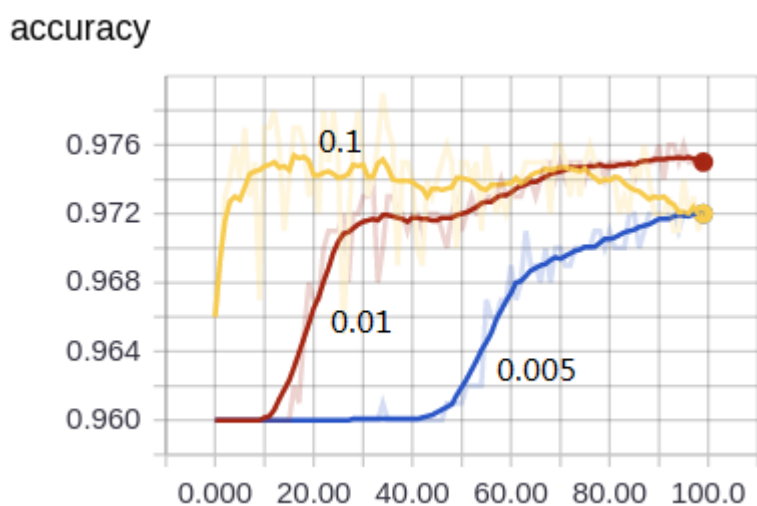


图 6.8 不同学习率参数的准确率曲线对比

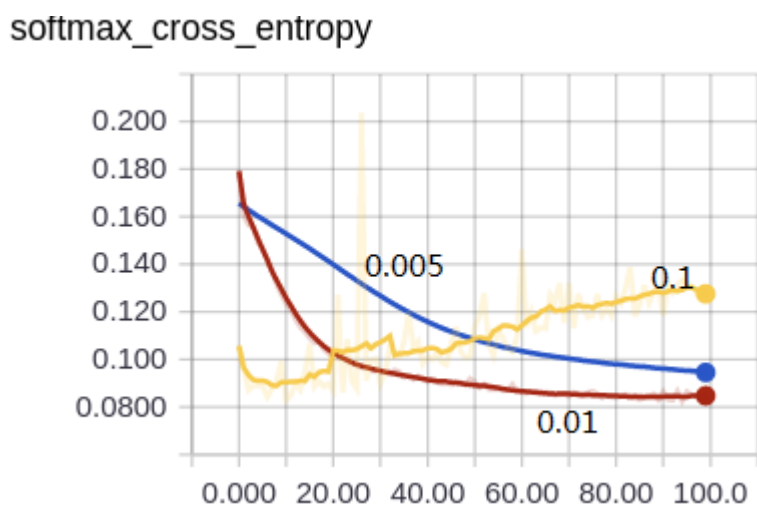


图 6.9 不同学习率参数的损失函数曲线对比

6.2.4 在线优化算法评估验证

对优化算法的评估，需要引入损失函数的概念，它描述了预测输出值与实际输出值之间损失，反映了拟合的参数模型与真实模型之间的差异。

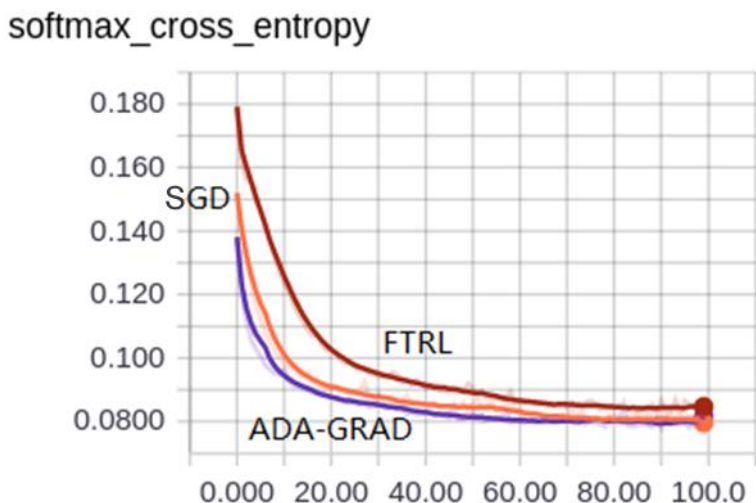


图 6.10 不同 BATCH 参数的损失函数曲线对比

如图 6.10 所示，在 100 次迭代的前提下，FTRL 对于本次实验样本模型的损失函数并不是最好的，而 SGD 和 ADAGRAD 在线优化算法损失函数比较小，ADAGRAD 最优。

而三种在线调优算法在对样本判断的准确度上对比，ADAGRAD 同样最优。准确度对比曲线如下图 6.11 所示。

