

南京大学研究生毕业论文中文摘要首页用纸

毕业论文题目：零件表面缺陷检测的研究和应用

多媒体 专业 2015 级硕士生姓名：宋佳
指导教师（姓名、职称）：郭延文 教授

摘要

随着计算机视觉领域和机器学习领域的不断发展和互相结合，迸发出了的巨大能量，取得了令人瞩目的成就，这些成功，不仅影响着学术界，被诸多学者们深入的研究，更是在工业界被证明行之有效，得到了广泛的应用。在计算机视觉和机器学习结合的领域里，目标检测是非常经典和基础的课题，而缺陷检测则是目标检测一个非常重要的应用领域。尤其是对于要求较高的金属零件来说，如汽车、飞机、航天器等，一旦有缺陷的零件漏检流出，轻则在更高一级的设备组装时发现被退回和索赔，重则造成设备的实际使用中出现故障，甚至引发事故。前者仅仅会造成经济损失，后者甚至会关系到使用者甚至社会的安全，因此缺陷检测显得尤为重要。

传统的目标检测算法往往分两步，首先提取图像中的特征，然后使用分类决策将其分开。其缺点是对特征选择要求严格，并且容易受到图片数据的纹理、光照、噪声、成像效果等条件的影响，难以取得较好效果。随着深度学习领域的蓬勃发展，越来越多的深度学习算法被应用的目标检测检测中，发展出了不同的检测模型，在某些领域尤其是竞赛中获得了理想的效果。但对于精密仪器来说，对精度的要求往往非常高，并且由于缺陷目标本身较小，过深的神经网络模型往往会由于抽象程度过高反而导致效果下降。

针对这一问题，本文首先设计和实现了提取零件法线图的方法，并对基于传统方法的目标检测算法和基于深度学习的检测算法进行了深入研究，将法线图作为输入结合传统算法和深度学习算法做缺陷的检测，取得了理想的效果。本文的主要工作和创新有：

1. 零件表面法线信息获取硬件设备的设计。现有的扫描物体表现信息的视觉扫描系统一般基于扫描仪或照相机采集信息。它们主要有两个缺陷，首先是信息采集不全，一般仅能采集色彩信息，而不能获得法线信息；其次是

采集数据的质量不稳定，难以达到实用标准。本文设计了一套专门扫描物体表面法线的设备，包括遮光模块、平台模块、灯光模块、拍照模块和控制模块五大模块，可以获得精准方向光源下的高质量照片。

2. 材质表面法线信息计算的方法。本文使用设计的硬件设备，提出了使用不同方向光源下照片来计算物体法线的方法。同时，为了提高拍摄照片的质量从而优化法线信息的精度，本文还对相机做了色彩校正、白平衡校正、畸变校正等校正，并针对方向光源设计了光线补偿算法，从而得到零件精细而平滑的表面法线图。
3. 将法线图应用到基于传统的缺陷检测算法中。将法线图和不同方向光源照片作为输入对比，并分析和对比了法线图的 Haar-like 特征、图像梯度特征、LBP 特征、GLCM 特征、HOG 特征等不同特征的分类效果。
4. 本文对于传统的缺陷检测算法，针对金属零件缺陷检测这一课题，研究了数据预处理方法、数据增强方法、支持向量机模型、Adaboost 算法、Cascade 算法等其他学习算法，首先解决了精密仪器缺陷零件较少，数据不平衡的问题，同时使用集成学习的方法，不仅提高了模型的漏检率，同时提高了模型的误检率，得到了一个符合目标要求的方法。
5. 本文还是针对这一问题将法线图作为输入，实验了基于深度学习的方法，并将其与我们设计的传统方法进行了对比。由于网络深度的要求，深度学习往往会提取图像“高层次”的特征，然而针对检测目标较小的问题，或许较高层次的特征并不一定比基于传统方法设计和提取的特征更有效。并且，由于深度模型的复杂性，不论是训练速度还是检测速度上，传统方法都有明显优势。

关键词： 法线图；缺陷检测；支持向量机；深度学习

南京大学研究生毕业论文英文摘要首页用纸

THESIS: A Research on Network Infrastructures
for Data Centers

SPECIALIZATION: Multimedia

POSTGRADUATE: Jia Song

MENTOR: Professor Yanwen Guo

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

With the continuous development and combination of computer vision and machine learning, the tremendous energy they have generated has brought remarkable achievements. Not only has it been deeply studied by so many scholars in academia, but it has also been widely used in the industry. In the field of computer vision and machine learning, object detection is a very classical and basic subject, while defect detection is a very important application field of it. Especially for the metal parts with high requirements, defect detection looks like particularly important. Once the defective part is missed to be noticed, it will lead to be returned in the assembly of the more advanced equipment, some companies will make a claim for it. Furthermore, those parts that has not been discovered will make a malfunction, induce users' danger and cause economic loss.

The traditional object detection algorithm is usually divided into two steps. First, the features in the image are extracted, and then the classifier with the fratures as in-

put is used to classify the image. There are two main disadvantages, one of them is the difficulty of feature engineering. And the performance is susceptible to the inputs' texture, lighting, noise, and other conditions. With the vigorous development of deep learning, more and more deep learning algorithms have been applied to object detection, and more and more different detection models have been developed, which have achieved ideal results in many fields, especially competitions. However, for precise parts, the requirements are pretty high at most time, and because the defection seems to be small as usual, the deep neural network would not lead to a good enough outputs sometime due to the high level's abstraction of input images.

To solve this problem, this paper imports the normal map as input for the first time, and does a lot research both on traditional methods but also some deep learning methods. We use the normal map as input as traditional classifier and convolutional neural network and get a state-of-the-art performance. The main work and innovation of this article are:

1. The design of the hardware to acquire the normal map. The existin normal map acquisition scheme is mainly based on two models: scanner or camera. There are two important issues of those models: 1.only can get the color information; 2.the normal map is not stabilize. We design a new hardware which could lead to a accurate and stabilize normal map with our algorithm.
2. And we put forward a model to compute the normal map with the information collected by our hardware maintained above. In this compution model, we make some coorection and enhancement before the compution such as color correction, distortion Calibration and light compensation to make up the imperfection of the capture equipment.
3. Also, we has experiment on the traditional object detection methods with our extracted normal maps as input. We test many diffierent kind of features such as haar-like, LBP, GLCM, HOG and others, and compare the results of our models with regular input.
4. Furthermore, considering the imbalance of data between the parts with defections and those without defection and the rareness of defection on the parts, we have to try a variety of ways to solve this problem. Also, we have to achieve a lower recall rate while the loss of missing defecting a defection part is pretty larger than regarding a

normal part as a defection one.

5. At last, we make some experiments with deep learning models, with the results of those models, we do some analyses on the performance and reasons combined with traditional methods' output.

keywords: Normal maps, Defect detection, SVM, Deep learning

第一章 绪论

本章首先介绍课题的动机、目的以及研究意义，引出这一课题的研究背景，接着在此基础上概述现阶段这一领域的研究方法和解决方案，这些方法包括传统机缺陷检测方法以和基于深度学习的缺陷检测方法，以及各自的优缺点，最后介绍本文的主要工作以及后续几章内容安排。

1.1 研究背景

随着工业的发展，金属零件的需求量和增长量大幅度增长。在工业生产中，对金属零件的加工基本已经全面实现自动化机械生产。在实际应用中，对金属零件的要求往往是非常高的，尤其是在汽车领、航空航天等的领域，对零件的要求严苛。一般精密的金属零件往往要求其表面不能有深度或者宽度超过5mm 的划痕或者裂缝等缺陷因素。但是，在零件加工过程中，由于本身设备的问题，或者加工过程中的环境因素以及其他因素，不可避免的会出现各种缺陷，比如裂纹、起皮、拉线、划痕、凹坑、凸起、斑点、腐蚀等缺陷，甚至零件偏芯、气孔等严重缺陷。传统零件加工中需要雇佣大量人力来使用人眼的力量检测缺陷，往往浪费大量人力、物力、财力但是缺仍然会有大量漏检。然而这些零件一旦漏检流出，轻则在装配检查过程中被查出来，企业不得不对其进行退回和赔偿，严重的甚至会导致组装完整的设备如汽车、飞机、轮船等的崩溃，牵一发而动全身，致使整个设备瘫痪，引发严重生产问题，甚至引发社会安全问题。如图 1-1展示了常见的几种金属零件。这些金属零件大都呈环形并带有一些齿轮和咬合口。并且工业上要求每一个齿轮都有相应的定位点用来对其定位。缺陷常常就出现在这些环形区域的表面，呈现出裂缝、划痕、腐蚀等症状。

随着计算机视觉和机器学习的领域的不断开拓和结合，二者迸发出来的巨大能量带了令人瞩目的成就，而在其中，目标检测作为图像处理的和计算机视觉的经典课题，目标检测是使用计算机相关算法通过输入的图片模拟人眼获取自己感兴趣目标的过程。接收到人眼获取的图片并理解其中的含义，找到自



图 1-1: 常规金属零件图

己感兴趣的目标这一过程，这对于人类来说是一种本能，如同呼吸喝水一样简单顺畅，但是对于计算机来说，它所获得的图片却只是简单地 [0 – 255] 的数值，很难理解这些信息组合起来代表了什么含义，分不清楚其中到底有什么物体，更不清楚这些目标的位置。因此，对于计算机来说，要识别一个目标本身就是一件非常艰巨的任务。

传统的算检测法往往分为两步，首先是提取经过人工设计的特征，如梯度直方图^[?](Histogram of oriented gradients, 简称 HOG) 特征、局部二值化^[?](Local Binary Patterns, 简称 LBP) 特征、灰度共生矩阵^[?](Gray-level Co-occurrence Matrix, 简称 GLCM) 特征、Oriented FAST and Rotated BRIEF^[?](ORB) 特征等等。然后使用这些特征作为数据输入，训练分类模型。分类模型又有决策树、支持向量机、逻辑斯回归等不同的选择。这些方法虽然取得了一定的效果，但是由于原始图片数据的质量、特征设计的好坏、算法本身的描述能力的强弱导致效果有限，不能达到实际应用的需求。随着技术的发展，深度学习逐步发展成为目标检测的中流砥柱，然而，其优良表现往往是由于其越来越深的网络结构大大提高了模型的表达能力，但往往正是这种复杂的网络结构使得该模型提出的特征过于抽象，对于小物体的目标检测效果不好，而往往精密金属零件本身就不大，其上面的缺陷显得愈加渺小，使得深度学习不能够很好的检测。

因此，考虑到既然对其精度要求如此之高，能不能设计和提取出一种能够表示缺陷宽度和深度的信息来进行计算呢？直接提取物体表面的高度信息往往是昂贵和精确地，但是，我们却可以根据不同的灯光入射角度拍摄得到的照片来计算物体法线图。而法线提额度就是在物体的凹凸表面的每个点上做法线，

通过 *RGB* 颜色通道来标记物体表面法线的方向，它恰恰反应物体表面高度的变化信息，因此我们提出了设计了一套获取物体表面法线的算法，并将其应用到传统目标检测方法和基于深度学习的目标检测方法中去，进行了丰富的实验和测试，取得了良好的效果。

1.2 现有解决方法和不足

在法线获取领域中，最早的材质表面信息库大部分通过相机拍摄或者直接通过图像处理软件得到，2013年 Vizoo 公司建立并推出了 Vizoo3D xTex，这是一款以相机为核心的材质提取软件，通过相机对材质进行拍摄，通过一系列的算法计算出材质的法线、漫反射、高光以及透明度信息，是目前比较成熟的材质扫描系统，该系统还融合了材质信息后期处理的一些功能，但是计算得到的材质贴图不足。国内也有类似的软件如厦门启尚科技研发的 FSM 3D 高清面料扫描仪，这款扫描仪小巧快速，但是能够扫描的材质的大小有限。

而在目标检测领域，传统的目标检测过程一般由获取原始图像、数据预处理、特征提取、训练分类器、使用分类器得到检测结果这几个步骤，如图 1-2 所示。

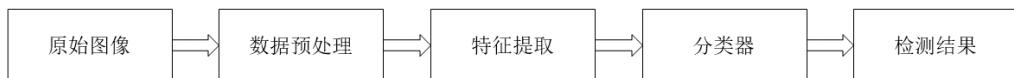


图 1-2: 传统方法缺陷检测步骤

在特征提取阶段，常用的特征有 LBP 特征、HOG 特征、CLCM 特征、梯度 (gradient) 特征、哈尔^[?](Haar-like) 特征等等。在分类器分类的时候，一般会使用图像中不同尺寸的滑动窗口作为候选区域做分类。分类器一般使用支持向量机^[?](Support Vector Machine, 简称 SVM) 等传统分类方法。在传统目标检测中，多尺度形变部件模型^[?](Deformable Part Model, 简称 DPM) 表现优异，连续获得 VOC(Visual Object Class)2007 到 2009 的检测冠军。DPM 把物体看成由多个部件构成，用部件之间的关系来描述物体，继承了 HOG 和 SVM 的有点，在不同的任务上都取得了不错的效果，但是相对复杂，检测速度较慢。传统分类目标缺陷检测一般存在两个主要问题，一是基于滑动窗口的区域选择策略没有针对性，而且时间复杂度高，窗口冗余；而是手工设计的特征对于图片多样性

的变化没有很好地鲁棒性，并且针对不同的问题，在选择和设计特征的时候异常的繁琐和复杂。

自从卷积神经网络^[?](Convolutional Neural Network, 简称 CNN) 出现以来，诞生了非常多成熟框架，其中最具代表性的有 LeNet^[?]、AlexNet^[?]、VGG^[?]、GoogleNet^[?]、ResNet^[?]等。其中，LeNet 是 CNN 的开山之作，主要被用于识别手写数字，接着 Alex Krizhevsky 提出了 AlexNet，一举获得了 2012 年的 ILSVRC 比赛冠军，无论是在速度还是在精度上都将第二名远远甩在身后，之后 VGG、GoogleNet 不断加深网络，进一步提升了模型的效果，ResNet 在研究了为何网络加深模型效果反而会变差的原因之后增加了一条从低层到高层的“高速通道”，是的网络更深，效果也变得更好。这些网络都在 ILSVRC 上取得了瞩目的成绩。基于深度学习的目标检测一般分为获取原始图像，数据的预处理，训练神经网络和使用神经网络得到检测结果四个步骤，如图 1-3 所示。

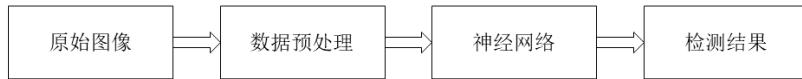


图 1-3: 基于深度学习的缺陷检测步骤

这些传统的框架大都是首先在分类上面得到较好的结果，接着提出了各种目标检测的算法。比较经典的目标检测算法有两大类，一类是以 R-CNN^[?]、Fast R-CNN^[?]、Faster R-CNN^[?]为代表的方法，该类方法还有 Mask R-CNN^[?]、RFCN^[?] 该类方法结合候选区域推荐 (region proposal, 简称 RP) 和 CNN 分类构成一个目标检测框架。首先用候选区域推荐的方法选出候选窗口，然后使用 CNN 将这些候选区域窗口分类，分为目标区域或者背景区域，一般最后使用极大值抑制的方法框选出最后的目标检测结果。该类方法在候选区域推荐的时候耗时过长，很难达到实时效果。另一类是一 YOLO^[?] 为代表的端到端 (End-to-End) 的方法，该类方法有 YOLO、YOLO v2^[?]、YOLO v3^[?]、Single Shot MultiBox Detector^[?](SSD)、Deconvolutional single shot detector^[?](DSSD) 等，这类方法无需候选区域推荐，而是简化了整个流程，使用网格的方法将图像划分成不同的窗口，然后使用卷积的方法优化滑动窗口的计算，大大加速了整个过程。这一类方法不提出候选区域，网络学习的物体特征并不精细，效果没有第一类方法好，但是胜在速度快，效果也非常优秀。

但是，无论是 YOLO 还是 R-CNN，他们都是使用基本的神经网络结构，我们可以看到深度学习的发展历程，效果也变得越来越好，但这往往要求网络够深，数据量够大，从而提取出数据更深层次、更抽象的特征，但这往往使得其对较小对象难以取得较好结果。

1.3 本文工作以及后续内容安排

本节首先介绍了本文的主要工作以及本文的创新内容，接着陈述本文结构，包含的章节和模块，并对各个章节做简要介绍。

1.3.1 本文工作

针对传统算法和基于深度学习的算法的各自优缺点和不足，我们首先提出了获取物体表面法线的算法，设计了一套硬件设备，并配套相应算法，可以获取物体表面精确细致的法线图。接着将物体法线图作为输入，验证了法线图作为零件缺陷检测输入的可能性和有效性。并且，本文还对传统算法做了深入研究，比较了不同特征应用于目标检测对的效果和性能，同时，通过对不同特征的组合，优化了算法的结果。本文还针对目标检测这一问题，专门优化了数据预处理、模型分类方法，同时针对精密零件中缺陷样本少，正负样本分布不均衡，缺陷目标较小等问题提出了相应解决方案，取得了良好效果。最后，本文设计和实验了基于深度学习的检测方法，与传统方法进行比较。针对这一问题，本文首先设计和实现了提取零件法线图的方法，并对基于传统算法和基于深度学习算法进行了深入研究，将法线图作为输入结合传统算法和深度学习算法进行缺陷的检测，取得了显著的效果。本文的主要工作和创新有：

1. 零件表面法线信息获取硬件设备的设计。现有的扫描物体表现信息的视觉扫描系统一般基于两种采集模式：扫描仪或照相机。它们主要有两个缺陷，首先是信息采集不全，一般仅能采集色彩信息，其次是质量不稳定，难以达到实用标准。本文设计了一套专门扫描物体表面法线的设备，包括遮光模块、平台模块、灯光模块、拍照模块和控制模块五大模块，可以获得精准方向光源下的高质量照片。
2. 材质表面法线信息计算的方法。本文使用设计的硬件设备，提出了使用不同方向光源下照片来计算物体法线的方法。同时，为了提高拍摄照片的质量从而优化法线信息的精度，本文还对相机做了色彩校正、白平衡校正、

畸变校正等校正，并针对方向光源设计了光线补偿算法，从而得到零件精细而平滑的表面法线图。

3. 将法线图应用到基于传统的缺陷检测算法中。将法线图和不同方向光源照片作为输入对比，并分析和对比了法线图的 Haar-like 特征、图像梯度特征、LBP 特征、GLCM 特征、HOG 特征等不同特征的分类效果。
4. 本文对于传统的缺陷检测算法，针对金属零件缺陷检测这一课题，研究了数据预处理方法、数据增强方法、支持向量机模型、Adaboost^[?] 算法、Cascade^[?] 算法等其他学习算法，首先解决了精密仪器缺陷零件较少，数据不平衡的问题，同时使用集成学习的方法，不仅提高了模型的漏检率，同时提高了模型的误检率，得到了一个符合目标要求的方法。
5. 本文还是针对这一问题将法线图作为输入，实验了基于深度学习的方法，并将其与我们设计的传统方法进行了对比。由于网络深度的要求，深度学习往往会提取图像“高层次”的特征，然而针对检测目标较小的问题，或许较高层次的特征并不一定比基于传统方法设计和提取的特征更有效。并且，由于深度模型的复杂性，不论是训练速度还是检测速度上，传统方法都有明显优势。

1.3.2 后续内容安排

本文主要包括绪论、零件表面法线提取、基于传统算法的缺陷检测、基于深度学习的缺陷检测、总结和展望五个章节。本章为绪论，主要介绍了本文工作动机、目的和意义，以及本文的研究研究课题的研究背景，以及相关工作现状，介绍相关工作的各自优缺点和不足，从而引出本文工作，并对本文作品内容作出简要概括，总结了本文的主要创新内容，最后在本小节陈述本文结构以及各部分内容。

第二章为零件表面发现提取，首先介绍我们为提取物体表面法线设计和实现的硬件设备，接着描述相关表面提取算法。提取表面法线的算法中又包括对拍摄照片的校正算法如白平衡校正、色彩校正、畸变校正等以及对光线对的补偿算法，最后才是我们设计的法线提取算法。

第三章给出了基于传统算法的缺陷检测内容，详细介绍了我们如何将法线作为原始图片，如何提取不同的特征作为模型的输入，如何对各种特征做组合，如何针对精密零件的缺陷检测这一问题固有的难点做数据预处理和数据增强，以及如何设计传统分类模型，并对其进行训练和做缺陷检测，最后给出了

不同特征的实现效果，以及最红模型的实验结果。

第四章的内容组织结构类似于第三章，只不过是在使用分类模型的时候我们使用的是深度卷积神经网络。该章还介绍了深度神经网络的基本概念以及我们使用的模型的架构，对模型的优化以及针对这一问题的处理方法。最后给出了实验结果并和基于传统的检测方法进行了对比和总结。

第五章对本文的工作进行了总结，并提出了本文工作中存在的不足以及可以改进的地方，对今后的工作内容进行了展望。

第二章 表面法线提取

本章主要介绍如何提取金属零件的表面法线信息。首先需要介绍的是，我们设计和实现的一款专门为了提取材质表面信息的硬件设备，从而为提取金属零件表面信息提供了数据。该设备通过巧妙地设计，能够精确地获取到表面法线算法所需要的照片。接下来我们将描述如何通过该材质表面信息扫描设备获得到的数据来计算材质法线，这一算法又包括计算前对输入照片进行的包括白平衡校正、色彩校正等校正算法，又包括了通过这些照片获得零件表面法线的算法。

2.1 硬件设备的设计与实现

为了精确地计算表面法线，我们的算法要求获取特定入射角度光源下的材质表面照片，为此，本文配合算法设计了一款精密的捕捉照片的扫描设备，该设备通过良好的物理设计和与之配合的软件驱动配合，可以完成对算法所需数据的采集。整个设备结构如图 2-1 该设备的硬件系统主要包含遮光模块、平台模块、灯光模块、拍照模块和控制模块五个模块，每个模块设计和功能介绍如下：

1. 遮光模块。为了捕捉特定入射角度光源下的照片，我们需要提供一个尽可能暗的拍摄环境，为此我们将整个设备设计在一个合适大小 ($60cm \times 60cm \times 110cm$) 的遮光箱内。为了整个设备的稳固和耐用，遮光箱用金属材质构建框架，并在框架上面覆盖了特定厚度的黑色塑料板，形成一个密闭的环境。同时，考虑到遮光箱本身材质会对光产生反射等影响灯光环境的现象，我们通过不同的尝试找到了一种黑绒布材料覆盖在遮光箱内部，从而大大削弱了遮光箱本身材质对光照环境的影响。
2. 平台模块主要由一块固定在距遮光箱底部 $10cm$ 处的一个平台组成，用来放置需要扫描的材质。该平台由一块白色亚力克匀光板构成。该匀光板放置物体的一面为磨砂面，反射效果较弱，从而削弱对入射光源的反射从而减小对灯光环境的影响。另一面为透明面，允许底部灯光透过照在平台物体

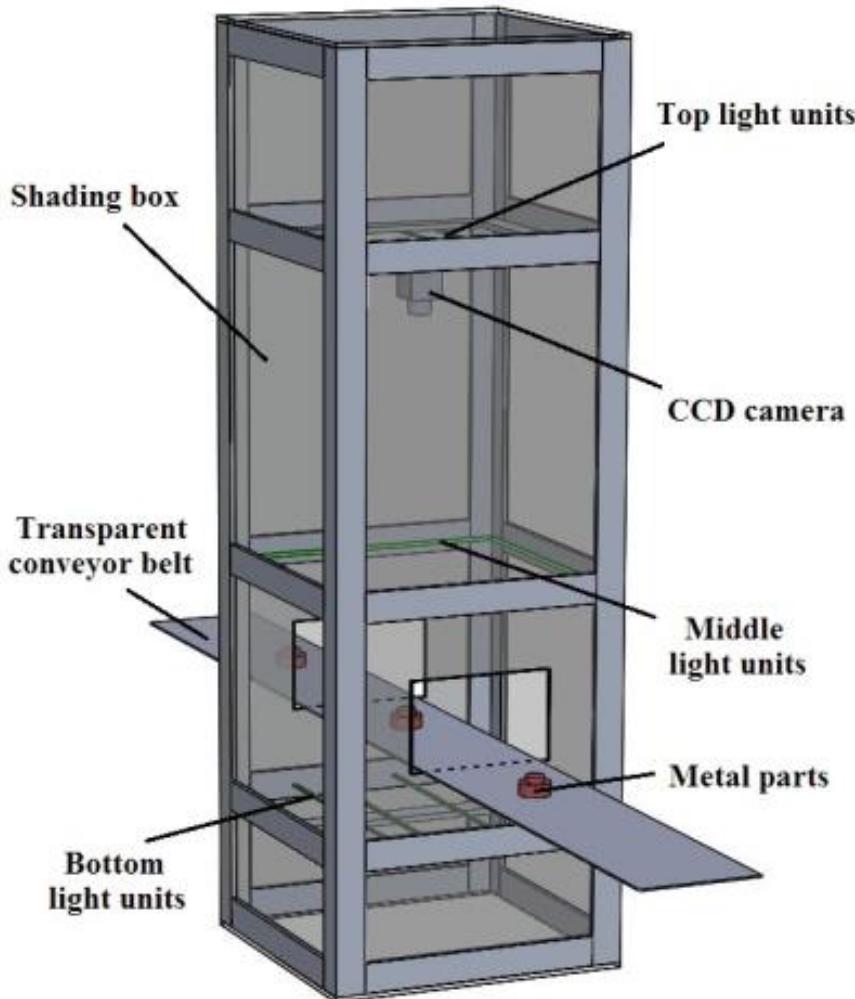


图 2-1: 设备架构图

上，从而计算透明度。

3. 灯光模块：灯光模块的作用是为拍摄提供特定的灯光环境，要求模块的灯光为尽可能满足一定角度的入射平行光源，同时尽可能的避免光源的散射和衰减导致拍摄图片获取的信息不准确。灯光模块由周边四组光源、顶部光源以及覆盖其上的 CPL 滤光膜。所有的光源均采用 LED 灯带组成，在保证了灯光质量的条件下保证了光源质量。周边四组光源分别从放置物体平台的前（东）、后（西）、左（南）、右（北）四个方向以 45° 角照射平台中央。周边光源设置在平台模块上 $20cm$ 处的遮光箱四周，从而并且通过一个金属槽卡住 LED 灯带，保证了 45° 的入射角度。拍摄时，当东部灯组亮起来的时候，拍摄出来的照片会呈现出上半部分比较亮，下半部分比较亮

的特征，西部灯组亮起时则相反；当南部灯组亮起来的时候，拍照时照片的左半部分会比较亮，右半部分比较暗，同样和北部灯组亮起来时拍摄得到的照片呈相反现象。顶部灯组由四条 LED 灯带组成，固定在相机镜头下 3cm 处，以垂直的方向照射平台。灯带的方向和拍照模块的配合放置，协调工作。CPL 滤光膜具有选择地让某个方向的光线通过的功能，同时能够配合数码相机的偏光镜头过滤高光。因此我们在所有的灯带上均放置了滤光膜，滤光膜的使用需要搭配相机的位置以及相机的偏光镜使用。

4. 拍照模块：拍照模块包括一款单反相机和与之配合的偏光镜镜头。本文所有实验是用的相机都是 Nikon D9000 拍摄出来的照片。相机固定于遮光箱顶部正中间位置，使镜头正对着平台正中心拍摄。同时，相机镜头下方有一个偏光镜头，这一偏光镜头通过一个步进电机控制的滑轮控制，通过程序控制在需要的时候转动滑轮使偏光镜处于镜头下方，在不需要的时候转回其他位置，从而和灯光模块的滤光膜配合使用。
5. 控制模块：控制模块主要由单片机和与连接的继电器以及步进电机和其配套部分组成。通过单片机连接继电器控制灯光，给单片机发不同的信号，单片机输出不同的高低电平组合控制继电器的通断来控制灯光的开闭。同时单片机也用来控制用来转动偏光镜的步进电机，根据是否需要偏光镜来调整偏光镜的位置。

完成所有模块的设计和实现之后，就可以使用我们的硬件设备对材质进行扫描，获取不同方向光源以及顶部光源下的拍照结果了。结果如图 2-2 所示，其中，图 2-2 中图组 (a) 为使用了偏光镜加滤光膜滤出了高光之后的结果，图 2-2 中图组 (b) 为没有滤除高光得到的结果。为了保证图片的质量，所有相机拍摄出来的照片都以 tif 格式存储，命名方式则根据灯光组的名称，用上北下南左西右东的方式来命名。如图 2-2 中图组 (a) 所示，在相机使用滤光镜头的时候，北部灯光组亮起，图片上半部分较亮底部较暗，记为 *Image_N1*，结果为图 2-2 中图组 (a) 种左上角标记了 *North* 的图片，南部灯光组亮起，下半部分亮上半部分暗的图片则记为 *Image_S1*，展示在为图 2-2 中图组 (a) 种左上角标记了 *South* 的图片，西部灯光组亮起，照片左边更亮右边更暗，记为 *Image_W1*，显示为图 2-2 中图组 (a) 种左上角标记了 *West* 的图片，东部灯光组亮起，图片明暗与西部灯光组亮起时相反，记为 *Image_E1*，为图 2-2 中图组 (a) 种左上角标记了 *East* 的图片。顶部灯光组亮起时，拍摄得到的照片记为 *Image_T1*，为图中标记了 *Top* 的图片。相机不使用滤光镜时，将对应的图片分别命名为

Image_N2, Image_S2, Image_W2, Image_E2, Image_T2。本文中所有拍摄照片的命名方式全部遵循这一规则，以后将不在赘述。

2.2 算法设计与实现

在本节中我们将介绍如何根据上一章节我们设计和实现的硬件设备扫描得到的图像数据来生成最终我们所需要的图像表面法线图。通过我们的硬件设计，最终我们将获得在不同入射角灯光照射下的材质图片。由于相机本身成像原理以及相机参数的设定如白平衡设定、曝光时长、光圈大小、相机焦距等，在成像过程中难免会产生与真实值不同的偏差，同时不同的镜头本身也会产生不同的畸变，比如色彩的畸变或者几何的畸变。因此，我们首先对相机的到的照片进行校正，包括白平衡校正、颜色校正以及畸变校正等。在进行了响应的矫正后，我们得到了与真实值更接近的图片，并以此为基础，推导、设计和实现了我们自己的材质表面法线恢复算法。所以，本小节讲先从各种校正算法的设计和实现讲起，最后描述表面法线恢复算法。