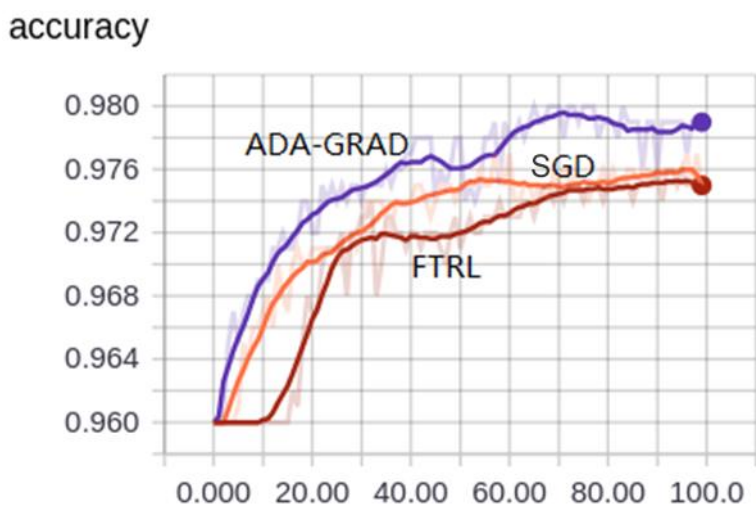


图 6.11 不同 BATCH 参数的准确率曲线对比

另外，在模型生成后，由于在线优化算法的引入，每次的样本输入均会对参数模型进行调整，所以不同的优化算法也会对实际模型运行状态下的时间有影响。在本次的实验条件下，使用 10 个测试样本进行实验，在执行效率上，SGD 最优，FTRL 最差，而 ADAGRAD 算法和 SGD 比较接近。3 种在线优化算法的执行时间如下表。

表 6.2 在线优化算法运行时间对比

	算法名称	10 个测试样本的执行时间（秒）
1	SGD	17.45
2	ADAGRAD	22.20
3	FTRL	37.28

通过实验，针对本次样本的匹配参数如下，迭代 100 次，BatchSize 设置为 10，学习率取 0.01，使用 ADAGRAD 调优算法，可使准确度达到 97.9%，并使损失函数最小，即最接近真实样本输入的模型。并且在线优化的时间在可接受的范围内。