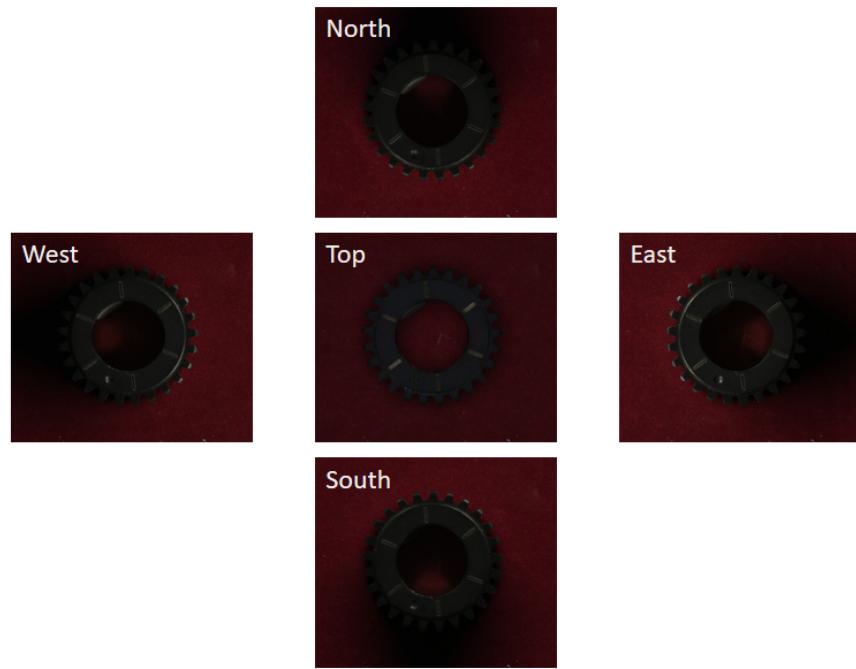
2.2.1 白平衡校正

白平衡校正主要指在图像处理过程中，对原本材质为白色的物体图像进行色彩还原，去除外部光源色温的影响，使得原来是白色的物体在成像之后的图片中也是白色。在本次试验的目的即是去除 LED 灯带本身所带色温。为了避免光源本身对成像色彩的影响，我们所选择的 LED 灯带型号发光颜色为白色，配合上我们采用的白平衡校正算法，能够较好的对白平衡进行校正，提高材质表面信息提取的准确度。

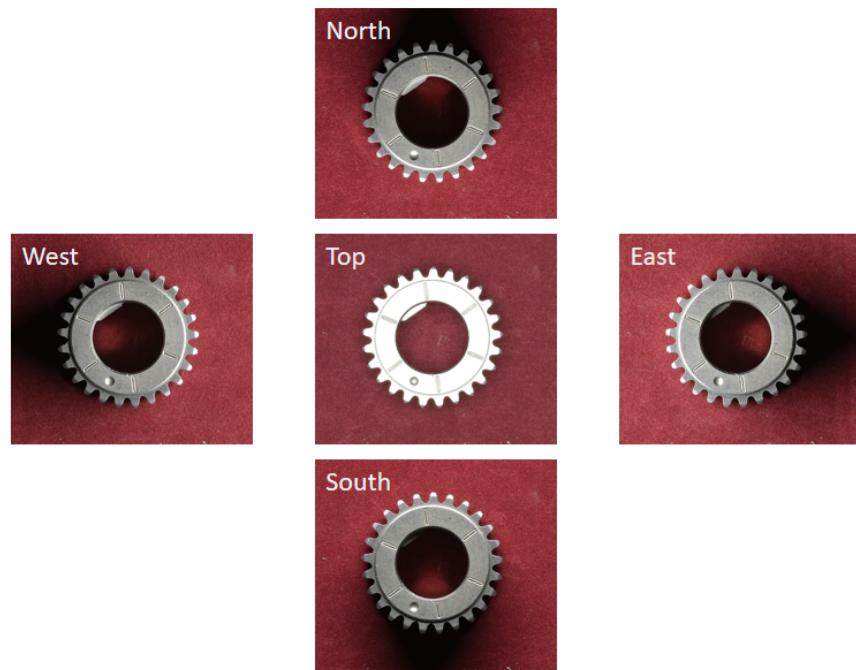
白平衡校正中有两个最基础的理论：灰度世界理论和完美反射理论。大部分白平衡算法都是基于这两个理论进行改进和优化，比如 Xiaoqiang Li^[?] 提出的算法就是基于灰度世界理论对灰度世界算法改进实现的白平衡算法，Lam 和 Edmun Y^[?] 将两者结合，提出和实现了一种自动白平衡算法。

本文算法在 Weng^[?] 等人算法的基础上使用白卡改进了白点检测的准确度。通过定位白卡的位置，读取白卡的内容，根据白卡的内容来计算色温，对图像补色来进行补偿，将整张图片的色温调整到白色。此算法中，我们需要对相机在不同的光源下分别做白平衡校正。

该算法仍主要分为两步，第一步是白点检测，第二步是白点调整。在第一



(a)



(b)

图 2-2: 扫描获取图片结果:(a) 表示添加滤光镜得到的结果, (b) 表示没有滤光镜得到的结果

步中，首先将整个图片从 RGB 颜色空间转换为 YC_bC_r 颜色空间。 RGB 颜色空间转换到 YC_bC_r 空间，如公式 (2-1) 所示。其中 R, G, B 分别代表原来 RGB 色彩空间中三个通道的值， Y, C_b, C_r 分别代表了 YC_bC_r 空间各个通道的值。

$$\begin{aligned} Y &= (0.256 * R) + (0.504 * G) + (0.098 * B) + 16 \\ C_b &= -(0.148 * R) - (0.291 * G) + (0.439 * B) + 128 \\ C_r &= (0.439 * R) - (0.368 * G) - (0.071 * B) + 128 \end{aligned} \quad (2-1)$$

论文中定义了一个 *near-white* 区域，并将该区域所有像素值作为白点的候选，通过一系列约束条件求解一个最优的白点集合当做白点。这一方法首先要统计所有 *near-white*，接着要进行一系列复杂的变换求解约束条件，大大限制了算法的速度。并且，约束条件以及 *near-white* 区域的选取都需要设定阈值，在不同的光照情况下可能需要人为调整参数来达到最优解，这在我们这一全自动化系统中是不可行的。因此，本文使用白卡来辅助确定白点。在本文算法中第一步首先对照片中的白卡进行定位和检测，识别出自白卡所在位置以及其中白色区域当做白点候选。并且在接下来的筛选过程中仅仅排除掉每一白色区域的离群点，只需要统计白卡中白色区域的 C_b, C_r 分量，并计算每个区域 C_b, C_r 分量的平均值 M_b, M_r ，然后去除掉每一白色区域的离群点（和均值差距比较大的值）即可。本文大大降低了白点统计的复杂度，从而提高了速度。并且在检测白点时加入辅助硬件设备白卡，确定了白点区域的存在以及准确度，因此提高了算法的准确性。

在确定了白点之后，算法使用 Von Kries 模型^[?] 来调整图片。通道增益可以从白点的均值得出来。为了使整张图片具有相同的亮度，算法使用亮度最大的值来得到通道增益。通道增益 $R_{gain}, G_{gain}, B_{gain}$ 可以根据一下公式 (2-2) 计算得到。

$$\begin{aligned} R_{gain} &= Y_{max}/R_{avew} \\ G_{gain} &= Y_{max}/G_{avew} \\ B_{gain} &= Y_{max}/B_{avew} \end{aligned} \quad (2-2)$$

其中， $R_{avew}, G_{avew}, B_{avew}$ 分别代表选中的白点 Red, Green, Blue 三个通道像素值的均值。 Y_{max} 表示整张图片的最大亮度值。根据 Von Kries 模型，照片中每一

个像素的像素值可以根据公式(2-3)调整。

$$\begin{aligned} R_{gain} &= Y_{max}/R_{avew} \\ G_{gain} &= Y_{max}/G_{avew} \\ B_{gain} &= Y_{max}/B_{avew} \end{aligned} \quad (2-3)$$

其中 R, G, B 代表原图片中三个通道的像素值, R', G', B' 则代表了调整之后的图片中三个通道的像素值。

2.2.2 颜色校正

颜色校正又被称为色彩校正, 顾名思义, 即是对拍摄结果照片的颜色调整。人类对颜色的感知在不同光照的条件下能够得到恒定不变的物体颜色, 被称为色恒定性。然而, 成像系统却不同于人类, 它们并不具有这一特征, 所获图像的色彩往往受光源、成像物体反射率和成像系统的光谱响应函数共同影响, 因此通常相机在不同的光照条件下对同一物体所捕获的图像也往往不同, 并且跟人眼中观察到的色彩有一定色差。为了避免成像系统探测器光谱响应函数与 CIE 标准观察者色匹配函数的偏差所带来的影响, 必须对图像进行色彩校正, 以保证相机捕获的图像能够正确的感知物体的本来颜色, 防止因为拍摄存在色差而对算法的计算造成损失。

对于颜色校正目前已有大量的研究工作, 并且种类繁多, 其中主要的方法有灰度直方图统计法、灰度平衡算法等。Porikli^[?] 使用了距离矩阵和模型函数的方法来对相机间的颜色进行矫正, Jinlong Lin^[?] 等人利用彩色图像的边缘色度来进行颜色校正, 提出了一种基于边缘检测的色偏校正算法。Hao Yu, Jian Wang^[?] 等人则是选择在 Lab 空间对颜色进行校正。

同样, 基于我们已有固定设备提供的单一实验环境, 我们并不需要在多种光源以及复杂地条件下实施适应性的色彩校正。只需要针对我们实验环境光源条件下对颜色进行校正。本文继承白平衡校正的硬件辅助, 使用标准色卡帮助校正。本文中算法首先将一张标准色卡放在实验环境的不同光源下拍摄一组照片, 然后对标准色卡进行识别, 识别出色卡中不同区域的像素信息, 并与标准色卡区域对应的真实值对比, 得到色彩偏差曲线, 然后对相机捕捉到的材质图像按照该函数曲线进行颜色校正。

本文所做实验基于标准 24 色色卡。对于一块标准的 24 色色卡, 每一块色

卡中都包含 24 种确定 R, G, B 值的色彩信息，分别位于 24 个矩形块中，我们将第 i 种色彩的标准信息记作 C_{org}^i ，而在不同的光照和相机成像函数的条件下，拍摄出来的照片的像素值会有差异。标准色卡有固定的形状，因此可以使用边缘检测并提取直线之后很简单的得到色卡中不同颜色区域。对不同的区域 i ，我们统计该区域的像素均值记为 C_{cap}^i 。显然，这里的 C_{org}^i 和 C_{cap}^i 都是三维向量，包含着 RGB 通道的像素值信息。显然，根据这些我们可以得到 24 个颜色点之间的对应关系，接着拟合一个函数 $f(R, G, B)$ ，该函数的输入为拍摄得到的照片的 R, G, B 通道像素值，输出矫正后的像素通道值。

2.2.3 畸变校正

由于相机镜头中所使用的成像镜头固有的透镜效应，在成像过程中会对真实的景象产生一种透视失真，即被称为镜头畸变。随着机械技术的日益发展，相机镜头制造技术日益精良，从镜头的材质以及光学设计入手对镜头畸变不断改善，试图消除镜头畸变。但是，即使是最完善的设计，最考究的工艺，搭配上最高质量的光学玻璃制造出来的镜片，也会产生畸变，想要消除镜头畸变，必须借助镜头畸变校正算法进一步处理。

在计算机视觉领域，畸变校正算法已经发展的非常成熟。其中最成功的是张正友教授提出的张氏标定法^[?]，现已被广泛应用于相机标定和畸变校正中。Weng J, Cohen P^[?] 等人中提出了先估计参数后迭代校正的两步校正方法，Heikkila J 和 Silven O^[?] 等人则在 Weng J 等人算法的基础上经行了扩展，提出了四步校正算法，该算法使用了额外的步骤补偿由于镜头的圆形特性引起的图像失真问题。综合这些参考文献，不难发现在畸变校正过程中最重要的步骤就是相机的参数估计。相机的参数又分为内参、外参和畸变参数。显而易见，其中畸变参数是畸变校正中最重要的参数。相机的畸变参数主要包括径向畸变系数和切向畸变系数。径向畸变的产生原因是光线在远离透镜中心的地方比靠近中心的地方更加弯曲，是由于透镜本身的凹凸性质导致的；而切向畸变产生的原因是成像透镜本身并不完全平行于图像平面，主要发生于成像仪被粘贴在摄像机的时候。畸变校正算法可以通过计算畸变系数，然后计算畸变系数对于图像的作用矩阵，通过对相机拍摄得到的照片应用与畸变作用矩阵相反的变换，进而得到消除畸变的矫正图像。本文采用张氏标定法对相机标定。具体原理以及方法在论文中有具体的描述，由于本文篇幅所限，在此不展开具体描述。

在张氏标定法只关注径向畸变。实际情况中径向畸变较小，所以可以用主

点 (principle point) 周围的泰勒级数展开的前几项进行描述。在张氏标定法中，选择了用泰勒级数的前两项来确定径向畸变的畸变系数。首先我们使用需要标定的相机拍摄来拍摄不同组棋盘格的照片。在拍摄过程中，我们保证相机位置不变，在不同的角度和位置下拍摄不同的棋盘格照片来进行相机标定。通过 n 张照片中提取棋盘格中的 m 个角点，可以得到 $2mn$ 个等式，对这 $2mn$ 个等式使用最小二乘法求解方程，可以得到最优的径向畸变参数 $k = [k_1, k_2]$ ，最后结合相机的内参和外参，联合进行极大似然估计从而得到更准确地畸变参数。并将计算得到的参数保存起来，在每次需要进行畸变校正的时候通过公式 (2-4) 对其进行畸变校正。

$$\begin{aligned} p &= p' + (p' - p_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \\ q &= q' + (q' - q_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \end{aligned} \quad (2-4)$$

其中， (p, q) 代表没有产生畸变时像素的坐标值，即进行畸变校正后的像素坐标， (p', q') 表示相机拍摄得到的照片中像素的坐标， k_1, k_2 表示张氏标定法中得到的相机畸变参数， (p_0, q_0) 为主点的坐标。本算法主要目的是对相机进行畸变校正，放置相机镜头畸变对生成图片的影响，如果相机本身已经进行了畸变校正，这一步可以省略。

2.2.4 光线补偿

尽管在设计硬件的时候使用了 CPL 等控制光线的设施，但是光线损失还是无法避免的。在拍摄的过程中，仍然会存在光线不均匀的情况，尤其是当需要拍摄的材质本身比较大，设备设计的也比较大的时候，这一光线衰减会更加明显，严重影响了我们稍后算法的计算。并且，不同灯光组拍摄得到的照片都存在着光线衰减的问题。因此我们需要对拍摄照片做进一步的光线补偿，从而减少由于光线衰减造成的损失。

已有的光照补偿算法如 [? ? ? ? ? ? ?] 中所提到的离散余弦变换，局部自适应模板匹配等技术都从不同的角度尝试对光照进行补偿。本文参考了这些方法，选择通过模板以及函数方法进行光线补偿。

为了计算光照补偿，我们首先统计了不同颜色以及不同灰度的材质在实验环境中所表现出来的光线损失。结果如图 2-5，其中 (a), (b), (c), (d) 分别代表了不同材质的光线图，结果表明，不同材质具有不同的光线损失程度，并且，往往材质灰度越低，光线损失越小，反之，材质亮度越大，光线的损失程度也越

大。,,

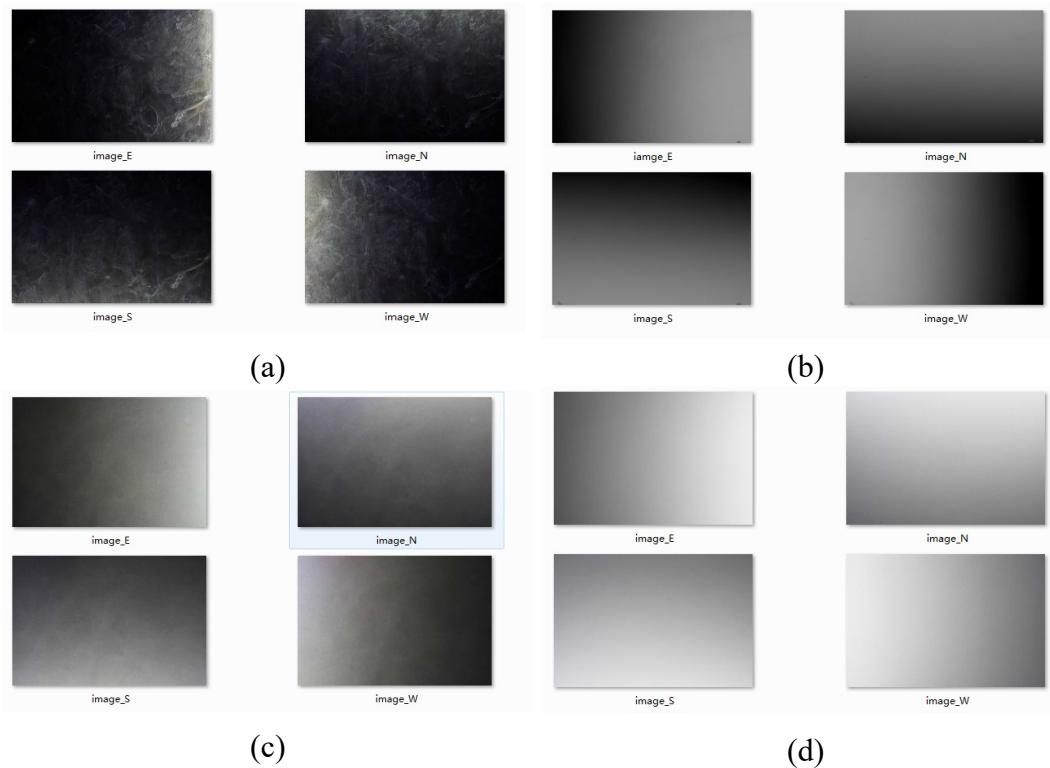


图 2-3: 不同材质光线损失图

同时，为了更直观的展现光线衰减与光源的关系，我们统计了图像和光源方向的衰减情况，光线衰减对比图（平行光源方向）所示，以及图像和光源方向垂直的光线衰减情况，如光线衰减对比图（垂直光源方向）所示。第一个图表示的是平行于光源方向统计的像素信息的亮度统计值，第二个图表示的是垂直于光源方向统计得到的结果图。 (E) , (W) , (N) , (S) 则分别表示了东西北南四个方向光源下的统计结果图。不同图中不同颜色的曲线 $(a)(b)(c)(d)$ 则代表了不同材质光线衰减程度图中对应命名图组中不同材质的统计结果。从图中可以明显看出，光线衰减在垂直于光源方向的地方相对较弱，基本难以发觉，但是在沿着平行于光源方向的方向变化极为明显，呈现出一种线性衰减趋势。

考虑到我们的实验环境中，相机位置以及相机参数均固定不变，同时，光源也就是我们的灯光模组在固定好了之后也是固定不变的。因此，我们提出是否可以在扫描具体材质之前先得到一组光照补偿模板，然后根据模板在针对不同的材质作出适当调整来补偿光线信息。这就是本文提出的模板方法的简单动机和原理。又因为我们只需要对光线信息进行调整，并不需要设计照片的具体颜色，因此我们直接在亮度信息上做光线补偿，并且模板中也只保留亮度信

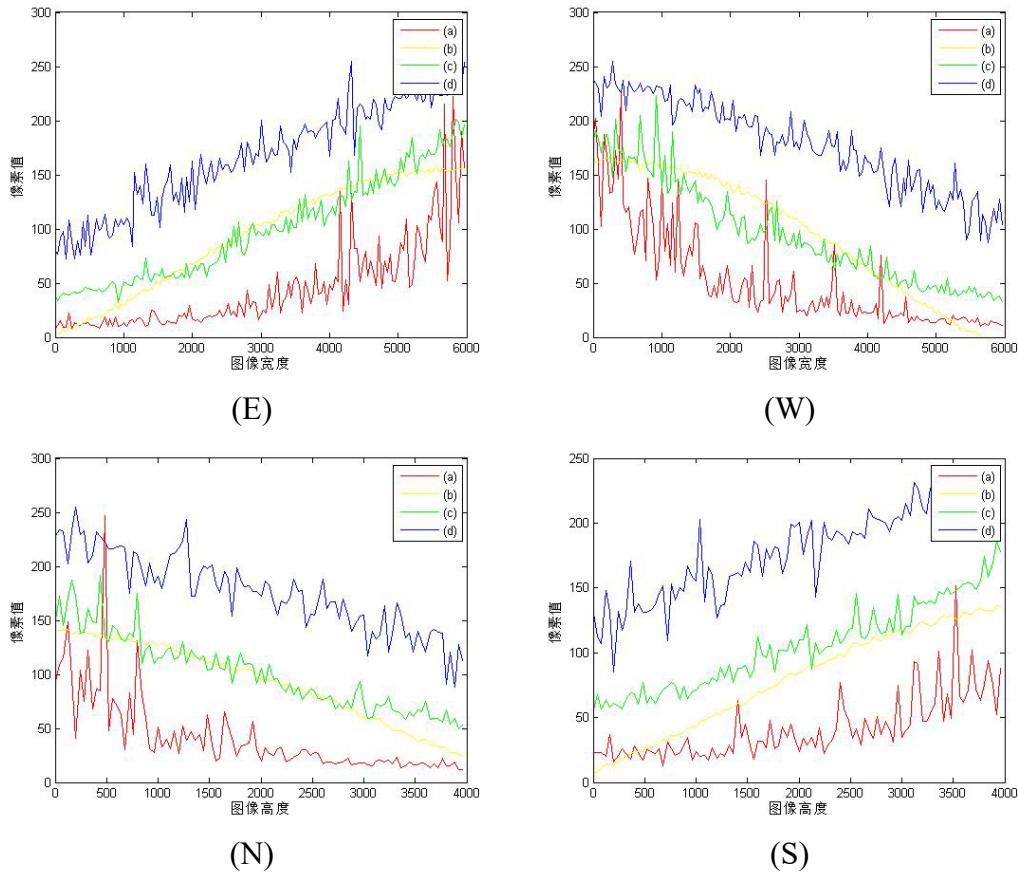


图 2-4: 光线衰减对比图 (平行光源方向)

息。图像的亮度有多种表示方法，但是为了和整个流程统一方便计算，我们使用 YC_bC_r 颜色空间中的 Y 分量来对亮度信息进行调整。

首先，先拍摄一组用来做光线补偿的校正图片，在光线补偿校正图片的时候，相机拍摄物体为一整块匀光板。匀光板具有良好的光源分散的性质，不容易造成光线损失。并且，在拍照的时候，将顶部灯光组拍摄得到的照片作为没有损失的图片，称之为标准图片，并标记为 $Base$ 。然后依次拍摄东部灯光组，西部灯光组，南部灯光组和北部灯光组光源下的待校正照片，分别标记为 $Adjust_E$, $Adjust_W$, $Adjust_S$, $Adjust_N$. 为了方便计算和节省存储空间，我们只保留他们的亮度信息，即 Y 通道的值，记为 $Base^L$, $Adjust_E^L$, $Adjust_W^L$, $Adjust_S^L$, $Adjust_N^L$. 使用 $Base^L$ 减去各个方向上的亮度信息，就可以得到不同方向光源上相对于 $Base$ 的光线衰减信息。为了防止出现负值的情况，我们对公式做了一定调整，保证了补偿信息的非负性。调整的方式是为整个亮度增加一个偏差 α ，我们采用顶部图片亮度与东南西北四个方向光源亮度对应差值最小的 0.1% 的值的均值作为偏差值，这在一定程度上保持了图片的稳定性，并且保证了光照

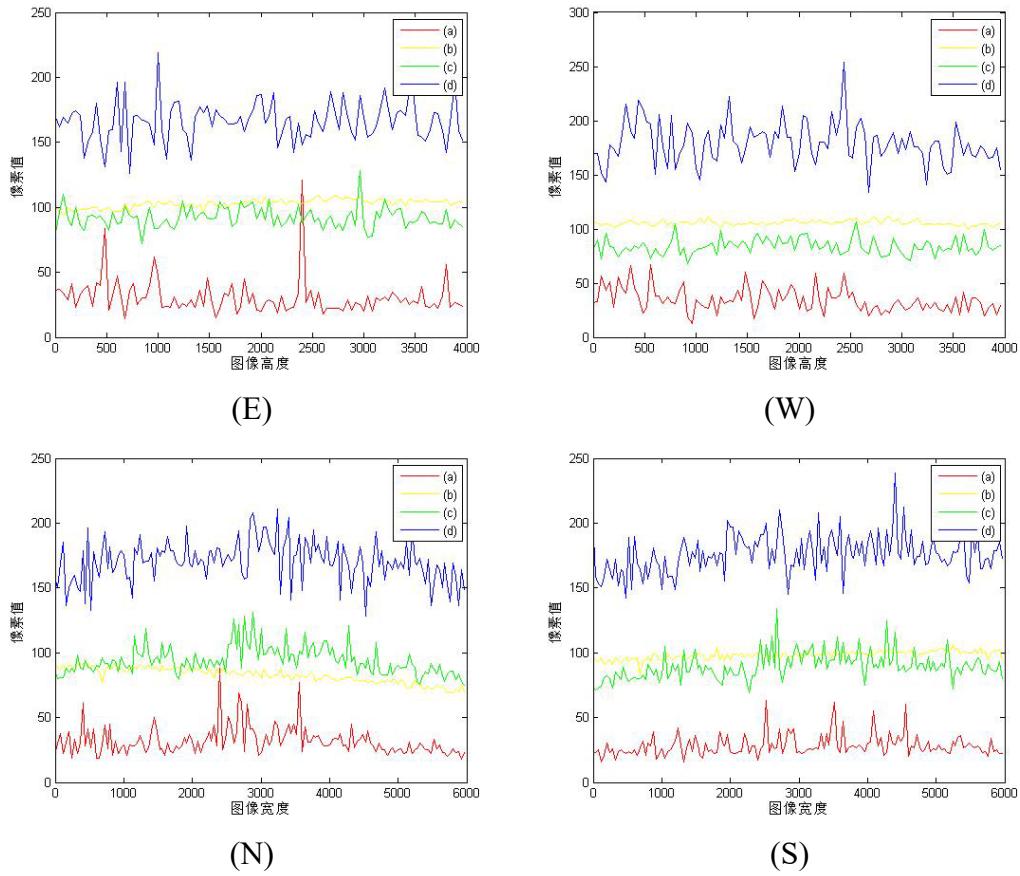


图 2-5: 光线衰减对比图 (垂直光源方向)

补偿没有负值。当然，如果光照补偿信息加上偏差值仍然小于 0，那么直接将其置为 0，认为此处没有损失即可。最终得到的补偿如公式 (2-5)。

$$Comp_d = \max(0, Base^L - Adjust_d^L + \alpha), \quad d \in \{E, W, S, N\} \quad (2-5)$$

其中 $Comp_d$ 表示对应方向光源下的补偿信息， α 表示偏差，计算方法如公式 (2-6)。

$$\alpha = \frac{1}{N_{min}} \sum_i^{N_{min}} Y_i, \quad Y_i \in NMIN \quad (2-6)$$

其中 $NMIN$ 表示集合对应偏差值的最小的 N_{min} 个数的集合， $N_{min} = N \times 0.001, N$ 表示图片的偏差值的总数量。

考虑到前面统计数据所表现出来的不同亮度材质的光线损失程度会由于材质不同而不同，我们根据不同的材质信息对补偿进行调整。根据前面的统计规律，使用线性函数对光线补偿进行调整。当拍摄到材质进行光照补偿时，记不

同方向光源下的材质图片为 $Image_T$, 其中 $d \in \{E, W, S, N\}$, 其顶部光源组拍摄得到的照片为 $Image_T$, 我们用公式 (2-7) 计算对应的补偿:

$$CImage_d^L = Image_d^L + Comp_d * (\text{avg}(Image_T^L) / \text{avg}(Base^L)) * \lambda \quad (2-7)$$

其中 avg 表示求平均值, $Image_d^L$ 表示不同方向图片的亮度信息, $CImage_d^L$ 表示光线补偿后的图片的亮度信息, λ 为一个参数, 用来表示对光线进行补偿的强度。本文实验 λ 取 0.8。如果需要增大补偿强度, 可以将其调高。

2.2.5 法线计算

在进行法线计算之前, 我们需要首先对相机做白平衡校正、颜色校正、畸变校正以及光线校正, 保存校正需要的参数, 用来以后对相机捕捉到的照片进行校正, 保证拍摄照片的准确性。并且, 随着时间的推移, 不管是灯光条件还是相机本身都会产生一定的变化, 因此, 我们有必要定期的重新对相机进行这几种校正, 并更新校正参数, 保证我们的设备能够捕捉到最真实、最准确的材质信息。经验表明, 至少每三个月需要重新做一次校正, 更新校正参数。

有了正确的校正信息之后。即可以捕捉材质照片, 进行材质表面法线的计算。为了捕捉材质表面的照片, 首先要将扫描目标放置在 3.1 所描述的硬件设备的平台中央区域, 然后运行我们的软件捕捉功能, 该功能会控制硬件设备在不同的灯光组亮起的情况下对材质拍两次照片。第一次镜头上会放置滤光镜, 软件依次控制亮起顶部灯光组, 东部灯光组, 西部灯光组, 北部灯光组, 南部灯光组并拍摄照片, 并分别对拍摄出来的照片做校正, 保存结果为 $Image_T1, Image_E1, Image_W1, Image_N1, Image_S1$, 最后同时亮起顶部灯光组, 东部灯光组, 西部灯光组, 北部灯光组和南部灯光组并拍照和校正, 存储结果记为 $Image_F1$ 。此时镜头加入了滤光镜, 会过滤到图片的高光信息。接着软件会控制电机转动移开滤光镜, 并控制设备重复第一次拍照的情况, 在没有滤光镜的情况下对材质拍照和校正, 并存出结果记为 $Image_T2, Image_E2, Image_W2, Image_N2, Image_S2, Image_F2$ 。

为了计算材质法线, Ryan Clark^[2] 提出了一种没有光线损失情况下的算法。Ryan Clark 并没有将该算法系统化, 而只是简简单单的使用 photoshop 工具模拟了算法。该算法所得到的法线信息在细节和整体上都存在着极大问题。并且, 作者也没有考虑任何光线损失对于拍摄照片的影响。本文在算法层面上增