6.3 缺陷检测评价过程及结果

实验过程如下：

1) 从样本库随机抽取 4000 个训练样本，基于 4000 个训练样本进行深度学习，生成模型，并保存。

2) 随机从样本库抽取 2000 个测试样本，如图所示，对样本做人工预判断，对合格和缺陷比例进行控制，保持人工判断的合格件 1500 个，缺陷件 500 个。

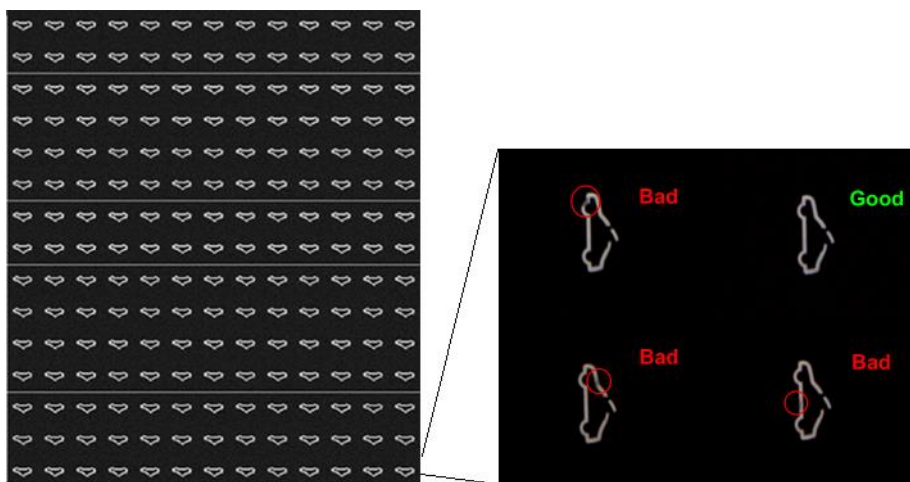


图 6.12 学习样本以及人工识别标签

- 3) 将 2000 样本模型输入保存好的深度学习模型，得出由计算机判断的结果。
- 4) 将人工判断结果与计算机判断结果进行对比，找到计算机判断错误的样本。
- 5) 使用在线学习模型对计算机判断错误的样本进行单样本重训练。
- 6) 重复 3-4 过程，观察结果。

在 BatchSize 设置为 10，学习率设置为 0.01，迭代次数设置为 100，在线优化

算法设置为 FTRL 的情况下，4000 个学习样本的学习时间为 32 分 10 秒，学习样本的准确率达到 97.5%。对 2000 个测试样本的测试时间为 2 小时 3 分 46 秒，平均 3.7 秒完成一次测试样本的识别和参数优化。采用与人工对比的交叉对比验证，两者出现 101 个样本的差异，准确率 94.95%。第二轮交叉验证，在同样测试样本的条件下，基于第一轮样本测试差异，对模型进行单个样本的重训练，则第二次实验结果为，出现 92 个样本差异，准确度 95.4%，有小幅改善。结果如下表所示：

表 6.1 测试样本验证结果

	测试样本数	人工判断合格件数	人工判断故障件数	合格件人机匹配数	故障件人机匹配数	准确率%
第一轮测试	2000	1500	500	1453	446	94.95
第二轮测试	2000	1500	500	1458	450	95.4

实验结果表明，由于随机样本的不同，学习样本在足够的迭代次数后可以到达较好的准确率，但测试样本是全新样本，与学习样本存在差异，因此准确率较低，进行单样本再学习之后，准确率稍许上升，说明模型可以按照在线方式进行优化。

6.4 小结

本章通过实验验证了光学检测部分的亮度检测的可靠性，以及缺陷检测在不同参数下对结果带来的影响，选择了针对此模型的最优化参数方案，并验证了并对比了 3 种在线优化算法的实际执行效果，确定 ADAGRAD 的调优算法与此模型匹配最好。并通过测试样本进行了两轮实测，说明了本缺陷模型对车载零件表面光刻面的缺陷检测是有效的，但还有可提升的空间。