加了对照片光线损失的补偿，并且利用不同方向光照综合计算物体法线信息，得到的法线无论是整体还是局部都有较好效果。

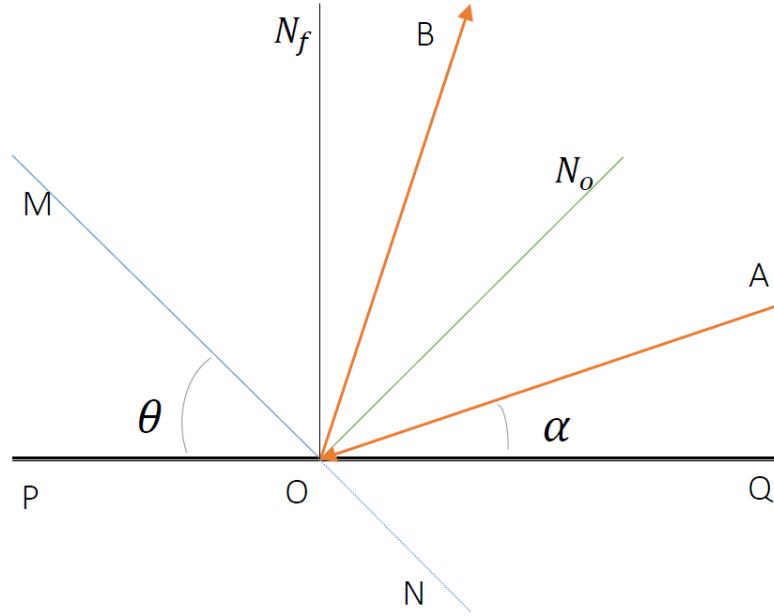


图 2-6: 光线反射图

如图 2-6 所示， PQ 表示水平面， MN 表示材质表面，实线部分 MO 为真实材质表面， ON 为虚线延长是为了方便观察和理解， \overrightarrow{AO} 为入射光， \overrightarrow{OB} 表示反射光， ON_f 表示底面的法线，垂直于 PQ ， ON_o 表示材质的法线，垂直于 MN 。 $\angle POM$ 为材质表面和水平面所成夹角的锐角，标记为 θ ， $\angle AOQ$ 为入射光与水平面的角度，标记为 α 。在本系统中，实验环境已经固定，其中相机镜头平行于水平面即 PQ ，不同方向光源的入射光与水平面的夹角也是固定的，我们设定的是 45° ，即 $\alpha = 45^\circ$ 。假设入射光线的强度为 I ，由 $\angle POM = \theta$ 和 $\angle AOQ = \alpha$ ，易得 $\angle BON_f = 2\theta + \alpha - 90^\circ$ 。根据光的反射原理，在不考虑其他介质对光的散射的条件下，反射到相机中的光线强度可以用公式 (2-8) 来计算：

$$I_{ref} = \lambda I * \cos(2\theta + \alpha - 90^\circ) \quad (2-8)$$

其中， λ 表示光线反射率，对于同一种光源，光线反射率不变。公式 (2-8) 中 α 的预设值 45° ，根据公式 (2-8)，理论上反射到相机中的光照强度只有材质表面的倾斜角度有关。光照强度反应在拍摄得到的图片的亮度信息中，因此

可以得出结论拍摄得到的照片的亮度信息必然与材质本身的倾斜角度有某种关系，即亮度信息和材质的法线有关系。根据这一推论，我们使用四个不同方向光源照射的图片来计算材质的法线信息。

通过前面章节的描述，拍摄照片并矫正之后我们将得到材质的照片 $Image_E1, Image_W1, Image_N1, Image_S1$ 。经过光线补偿算法对得到的输入进行光线补偿，并得到其亮度通道信息，存储为 $Image_E^L, Image_W^L, Image_N^L, Image_S^L$ 。接着我们创建两个新的三通道的图像 $NorthWest$ 和 $SouthEast$ 用来保存中间数据和方便计算，其中 $NorthWest$ 的 R 通道保存 $Image_W^L$ ， $NorthWest$ 的 G 通道保存 $Image_S^L$ ， $SouthEast$ 的 R 通道保存 $Image_E^L$ ， G 通道保存 $Image_N^L$ 。将 $NorthEast$ 的色阶调整到 $[0.0 - 0.5]$ ，将 $SouthEast$ 的色阶调整到 $[0.5 - 1.0]$ 。最后将 $NorthWest$ 和 $SouthEast$ 以叠加的方式混合成 $Normal_Temp1$ 。调整色阶的计算如公式 (2-9) 所示。

$$p' = p * (a - b) + b \quad (2-9)$$

其中 p 表示图像的像素值， a 为调整色阶的上界， b 为调整后色阶的下界， p' 为调整后的像素值。图像叠加混合成新的图像使用公式 (2-10) 计算。

$$Normal_Temp1 = 2 * NorthWest * SouthEast \quad (2-10)$$

这样，我们计算了法线信息的两 R 和 G 通道的信息，又由法线是一个归一化向量的性质，可以根据 R 和 G 通道的值来计算 B 通道的值。有 R 通道像素值为 r ， G 通道的值为 g ，则 B 通道的值 b 可以通过公式 (2-11) 计算。

$$b = 2.0 * \sqrt{1.0 - (r - 0.5)^2 + (g - 0.5)^2} - 1.0 \quad (2-11)$$

将 $Image_N2, Image_S2, Image_W2, Image_N2$ 也使用上述处理方式，可以得到有高光情况下的发现图 $Normal_Temp2$ 。为了得到更好地法线，我们还对得到的法线做了一些后期处理，处理方式包括亮度调节、对比度调节和褶皱去除。处理后的结果标记为 $Normal1, Normal2, Normal1$ 为镜头加上滤光镜和 CPL 拍摄出来的滤除高光之后的法线信息，整体光线信息较弱，在法线的细节部分较差，甚至有些细节会丢失。 $Normal2$ 为没有偏光镜和滤光膜条件下得到的法线信息，这一法线光线强烈，细节表现明显，但整体过强，个别强光处容易出现法线的跳跃现象。因此我们使用 alpha 融合来对两个法线信息，融合算法使

用公式(2-12)表示，从而得到更为平滑和细致的法线信息。

$$\text{Normal} = (\text{Normal1} + \text{Normal2}) * 0.5 \quad (2-12)$$

实验证明，这种使用alpha融合两种条件下的法线图既保留了细节信息又不会出现法线跳跃现象，具有良好的性质。最终的到的法线图结果如图2-10所示，其中(a)表示原图,(b)表示使用我们的设备和算法得到法线图。

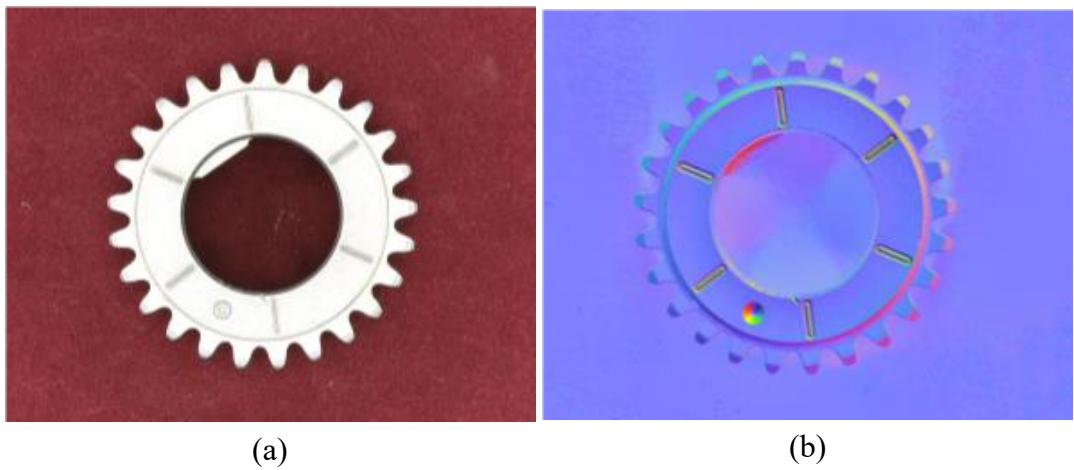


图 2-7: 法线提取结果图图

2.3 实验结果

本章设计的法线提取算法不仅仅只适用于金属表面零件缺陷检测这一主题，而是可以适用于任何材质、物体的表面法线信息的提取。因此我们将它独立出来做成一个法线提取模块，用来提取和计算材质的法线信息。本节将针对这一模块做介绍，首先介绍我们的硬件成果，接着描述我们的法线提取结果并与其它相关工作得到的结果进行对比。

2.3.1 硬件实现

按照3.1节中所描述的硬件设计，我们给出了实际的实现，并应用到了实际生产中，构成了一套硬件设备。该设备正如3.1节中所描述的拥有5大模块，这些模块被封装到一个白色的铝合金外壳中。实际结果如图2-8所示。其中，图2-8中(a)展示了这一设计的外观，是有一个金属合金构成的长方体，并

为了方便移动在下面安装了四个滚轮。在这个框架上面有一个开口，可以打开如图 2-8 中 (b) 所示，可以看到固定在顶部的相机以及相机下面的滤光镜和顶部灯光组，详细如图 2-8 中 (c) 所示。设备中间部分图 2-8 中 (a) 中可见把手部分可以打开，如图 2-8 中 (d) 所示，打开之后可以看到摆放材质的平台，需要扫描的材质信息就可以放在这个平台中间，同时，如图 2-8 中 (e) 所示，平台上方固定这不同方向的灯光组，以 45° 的照射角度照向平台中间。从图 2-8 中 (b),(c),(d),(e) 中可以看到，所有的内部金属部分都是用了吸光的黑色绒布覆盖，大大减少了对光照的影响。

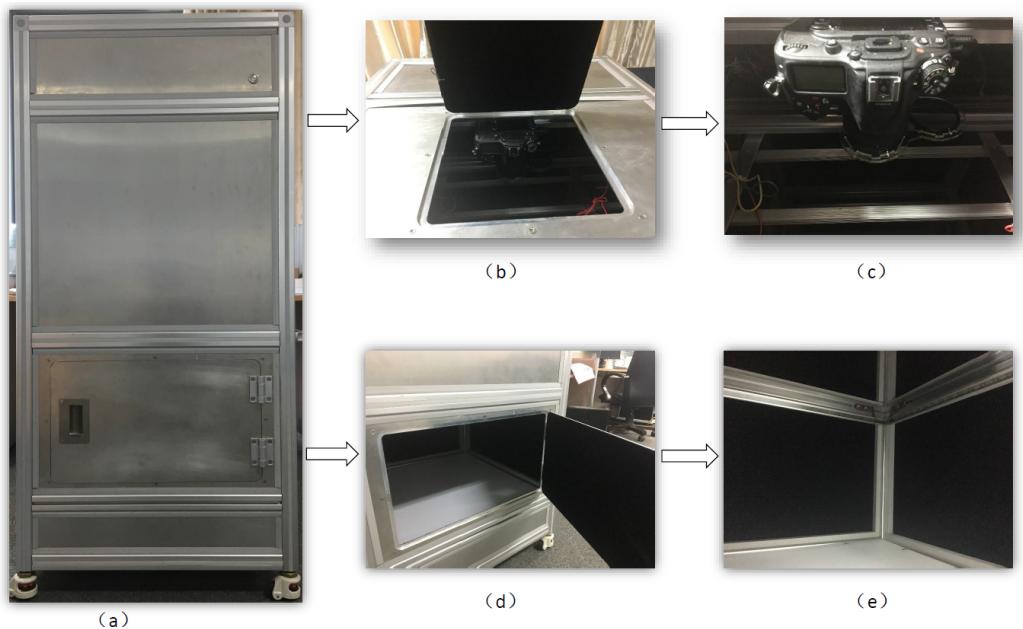


图 2-8: 硬件结果示意图

2.3.2 法线提取结果

在材质表面信息获取这一领域中，开始最早的是 Vizoo3D xTex，并且目前为止 Vizoo3D xTex 的技术仍然保持着业界领先水平。因此我们将我们的结果与其结果作对比。结果如图 2-9 所示，其中图 2-9 (a) 表示 Vizoo3D xTex 得到的法线结果，(b) 为我们扫描出来的相似材质的法线结果，从图中可以看出这两组图片的法线都比较平整不存在褶皱等问题，但是 (a) 中的法线信息在中心处出现了模糊现象，材质为同一种材质，却在中心处要比边缘部分模糊，我们据此推测这可能是由于光线不均或者衰减导致的。而在我们的法线计算过程中考虑到了光线不均和衰减造成的影响，并给予了一定的补偿，因此整幅图片中的

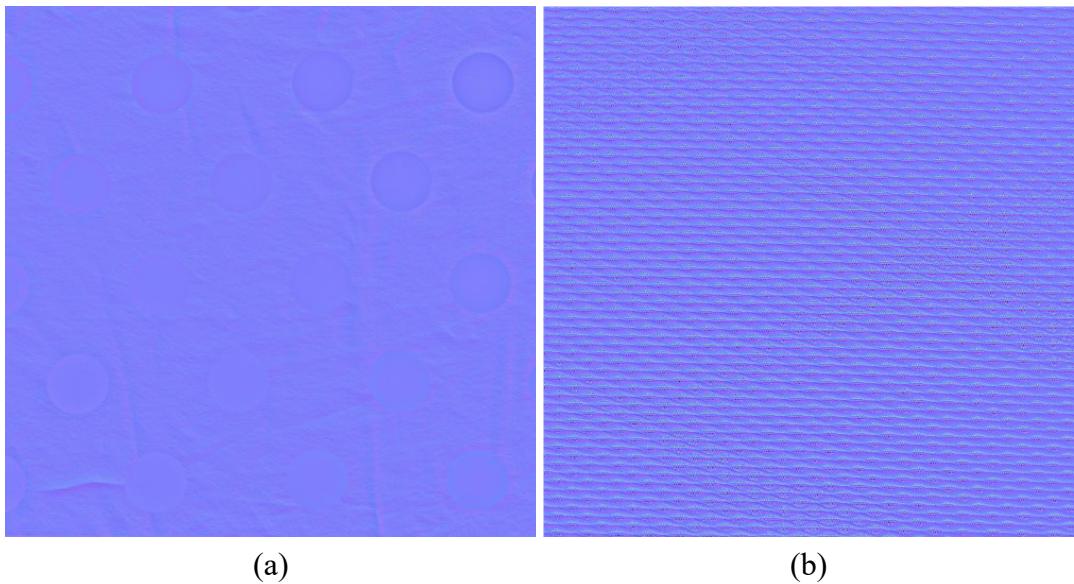


图 2-9: 法线结果对比图

法线呈现出一致效果。以下为我们扫描出来的不同材质的法线结果展示图 2-10, 图 2-10中 (a),(b),(c),(d) 分别表示提取的不同材质的法线, 可以看到对于不同的材质我们都能够得到优异的结果。