第7章 结论与展望总结

7.1 结论

本文完成了对整体系统的可行性分析，并成功实现了汽车零配件表面视觉在线检测系统的设计，主要成果如下：

1) 完成了视觉检测系统的硬件架构设计以及软硬件选型，并最终实现了整套系统的建立。

2) 提出了使用工业相机替代光度色度计的实现方式，通过白平衡和极限样件，对系统进行参数设置和标定，确立了实现亮度色度检测的执行过程。

3) 实现了使用深度学习模型来测量表面信标缺陷，并通过加入在线优化算法，使其可以在生产现场进行部署，成功建立了使图像可以进行在线的深度学习缺陷检测的系统模型。

4) 完成了整个检测系统所有的软件设计，满足了从控制到颜色亮度检测，以及图像传递和缺陷模型运算的所有功能。

5) 针对亮度和缺陷检测进行了实验验证，特别是缺陷模型使用不同的参数进行尝试，寻找到了最适合此模型的参量。

整套系统最终达到了比较理想的视觉检测效果，说明其针对亮度和颜色的标定方式是可靠的，针对缺陷的检测方法是可以实现的。

7.2 展望

由于对于深度学习的算法理解有限，以及时间限制，无法将所有参数进行匹配调优，因此系统对缺陷的检测距完全脱离人工的目标还有一段距离，目前仍然需要运行在低频率的人工干预模式下。不过相信增加样本数量以及对参数的调优，在模型长时间的运行之后，一定可以达到完全无人干预的执行模式，实现无人化的车载人机交互面板零配件的在线视觉检测系统。未来对于作者的工作方向如下：

1) 需要进一步完善系统模型，尝试对深度学习的模型结构参数进行优化匹配，寻找效率和准确率的平衡点。

2) 对在线优化算法进行深入研究，找到其执行方式以及对模型的影响因素。

3) 对于其他生产环节的测试方向也将引入机器学习算法进行优化研究，以实现部分产线的智能制造。

机器代替人的检测系统是实现智能制造的基础，尤其近年来对于人的感官系统

的模拟测试更是对检测系统提出了更高的要求。随着中国制造 2025 的推进，相信在未来，越来越多的制造企业会引入高柔性化的生产设备和测试设备，以满足个性化的生产需求。视觉系统的发展相信也会向着智能系统的方向发展，逐步的接近人类的视觉感官。从智能识别、智能定位、智能装配到智能测试，将越来越多的从浅层智能向深层智能发展。可以预见深度学习技术将在工业生产环节的物体识别和缺陷检测方面首先推开，而后深入到其他生产支持系统。生产数据不再是枯燥的数字记录，而是可以使用深度学习对其进行特征分析，从而指导生产，代替人类对生产系统做初级的决策。除了视觉系统，触觉系统也会引入深度学习来模拟人类的触觉感官，灵活的装配系统也会随之而来。也许不久的将来，汽车零配件生产再也无须工人，完全靠订单支持，从整车的设计图纸直接转化为各个全自动化产线的生产参数，完全柔性化的实现各种零件的批量生产。

致谢

漫长的技术探寻之路最终形成了此文，欣慰有嘉，它带给我的知识财富是不可估量的，拓展了我的知识面，使我更有信心面对今后的图像和深度学习领域新技术的到来。特别感谢我的导师罗峰，在我的研究生生涯中给予了极大帮助和关怀，从论文的选题到论文的素材收集均提供了巨大的支持，并在最终的论文修改过程中耐心的指导我一遍又一遍的修改，让我在实际工程领域的总结能力又上升了一个台阶。同时也感谢我的校外导师沈建于，以及工程师王鑫在技术上给予我的支持，尤其是王鑫带领的核心技术团队在整个项目的过程中，克服了项目时间短任务重的困难，经历了无数次的实验，最终提供了可靠的实验数据，为此项目的顺利完成做出了巨大贡献。另外感谢所有参与项目的人员，包括工艺、机械设计、机械装配、电气设计、电气集成、软件调试工程师的大力支持，特别感谢为此项目支持的所有机器视觉领域和深度学习领域的各位专家。