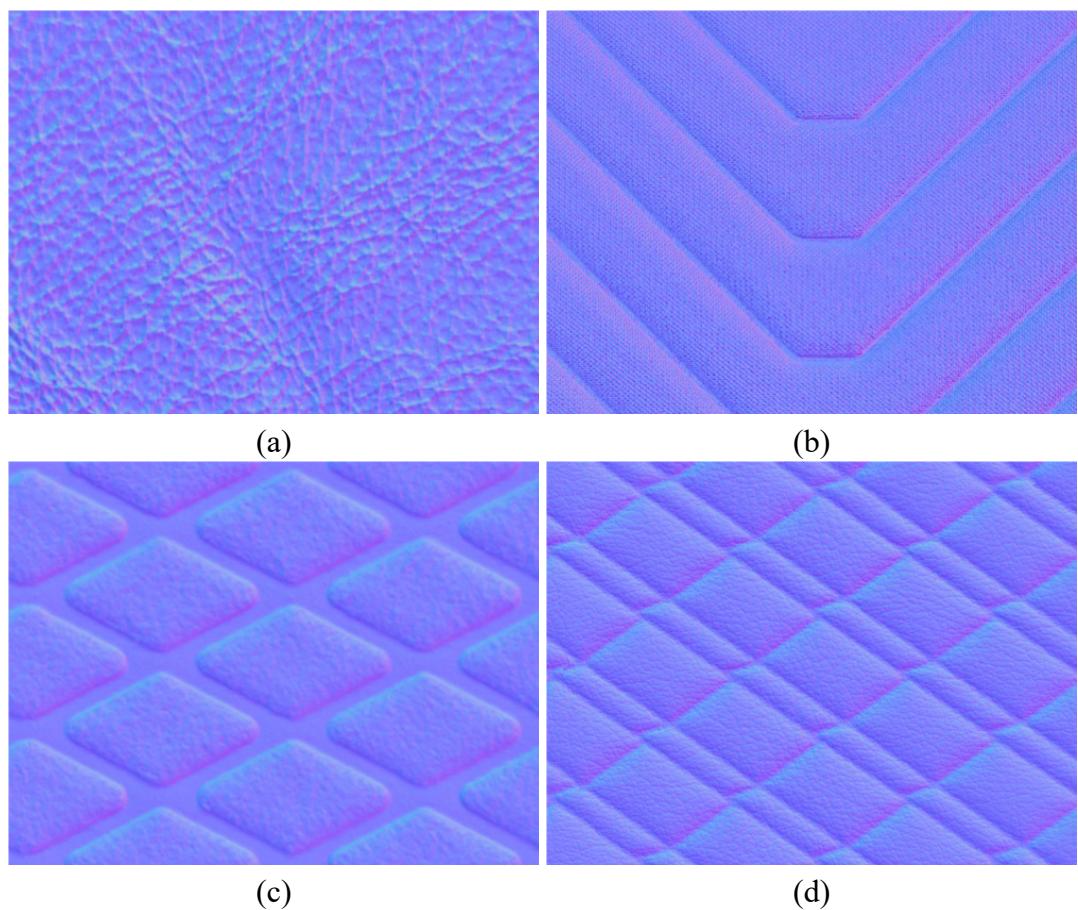


图 2-10: 法线结果图

第三章 基于传统方法的缺陷检测

视觉一直是人类和动物最重要的感官。据科学家统计人和动物获取从外界信息的过程中视觉至少占有 80% 的比例，通过视觉，人和动物能够感知外界物体的形状，体积，颜色，明暗以及运动等各种对生存极具重要意义的信息。

而计算机视觉就是一门研究如何让计算机向人和动物一样拥有视觉的科学，是研究如何使用摄像机和计算机代替人眼对目标进行识别、定位、跟踪和测量等的机器视觉，包括获取、处理、分析和理解图像或者更一般意义的真实世界的高维数据，目的是产生决策形式的数字或者符号信息。

随着计算机算法和计算机视觉领域的发展，越来越多的算法被应用到了不同的领域尤其是工业领域中来，同时，计算机视觉在工业领域中的有效应有又反过来推动者前沿科研工作者的科研工作。而在计算机视觉中，检测算法是其中非常重要的一个部分。无数的科研工作者参与其中，并且在不同的领域中取得了良好的进展。

传统的检测算法一般分为三步。首先，需要准备数据，一般需要人工标记数据，并手工设计特征和特征提取算法提取特征，结合人工标记的信息作为数据，接着训练分类模型，根据实际应用场景和数据特征选择如贝叶斯模型、决策树、支持向量机（SVM）或者逻辑回归等传统分类器，并使用第一步的标记数据训练模型，最后做检测，使用训练好的模型对新的样本进行分类和检测，根据检测结果可以系统可以有所反馈。

本章节所描述的算法即基于以上步骤，并对其做了相关优化。在数据准备阶段，我们使用上一章设计和实现的发现提取算法提取数据的法线作为输入数据图片，并使用了一些特殊的技巧巧妙地扩大了数据集，并增强了模型的旋转不变性和尺度不变性，以及克服了数据样本不平衡的问题。并且，除了手工标记数据以外，在特征的提取过程中，我们尝试和比较了不同的特征，包括不同常用特征的组合。在选择和训练模型的过程中，除了选择了适合本应用的 SVM 模型以外，我们还将其作为基学习期，使用不同的集成学习的方法训练了集成学习模型。本章将详细描述这些步骤。

3.1 数据预处理

数据预处理是机器学习中非常重要的部分。大师们常说：数据和特征决定了机器学习的上限，模型和算法仅仅是逼近这个上限而已。由此可见，数据预处理过程中对整个实验的重要性。

为了识别零件表面缺陷，我们首先需要获取零件表面信息。在第三章中计算出零件发现图过程中，本身是对原始图片信息的一种加工和提取。但是，由于精密仪器加工工艺本身精度很高，在生产过程中产生缺陷的概率很小，并且缺陷的种类多样，同类型的缺陷种类就更加稀少。这一条件限制了原始数据的规模，针对这一问题，我们使用数据预处理中数据增强的各种方法来增加样本数量，并且，在进行缺陷检测的过程中，我们并不直接将整个零件当做一个样本，而是采用滑动窗口的形式，选择一个合适的固定大小的窗口作为模型的输入。另一个挑战是数据样本的不均衡，有缺陷的样本本身数量很少，比例较低，使用了滑动窗口方法之后无疑进一步加剧了这一现象，因此我们还需要处理这种样本不均衡的情况。本节将对我们使用的数据预处理方法做详细描述。

3.1.1 零件主表面获取

一方面，观察第三章法线算法中得到的法线图，直观的看整个零件所占图像比例过低，背景占有较大面积，在之后的计算中严重影响了算法的表现和性能。另一方面，在零件生产过程中，边缘本身是否正确容易检测，并且不容易产生表面的划痕、凹陷等缺陷，缺陷以极大概率集中于零件中非边缘的地方。因此，我们将使用本章设计的算法，提取出来零件这一部分，将其称为主表面，即整个零件中最主要的表面。针对这一部分做数据增强以及稍后模型的训练。并且，在工业界中，零件本身是有一定具体规则的，并且同一款零件具有相同规则，这使得我们这一方法能够简单地推广到不同种类的零件中，使其方便的应用于实际生产中。

针对不同的零件可能拥有不同的主表面。但是绝大多数零件都是一种齿轮零件，其主表面形状为一个圆环，结合每一个零件本身含有的定位点，可以方便的定位到主表面。

首先，因为在拍照过程中，相机位置和零件位置都是不变的，因此针对同一零件，不同灯光角度条件下拍照的到的照片，其零件本身在照片中的位置是相同的，因此，我们只需要在一张图片中定位到主表面的关键信息既可

以获得所有照片以及发现图的主表面信息。针对该实验所拥有的所有输入 $Image_N1, Image_N2, \dots, Image_T1, Image_T2, Normal$ 。

首先，选择一个输入图片，使用公式 (3-1) 将其转化为灰度图。其中， $Gray$ 表示转化为灰度图之后的值， R, G, B 输入图片 RGB 色彩空间中对应通道的值。

$$Gray = R * 0.299 + G * 0.587 + B * 0.114 \quad (3-1)$$

将灰度图二值化，二值化的方法是首先使用最大类间方差来确定合适的阈值 b_t ，然后将所有大于该阈值值设置为 1，小于等于该阈值的值置为 0。最大类间方差法又叫做大津法，该方法假设阈值为 t' ，使用 t' 将整个图片分为前景（值为 1）和背景（值为 0）两个区域，其中属于前景的像素点占整幅图像的比例记为 w_0 ，平均灰度值信息记为 u_0 ，背景像素点占整幅图像的比例为 w_1 ，平均灰度值 u_1 。整幅图像的平均灰度值记为 u ，类间方差用 g 表示。整幅图像的大小为 $M \times N$ ，图像中小于阈值 t' 的像素个数记为 n_0 ，大于 t' 的像素数用 n_1 表示，于是我们可以得到公式 3-2, 3-3, 3-4。

$$w_0 = n_0 / (M \times N) \quad (3-2)$$

$$w_1 = n_1 / (M \times N) \quad (3-3)$$

$$g = w_0(u_0 - u)^2 + w_1(u_1 - u)^2 \quad (3-4)$$

又有灰度值的范围为 $[0 - 255]$ ，可以简单地使用遍历的方法寻找到使 g 最大的 t' ，即可得到需要的阈值 b_t 。

接着对生成的二值图像做腐蚀操作，并去除一些小的连通区域，然后对其做膨胀处理，最终得到了将零件和背景分开的二值区域。

并且，在这个二值区域中，背景成功被零件分割成两个部分，一部分位于零件圆环中心，另一部分将零件包裹。

因此，我们只需要提取中间这一部分的质心和提取其圆的半径即可得到内部内切圆，以及外面零件部分的圆心和半径就可以得到定位点。然后针对每个零件有的工业上的定位点，使用也可以使用同样的方式对确定的圆环部分二值化并提取一个小圆的描述子即可。中间结果如图 3-1 所示。图中左上角标记了 $Originalimage$ 为扫描得到的顶部灯光组拍照得到的结果，图中 $Normalimage$ 表示提取出来的法线结果。图中标记 $Localedimage$ 的图片表示了我们这一步中处理到的结果，该图中蓝色虚线代表了金属零件内部的圆，实现的蓝色圆代表

了零件的定位点，红色的直线连接了定位点的圆心和零件圆环的圆心。左边的图中我们尝试去掉了零件内部背景，以及外围背景，保留了零件圆环状区域，得到了如图 *Rectifiedmaps* 命名的结果。这一圆环状区域非常不利于接下来的数据预处理以及学习模型的设计。因此我们将圆环展开成一个矩形方便计算，并且在展开成矩形之后进一步去掉边缘齿轮部分，只保留了最核心的表面。其结果为图 3-1 中最下方的矩形图片，可以看到其保留了零件中最重要的表面信息，也是缺陷最容易产生和最难检测的部分。本文所描述的算法即主要未解决这一问题展开。我们将最终获得的结果称为零件主表面图。

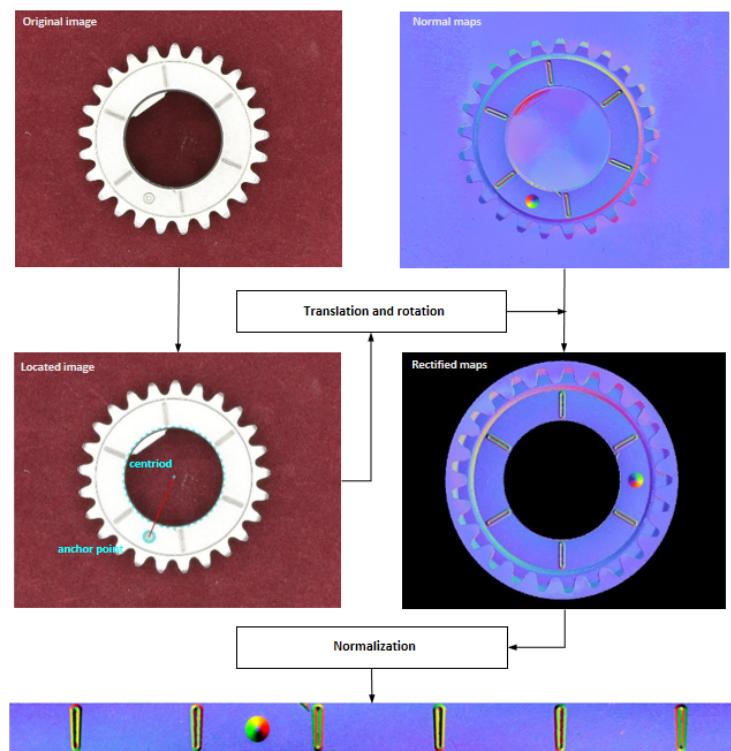


图 3-1: 主表面获取流程图

3.1.2 数据增强

首先，前文中早已提到，随着制造业、冶金业的高速发展，零件的制造工艺是不断提高的，尤其是高精度金属零件对工艺要求更严格的情况下，本身产生缺陷的概率是很小的，因此本身数据量就是一个很大的限制。其次，生产过程中产生的缺陷，不论是放大还是旋转了一定角度，它都仍然是一个缺陷，具

有一定的旋转不变性、平移不变性和尺度不变性。而仅仅使用原始数据不做任何处理，往往很难体现缺陷的这一性质。因此，我们将使用本节所述数据增强技术来处理数据，从而扩充数据库，增强模型的旋转不变性、平移不变性和尺度不变性。

3.1.2.1 分割

在实验过程中，我们并不是简单地讲一个主表面展开图作为输入，而是设定了一个固定的图片大小当做模型的输入。再生成数据的时候，用滑动窗口的形式将得到的数据划分成一个个固定大小的窗格，得到更多的数据。我们用 $WinSize$ 来表示滑动窗口的大小， $InputSize$ 来表示模型输入图片的大小。这里，我们有 $WinSize = InputSize$ 。为了确定 $WinSize$ 的大小，我们首先人工标记了所有有缺陷零件的缺陷中心点的位置 (x_i, y_i) 和缺陷的矩形包围盒的大小 (w_i, h_i) ，宽和高的最大最小值分别为 $w_{max}, w_{min}, h_{max}, h_{min}$ ，我们以 5 的宽度统计了 w_{max} 和 w_{min} 之间的宽度直方图，同样以 5 为宽度统计了 h_{max} 和 h_{min} 之间的高度直方图，通过对直方图的观察，确定 $WinSize$ 的大小。该大小能够使缺陷本身占据较大比重同时又能尽量包含到各种大小的缺陷。在实际实验过程中，我们试过 $21 * 21, 25 * 25, 32 * 32, 40 * 40$ 等不同尺寸的窗口，其中 $25 * 25$ 的实验效果最理想，也是我们最终确定的窗口。窗口的划分方式如图 3-2 所示。用滑动窗口的方式分割图片， $WinSize$ 为 $25 * 25$ ，这里产生数据是采用的步长为 17。



图 3-2: 滑动窗口分割示意图

3.1.2.2 亮度调整、镜像、旋转

在经过了缩放和分割之后，我们拥有了很多 $Size$ 为 $25 * 25$ 大小的训练样本。我们可以简单地随机调整这些样本的亮度，对这些样本做镜像，以及旋转不同的角度得到新的样本。

我们认为，一个缺陷在法线图上的的镜像以及旋转之后的结果，仍然是一个缺陷，同时，只是调整法线上的亮度值信息，只会改变法线的大小，即是的法向的模不在等于 1，而不会改变法线的方向，因此，只是改变亮度值信息也不会影响缺陷本身。而向其他的数据增强方法如色彩调整，可能会改变法线本身方向，是的原本是缺陷的地方在法线的表现上不在是缺陷，因此没有使用。

亮度调整的方法以及镜像、旋转的方法非常简单，可以直接调用 opencv^[?] 或者 matlab^[?] 的函数库，在这里由于篇幅原因不在赘述。

3.2 特征选择和提取

通过数据的增强，我们得到了非常多的可以用来做训练的样本。为了更好地分类和检测，通常我们并不是单纯使用原始图像数据作为模型的输入，而是使用巧妙设计的特征来对模型进行训练。随着计算机视觉领域的发展，无数先贤设计了不计可数的特征。本节将详细介绍本文中所使用特征以及对其特征的改进。对于接下来的基于传统方法的零件检测中，将要使用本节所描述的特征作为输入。并且，在训练和测试阶段，我们并不是单纯是使用其中某一种特征，而是联合了多种特征作为分类模型的输入样本，综合考察了不同特征以及不同特征组合对模型的影响。

3.2.1 Haar-like 特征

Haar-like 特征最早由 Papageorgiou 等人提出来应用于人脸表示，之后 Viola 和 Jones^[?] 使用 3 种类型 4 种形式的特征。该特征能反应图像的灰度特征变化。Haar-like 特征分为边缘特征、线性特征、中心特征和对角线特征，分别如图 3-3 中 A, B, C, D 所示，组合成特征模板。

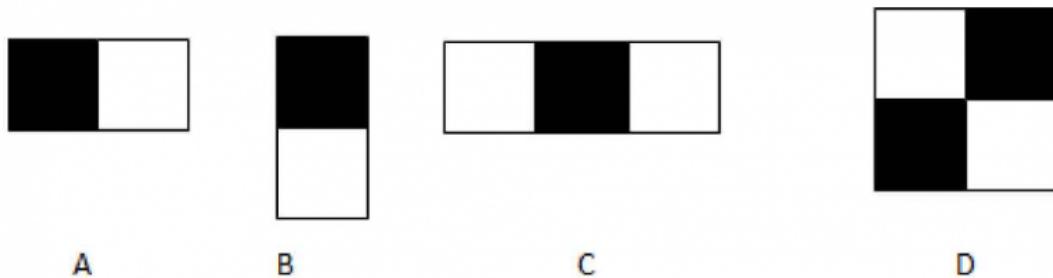


图 3-3: haar-like 特征类型示意图

模板内有白色和黑色两种类型的矩形，该摸版的特征值为白色矩形像素和减去黑色矩形像素和。提取特征的时候使用一个滑动窗口，在原始图片中不断移位滑动，子窗口每到一个位置时，就计算该区域的特征值。矩形特征可位于图像任意位置，大小也可以任意改变，因此在检测过程会产生非常多的矩形，

如果直接计算时间复杂度会非常大，但是我们可以使用一种积分图的方法来加速计算，从而降低该特征的时间复杂度，节约时间。

3.2.2 图像梯度

梯度 (Gradient) 是一个具有大小和方向的矢量。在数学中，梯度的方向是函数 $f(x, y)$ 变化最快的方向。计算机视觉领域把梯度的模简称为梯度。对于一个数组图像，可以将其看成二维离散函数，而图像的梯度就是这个二维离散函数的导数。与在数学中的意义相似，图像的梯度值较大，表示其灰度值变化较大，可能存在边缘，梯度值比较小，表示了响应灰度值变化比较小，图像比较平滑。数字图像中梯度可以用公式 (3-5) 表示，其中 dx 表示 x 方向上的梯度，用公式 (3-6) 求解， dy 表示 y 方向上的梯度，用公式 (3-7) 求解，

$$G(x, y) = dx(i, j) + dy(i, j) \quad (3-5)$$

$$dx(i, j) = I(i + 1, j) - I(i, j) \quad (3-6)$$

$$dy(i, j) = I(i, j + 1) - I(i, j) \quad (3-7)$$

在图像处理过程中，更多的是使用差分来求倒数的近似。经典的图像梯度算法是考虑图像的每个像素的某个邻域内的灰度变化，利用边缘近似的一阶或者二阶导数变化规律，对原始图像中像素某个邻域设置梯度算子。常用的梯度算子有 Sobel 算子^[?]，Robinson 算子^[?]，Laplace 算子^[?]等。本文使用 Sobel 算子提取图像梯度。Sobel 算子使用两组 3×3 的卷积核对图像做卷积操作，即可分别求出横向和纵向的梯度近似值，分别用 G_x 和 G_y 表示。假设我们有输入图像 $Image$ ，要求其梯度值，首先使用上文提到的方式将其转换成灰度图，然后使用 Sobel 算子的两个卷积核进行卷积操作，即可求得 G_x, G_y 。卷积操作如公式 (3-8),(3-9) 所示。

$$G_x = S_Conv_x * A \quad (3-8)$$

$$G_y = S_Conv_y * A \quad (3-9)$$

其中, S_{Conv_x} 表示 Sobel 沿着 x 方向的卷积核, S_{Conv_y} 表示 Sobel 沿着 y 方向的卷积核。卷积核的形式如公式 (3-10),(3-11) 所示。

$$S_{Conv_x} = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad (3-10)$$

$$S_{Conv_y} = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (3-11)$$

由此, 我们得到了图像中每个像素的横向以及纵向梯度值信息, 最终图像的梯度图 G 可以通过以下公式 (3-12) 计算。

$$G = \sqrt{G_x^2 + G_y^2} \quad (3-12)$$

为了提高计算效率, 我们不在求梯度的平方后开根号, 而是直接使用公式 (3-13) 计算求得一个近似解:

$$G = |G_x| + |G_y| \quad (3-13)$$

该方法对噪声具有平滑作用, 提取出的梯度特征不仅能够捕捉轮廓以及其他纹理信息, 还能起到弱化光照的影响。

3.2.3 梯度直方图

梯度直方图^[2](HOG) 特征最早由法国研究员 Dalal 在 CVPR-2005 上面提出, 是计算机视觉中非常常用的一种特征, 是一种在计算机视觉和图像处理中来进行物体检测的描述子, HOG 特征集合 SVM 分类器已被广泛应用于图像识别领域中, 并取得了良好的效果。该特征在图像梯度的基础上, 通过统计局部区域的梯度直方图来构成更抽象的特征。

我们可以直接根据章节 3.2.2 描述的计算梯度的方法得到图像横向梯度 G_x , 以及纵向梯度 G_y , 并且使用公式 (3-13) 计算出梯度的大小 G , 与单纯计算梯度信息不同的是, 在计算 HOG 特征的时候, 我们还需要计算出梯度的方向信息, 我们用角度来表示方向, 假设梯度的角度为 θ , 我们可以使用公式

(3-14) 计算梯度的角度。

$$\theta(x, y) = \tan^{-1} \frac{G_x(x, y)}{G_y(x, y)} \quad (3-14)$$

接着将图像分成若干个 *cells* (单元), 并且相邻的 *cell* 之间不重叠, 在每个 *cell* 内, 统计梯度方向直方图, 将所有提的方向划分为 9 个 *bin*, 作为直方图的横轴, 角度范围所对应的梯度值累加值作为直方图纵轴, 每个 *bin* 的角度范围如图 3-4 所示。由于局部光照的变化和前景背景对比度的变化, 可能梯度强度的

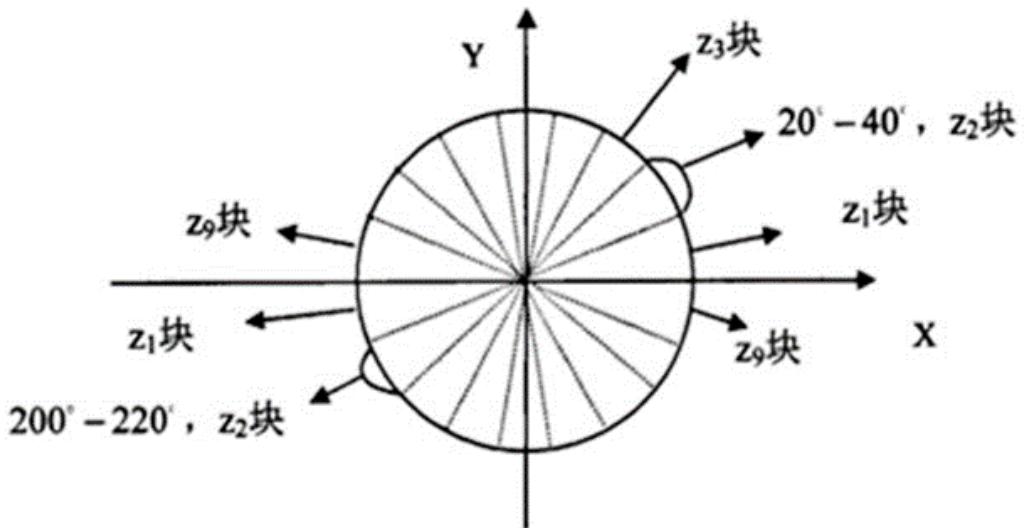


图 3-4: 梯度方向划分图

变化范围非常大。因此我们需要对梯度做归一化, 从而进一步地对光照、阴影和边缘进行压缩。归一化的做法是把各个 *cell* 组成大的、空间上连通的区间 (*blocks*), 这样, 一个 *block* 内所有的 *cell* 的特征向量串联起来便得到该 *block* 的 HOG 特征, 这些区间可以互相重叠, 可用滑动窗口的形式来统计每个 *block* 的特征, 并将它们连接成最终的特征向量。针对我们输入图片的大小, 我们使用 *blocksize* 的大小为 (10, 10), *cellsize* 的大小为 (5, 5), 这样每个 *block* 内包含了四个 *cell*, 统计 *block* 时滑动窗口的步长 *blockstride* 为 (5, 5), 这样使得 *block* 之间是有重叠的, 每一个 *cell* 的特征会以不同的结果多次出现在最后的特征向量中。

3.2.4 局部二值模式

局部二值模式 (LBP) 也是一种常见的用来描述图像局部纹理特征的算子，该算子对光照不敏感，并且计算方法简单，数据量较小，是非常实用和常用的一种特征。原始的 LBP 算子首先由 T. Ojala, M. Pietikäinen, 和 D. Harwood 提出，用来做纹理特征的提取，之后被研究人员不断改进和优化。Ojala 等人提出了圆形 LBP 算子^[?]将 3×3 邻域扩展到任意领域，使其具有了灰度和旋转不变性；Ojala 还提出了采用一种“等价模式”^[?]来对 LBP 算子进行降维，使得数据量减少的情况下能最好的代表图像的信息。

我们将 LBP 算子定义在 3×3 的窗口内，一窗口中心像素为阈值 l_thre ，比将其周围相邻的 8 个像素值的灰度值与其进行比较，若周围像素点的灰度值大于 l_thre ，那么该点的位置被标记为 1，否则被标记为 0，这样该中心像素点周围 3×3 个邻域内的 8 个点经过比较可以产生 8 个 0 或者 1，该二进制按照顺时针排列可以得到一个 LBP 值，过程如下图 3-5 所示。可以从图 3-5 中看到，该二

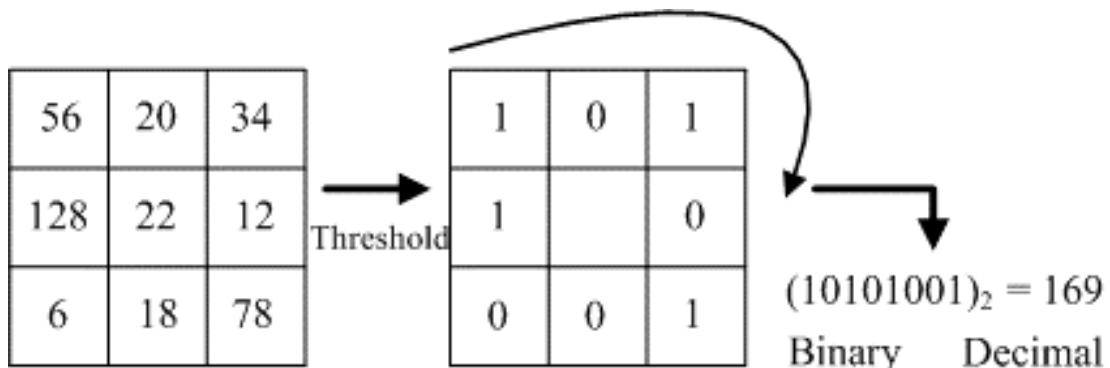


图 3-5: LBP 算子提取示意图

进制模式是以一定方向、顺序编码得到的，如果图像发生旋转，这个 LBP 值就会发生变化，所以我们主动的旋转其邻域，得到一系列的 LBP 值，取其中最小的值作为该滑动窗口中心像素点的 LBP 值，旋转过程示意图为图 3-6。

接着，我们观察得到的 LBP 值中从 $0 \rightarrow 1$ 或者从 $1 \rightarrow 0$ 的跳变次数，如果最多只有两次跳变，则认为这个 LBP 值所对应的二进制是一个等价模式，保留这一 LBP 值，否则，将其划分为混合模式类，所有混合模式类都被设置成一个 LBP 值。

并且在最后的时候，我们并不直接使用 LBP 值，而是类似于 HOG 特征一样，用一个滑动窗口来统计窗口内 LBP 值的直方图，将这个直方图作为该窗口

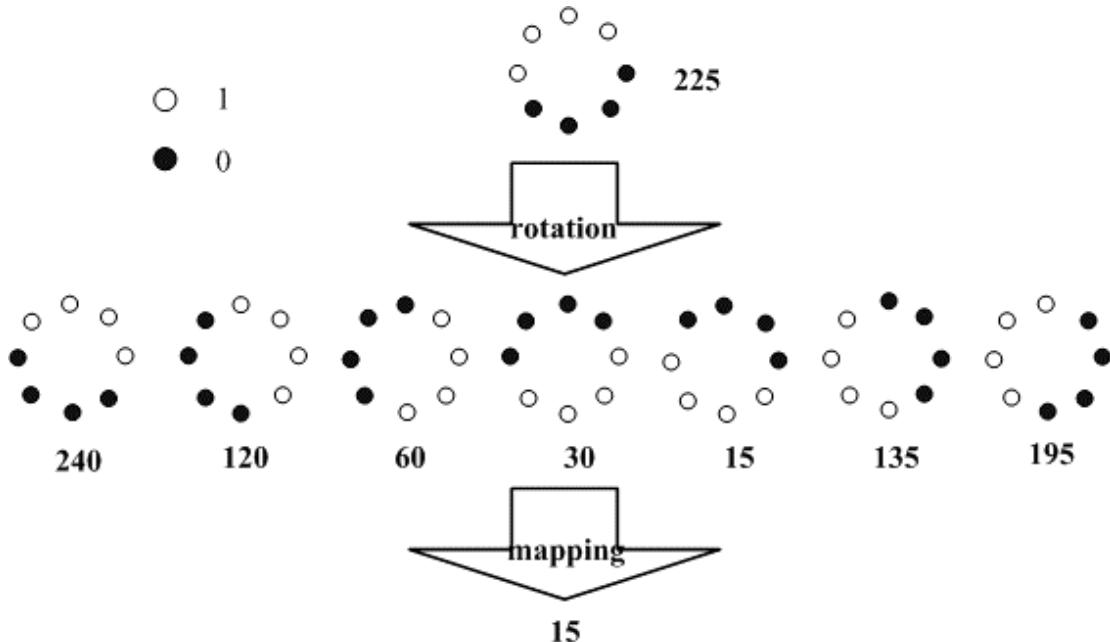


图 3-6: LBP 旋转不变性图

的特征，整幅图像的特征由这些窗口的特征连接得到。本文实验中，窗口的大小和滑动步长和 HOG 特征中 *block* 的大小和步长相同。

3.2.5 灰度共生矩阵

灰度共生矩阵是通过计算灰度图像得到它的共生矩阵，然后通过计算该共生矩阵得到矩阵的特征值来分别代表图像的某些纹理特征。该特征能够反映图像灰度关于方向、相邻间隔、变化幅度等综合信息。本小节将描述本文如何提取灰度共生矩阵的方法。

和许多局部特征一样，灰度共生矩阵特征也是建立在图像灰度图的基础上的，因此首先我们仍然需要计算过的灰度图 *Gray*，接着统计其灰度共生矩阵。取图像中任意一点 (x, y) ，其灰度值为 g_1 以及片里它的另一点 $(x + a, y + b)$ ，其灰度值记为 g_2 ，则可以得到一个灰度值对 (g_1, g_2) 。对图像中所有的点 (x', y') 以及相同 a, b 对应的位置 $(x' + a, y' + b)$ 都可以得到一个灰度对，对于整个画面，统计出每一个相同的 (g'_1, g'_2) 出现的次数，然后将它们排列成一个方阵，方阵中在 (g_1, g_2) 上的值为 (g_1, g_2) 出现的概率， $p(g_1, g_2)$ 。 (a, b) 被定义为距离差分值，在实际操作过程中常采用不同的组合，以得到不同情况下的联合概率矩阵。本文实验中我们选择 $(1, 0), (0, 1), (1, 1), (-1, -1)$ 分别对应了 0° 的扫描， 90° 的扫描， 45° 的扫描以及 135° 的扫描结果。这样，就可以将 (x, y) 像素点的空间