2017 年 1 月

参考文献

- [1] Peshkin L, Pfeiffer A. Bayesian Information Extraction Network[C]. Proceedings of the 18th International Joint Conference on Artificial Intelligence. Acapulco, Mexico: Morgan Kaufmann Publishers Inc., 2003:421–426.
- [2] Mc Callum A, Freitag D, Pereira F. Maximum Entropy Markov Models for Information Extraction and Segmentation[C]. Proceedings of the Seventeenth International Conference on Machine Learning. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000:591–598.
- [3] Mansouri A, Affendy L S, Mamat A. A New Fuzzy Support Vector Machine Method for Named Entity Recognition[C]. International Conference on Computer Science and Information Technology. Singapore: IEEE Computer Society, 2008:24–28.
- [4] Tsuboshita Y, Okamoto H. Graded information extraction by neural network dynamics with multihysteretic neurons[J]. Neural Netw, 2009, 22(7):922–30.
- [5] Li X, Wang Y Y, Acero A. Extracting structured information from user queries with semi-supervised conditional random fields[C]. Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval. Boston, MA, USA: ACM, 2009:572–579.
- [6] 金黄斌, 秦兴, 郭斌林. 基于灰度世界和白点检测的自动白平衡算法[J]. 电子器件. Vol.34, No.2, Apr.2011.
- [7] 邵新玉, 华继钊. 基于机器视觉的无纺布缺陷自动检测系统[J]. 计算机科学, 2014, 6
- [8] 周卫国. 基于图像分析的炉体喷涂缺陷自动检测方法[D]:[硕士学位论文]. 天津大学, 2008
- [9] 郑剑锋. 带钢表面缺陷检测方法研究[D]:[硕士学位论文]. 西安建筑科技大学, 2006
- [10] 张秀玲, 来永进, 高武杨, 等. 基于 DBN 神经网络的板形识别研究[C]. Proceeding of the 35th Chinese Control Conference, 2016, 6
- [11] Hinton, G.E. Learning multiple layers of representation[J]. Trends in Cognitive Sciences, vol.11, pp.428-434, 2007.
- [12] Hinton, G. E., Osindero, S. and Teh, Y., A fast learning algorithm for deep belief nets[J]. Neural Computation 18:1527-1554, 2006
- [13] Jaitly N, Hinton G. Learning a better representation of speech sound waves using restricted Boltzmann machines[J]. Acoustics, Speech and Signal Processing(ICASSP), 2011, vol(5):5884-5887.
- [14] 张志军, 薛磊, 许阳明, 等. 深度学习研究综述[J]. 计算机应用研究, Vol.29 No.8 2012.8
- [15] Cortes C, Vapnik V. Support-Vector Networks[J]. Machine Learning, 1995, 20(3):273-297.
- [16] Haykin S. Neural Networks: A comprehensive foundation[M]. 2nd ed. New York: Prentice-Hall, 1999
- [17] Bengio Y. Learning deep architectures for AI[J]. Foundations and Trends in Machine Learning, 2009, 2(1):1-127
- [18] 刘建伟, 刘媛, 罗雄麟. 玻尔兹曼机研究进展[J]. 计算机研究与发展, 2014, 51(1)
- [19] L. Xiao. Dual averaging method for regularized stochastic learning and online optimization[C]. In NIPS, 2009

- [20]H. Brendan McMahan , Gary Holt , D. Sculley, Ad Click Prediction: a View from the Trenches[C].in KDD'13 August 11-14, 2013
- [21]王亚鹏.机器视觉系统中镜头的选用技巧[J].自动化博览. 2006(01)
- [22]王正胜, 施朝健.基于两种色彩空间的颜色选择方法[J].计算机应用与软件.Vol.21, No.2, 2014.2
- [23] Munisamy Anandan. Progress of LED Backlights for LCDs[J].Journal of the SID, 2008, 16(2):287-310.
- [24] Rafael C Gonzalez, Richard E Woods. Digital Image Processing (2ndEdition)[M].USA: Pearson Education Inc , 2002.
- [25] 章毓晋, 图像处理和分析[M], 北京:清华大学出版社, 1999.
- [26] 郑南宁, 计算机视觉与模式识别[M], 北京: 国防工业出版社, 1998.
- [27] 马琳.基于机器视觉的模具残留物检测系统研究[D]:[硕士学位论文]东华大学, 2015
- [28]Fogel, I, Sagi, D. Gabor filters as texture discriminator[J].Biological Cybernetics. 61:103-113(1989).
- [29]Hinton G E, Sejnowski T J. Learning and relearning in Boltzmann machines[M].Cambridge: MIT Press, 1986: 282-317.
- [30]Hinton G E. Training products of experts by minimizing contrastive divergence[J].Neural Computation.2002. 14(8):1771-1800
- [31] 高强, 马艳梅. 深度信念网络(DBN)网络层次数量的研究及应用[J].科学技术与工程, 2016, 8
- [32]Yoshua Bengio, Pascal Lamblin, Dan Popovici and Hugo Larochelle, Greedy Layer-Wise Training of Deep Networks[C]. NIPS , pp. 153-160, MIT Press, 2007
- [33]Martin T Hagan, Howard B Demuth, Mark H Beale 著, 戴葵等译神经网络设计[M].北京: 机械工业出版社, 2002, 9
- [34] 任运生.BP 神经网络技术在信用社客户分类中的应用研究[D]:[硕士学位论文].湖南大学, 2014
- [35] Lee H, Grosse R, Ranganath R, et al. Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations[C]. Proceedings of the 26th Annual International Conference on Machine Learning. New York, NY, USA: ACM, 2009:609–616.
- [36] Liu Y, Zhou S, Chen Q. Discriminative Deep Belief Networks for Visual Data Classification[J]. Pattern Recognition, 2011, 44(10-11):2287–2296.
- [37]Maskey S, Zhou B. Unsupervised Deep Belief Features for Speech Translation[C]. Interspeech.[S.l.]: ISCA, 2012:2358–2361.
- [38] Kim J, Nam J, Gurevych I. Learning Semantics with Deep Belief Network for Cross-Language Information Retrieval[C]. 24th International Conference on Computational Linguistics. Mumbai, India: The COLING 2012Organizing Committee, 2012:579–58
- [39]陈宇.基于深度置信网络的中文信息抽取方法[D]:[博士学位论文].哈尔滨工业大学, 2014
- [40]杨春德, 张磊.基于自适应深度置信网络的图像分类方法[J].计算机工程与设计, 1000-7024 (2015) 10-2832-06
- [41]晁静.基于 DBN 的汇率检测研究[D]:[硕士学位论文].南京大学, 2012.5
- [42]李新国 黄晓晴. 一种基于 DBN 的高光谱遥感图像分类方法[J].电子测量技术, 2016, 7
- [43]Harris D M, Harris S L. Digital design and computer architecture [electronic resource][M]. China Machine Press, 2007.

- [44]李子久.人工鱼群-BP 神经网络算法在文本分类中的应用研究[D]:[硕士学位论文].河南工业大学, 2011
- [45] J. Duchi and Y. Singer. Efficient learning using forward-backward splitting[C]. In Advances in Neural Information Processing Systems 22, pages 495-503. 2009.
- [46] H. Brendan McMahan and Matthew Streeter. Adaptive bound optimization for online convex optimization[C]. In COLT, 2010
- [47] Gary Bradski, Adrian Kaebler. Learning OpenCV[M].北京: 清华大学出版社, 2009.
- [48] 梁勇 (Y.Daniel Liang). Python 语言程序设计[M].北京: 机械工业出版社, 2015, 2
- [49]. Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: A simple way to prevent neural networks from over fitting[J]. The Journal of Machine Learning Research, 2014, 15(1): 1929-1958.
- [50]吴遵高.测量系统分析[M], 中国标准出版社, 2004-1

个人简历、在读期间发表的学术论文与研究成果

个人简历:

王彤, 男, 1983 年 4 月生。

2005 年 7 月毕业于河南大学测控技术与仪器专业, 获学士学位。

2012 年 3 月入同济大学读硕士研究生。

已发表论文:

[1] Positioning Control System Based on Computer Vision , ICCAR 2016 , ISBN:978-1-4673-9858-9, IEEE Catalog Number:CFP16IWZ-PRT, P108-P111.