坐标转化为 (g_1, g_2) 的描述，形成灰度共生矩阵。

3.2.6 色彩空间

除了传统人工设计的特征以外，我们还分析了法线图本身 R, G, B 通道颜色值的特点。因为本身法线图代表了三个方向上的变化情况，是具有实际意义的。因此，我们分别对三个通道在垂直方向上做投影，统计其颜色总值信息。

针对我们的展开后的法线图 *Image*，我们假设其大小为 $M * N$ ， M 表示其宽度 *width*， N 表示其高度 *Height*。那么该图在垂直方向上的投影值可以用公式 (3-15) 表示。

$$p_i = \sum_{j=0}^{N-1} I(i, j), \quad i \in [0, M - 1] \quad (3-15)$$

其中 p_i 表示在水平方向 i 点的位置上的垂直投影的和，接着我们使用所有 p_i 的最大值和最小值做归一化，是的所有的 p_i 的值都处于 $[0, 1.0]$ 的范围内。其中第三个通道的统计结果有较明显的统计规律。其一般情况如图 3-7 所示。其中最上面 *3-high-with-point* 表示的是其中一个缺陷零件的展开图，*channel3* 是提取得 B 通道的单通道图像，*Horizontalprojection* 表示其水平投影，水平投影的中水平维度对应图像的宽度位置，垂直维度表示的是对应相同垂直点的所有图像中所有像素值的和的归一化结果，其计算方法前文已有描述。*Smoothinghorizontalprojection* 表示的是对垂直投影平滑了之后的结果，*Localmagnificationofsmoothinghorizontal* 放大展示了平滑之后的垂直投影在 $[0.0, 0.1]$ 范围的结果。随后，我们还统计了平滑之后的垂直投影图的导数信息，命名为 *curveslopeofsmoothinghorizontalprojection*。并和平滑后的垂直投影相同展示了一个放大的结果在 *Localmagnificationofcurveslope* 下展示。红色垂直框出了有缺陷部分其垂直投影的表现，黄色竖框框出了正常部分和微小缺陷的投影结果。可以很明显的看出来，在有缺陷的地方，颜色投影有一定程度的抖动，而在没有缺陷的地方，垂直投影相对平滑。因此我们也考虑直接将色彩信息尤其是 B 通道的色彩信息作为图像的特征。

3.3 分类器设计和检测

本节将主要介绍本文基于传统计算机视觉的检测方法。本文首先使用支持向量机 (SVM) 作为基学习器，使用上一节提出的图像特征训练不同的基学习

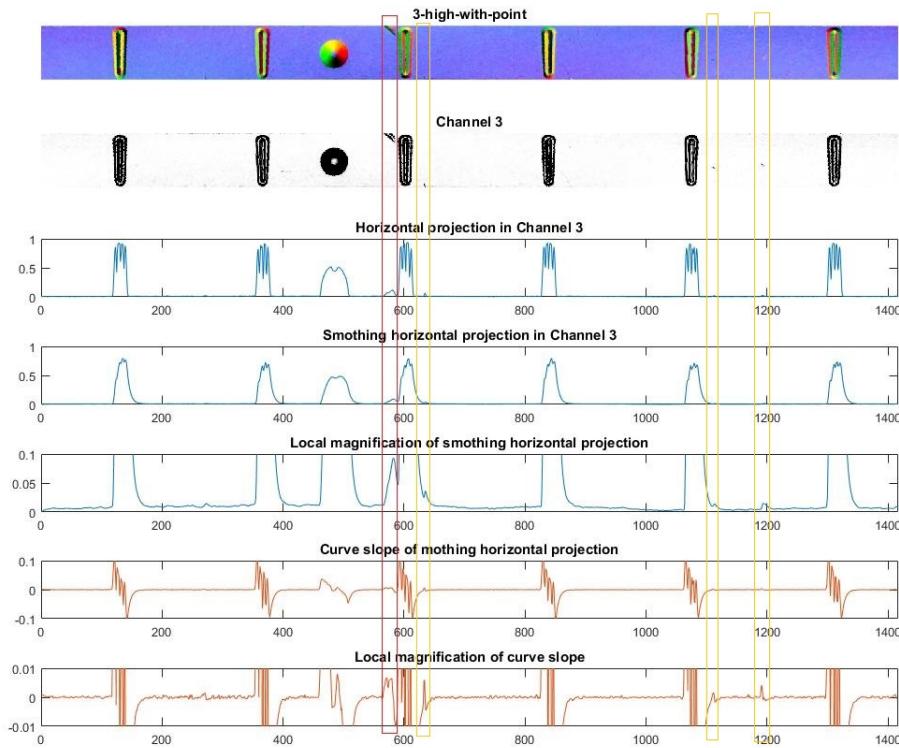


图 3-7: 色彩空间垂直投影分析结果图

器，接着使集成学习的方法，将这些 SVM 模型组合成一个强学习器。在训练的时候，考虑的样本不均衡问题，我们并不是单纯的使用全部样本训练出一个 Adaboost 组合 SVM 构成的强学习器，而是通过不同的样本划分，训练出多个强分类器，最后使用级联的方法将多个 Adaboost 训练好的强学习器用来检测。同时，在训练模型的时候，本文尝试了不同的特征以及特征组合，分别对模型进行训练以及组合，最终得到一个较好的模型。在本章中我们首先会介绍我们使用的基学习器 SVM，以及将原始样本空间映射到高维空间中时使用的核函数方法，接着描述 SVM 模型训练的过程，之后展示了如何使用 Adaboost 的方法训练和组合 SVM 模型，最后详细描述了如何使用级联的方法训练和组合多个 Adaboost 模型，并最终用于分类。

3.3.1 支持向量机

本文使用主要使用滑动窗口形式和分类模型做缺陷检测。分类一直是机器学习领域中非常的重要和基础的一个模块。并且，随着机器学习和计算机视觉的发展，人们提出了各种各样的分类模型。像随机森林^[?](Random Forest, 简称 RF)、对数几率回归^[?](logistic regression, 简称 LR)、贝叶斯分类器以及支持向量机等等。其中，支持向量机作为一种基于统计学习理论的监督学习算法，在样本数量较少的情况下也能快速学习出一个不错的分类决策。SVM 具有准确度高，可以解决高维问题，能够处理非线性特征的相互作用，无需依赖整个数据，拥有良好的泛化能力等诸多优点，从 2000 年开始，SVM 就被应用到各个领域中。基于本文缺陷检测所具有的特征维度高，线性不可分，无数据缺失等特点，我们在传统检测算法中使用 SVM 作为基分类模型。

为了检测缺陷，我们将样本分为两类，一类是有缺陷的数据，标记为 +1，一类为正常零件标记为 -1，联合 3.2 描述的特征作为 SVM 的数据样本集 $D = (x_1, y_1), (x_2, y_2), \dots, (x_m, y_m), y_i \in \{-1, +1\}$ 。支持向量机通过构建一个超平面来划分不同的类别，这个超平面就是这两个类的分类边界。以一个简单地二分类为例，在样本空间中，划分超平面可以使用公式 (3-16) 描述。

$$w^T x + b = 0 \quad (3-16)$$

公式中， $w = (w_1; w_2; \dots; w_d)$ 为该平面的法向量， x 表示样本点，决定了分割平面的方向， b 为其唯一想，决定了超平面与原点之间的距离。因此根据 w 和 b 可以唯一确定一个超平面，我们将其记为 (w, b) 。样本空间中任意一个点 x 到超平面 (w, b) 的距离 r 使用公式 (3-17) 得到。

$$r = \frac{|w^T x + b|}{\|w\|} \quad (3-17)$$

假设超平面 (w, b) 可以是样本正确分类，那么有任意 $(x_i, y_i) \in D$ ，若 $y_i = +1$ ，则有 $w^T x_i + b > 0$ ；若 $y_i = -1$ ，则有 $w^T x_i + b < 0$ 。我么可以将其表示为公式 (3-18)。

$$\begin{cases} w^T x_i + b > +1, & y_i = +1 \\ w^T x_i + b < -1, & y_i = -1 \end{cases} \quad (3-18)$$

如图 3-8 所示，距离超平面个最近的几个样本是的上式中等号成立，它们被称

为“支持向量”，两个不同类支持向量到超平面的距离之和 γ 被称为间隔，可以用公式 (3-19) 表示。

$$\gamma = \frac{2}{\|w\|} \quad (3-19)$$

SVM 的核心是找到一个具有最大间隔的超平面，即找到满足公式 (3-18) 中约

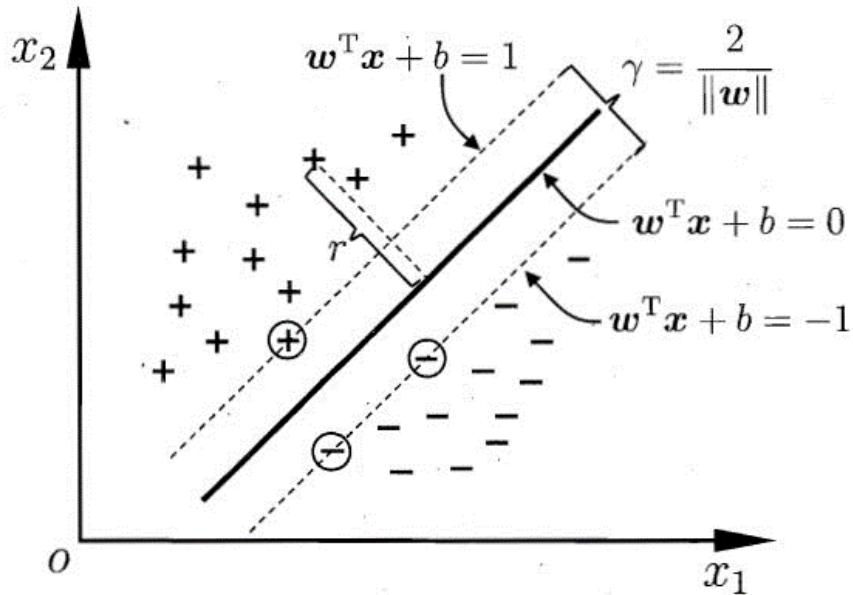


图 3-8: 支持向量机示意图

束的参数 w 和 b 使得 γ 最大，约束条件可以表示为公式 (3-20)。

$$\begin{aligned} & \max_{w,b} \frac{2}{\|w\|} \\ & \text{s.t. } y_i(w^T x_i + b) \geq 1, i = 1, 2, \dots, m. \end{aligned} \quad (3-20)$$

观察上式，为了最大化间隔，只需要最大化 $\|w\|^{-1}$ ，这等价于最小化 $\|w\|^2$ ，于是可以将公式 (3-20) 改写为公式 (3-21)。

$$\begin{aligned} & \max_{w,b} \frac{1}{2} \|w\|^2 \\ & \text{s.t. } y_i(w^T x_i + b) \geq 1, i = 1, 2, \dots, m. \end{aligned} \quad (3-21)$$

该公式本身是一个凸优化二次规划问题，可以直接用现成的优化计算包求解，也可以使用更高效的解法求解。

上述讨论过程中我们所要求解的目标一直是一个可以将不同类的样本完全

分开的超平面，现实中任务中往往很难找到这样的超平面，并且，就算在高维空间中找到了一条这样的超平面，也很难断定这个结果是不是由过拟合造成的。为了缓解该问题可以使用“软间隔”的概念，允许某些样本不满足约束，在最优化的间隔的同时，不满足约束的样本尽可能少，于是优化目标可以改写为公式(3-22)所表示的形式。

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m y_i (w^T x_i + b) - 1 \quad (3-22)$$

其中 $C > 0$ ，是一个常数，被称为正则化常数，是 SVM 中我们需要优化的参数，后面会讲到如何优化它。 $l_{0/1}$ 是“0/1 损失函数”，由于其非凸、非连续，不易直接求解，常使用其他函数替代。我们使用的 SVM 库中采用 hinge 损失，并引入松弛变量的概念对整个模型进行求解。

3.3.2 核函数

在 3.3.1一节中描述了 SVM 的基本概念和求解问题，但是其本身还是一个线性模型，建立在两个类在样本空间中是线性可分的。然而实际情况中，往往样本在样本空间中是线性不可的，两个类之间往往并不存在一个能够完美将训练样本分开的超平面。为了解决这种样本空间中线性不可分的问题，可以将数据从原始空间映射到一个维度更高的特征空间中，在高维的样本空间中寻找一个可以将两个类划分开来的超平面。

SVM 使用核函数来将原始样本空间映射到高维空间中。对于原始空间中特征向量 x ，令 $\phi(x)$ 表示 x 映射到高维空间中的特征向量。于是，我们可以得到高维空间中划分超平面所对应的模型表示为公式(3-23)的形式。

$$f(x) = W^T \phi(x) + b \quad (3-23)$$

该模型中， w 和 b 为模型参数，是需要求解的变量，类似于 3.3.1 中提到的 SVM 最优化形式，由此可以得到公式(3-24)。

$$\begin{aligned} \min_{w,b} & \frac{1}{2} \|w\|^2 \\ \text{s.t. } & y_i (w^T \phi(x_i) + b) \geq 1, \quad i = 1, 2, \dots, m. \end{aligned} \quad (3-24)$$

求解这一问题涉及到计算样本 x_i 和 x_j 映射到高维特征空间之后的内积

$\phi(x_i)^T \phi(x_j)$ 。但是，由于我们要将样本空间映射到高维空间甚至是无穷维空间中，因此直接计算 $\phi(x_i)^T \phi(x_j)$ 往往非常困难。“核函数”正式为了解决这一障碍，核函数 $k(.,.)$ ，核函数具有如公式 (3-25) 表示的性质。

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle = \phi(x_i)^T \phi(x_j) \quad (3-25)$$

即 x_i 和 x_j 在高维空间中的内积等于他们在原始样本空间中的内积通过和函数计算的结果，有了这样的函数我们就不需要在计算高维甚至无穷维特征空间中的内积。常用的核函数有线性核函数、多项式核函数、卡方核函数、高斯核函数、拉普拉斯核函数以及 *Sigmoid* 核函数。

本文实验中 SVM 算法主要是用的库是 libsvm^[?]，libsvm 中本身定义了很多基本核函数的实现，可以方便的训练我们自己定义的 SVM 模型。在实验中，我们测试了不同的核函数在数据集上的表现，其中高斯核是表现最好也是最稳定的核函数。因此，本文实验训练的基础模型所使用的核函数都是高斯核。

$$k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \quad (3-26)$$

高斯核的基本形式如公式 (3-26) 所示，其中， $\sigma > 0$ ，表示高斯核的带宽，是使用高斯核函数需要选择和调整的参数。

3.3.3 模型训练

根据在 3.3.1 节中介绍了支持向量机的分类方法。本文实验所用模型使用软间隔和高斯核函数，来达到更好的分类效果。其中，需要设定的超参数是正则化常数 C 和高斯核函数的带宽 σ ，在实际操作过程中，我们并不直接设定 C 和 σ 的大小，而是使用一个简单的工具来选择不同的参数，并使用交叉验证的方法来验证参数的效果，最终使用交叉验证结果最好的参数作为超参数训练模型。

因为 C 和 σ 是相互独立的，因此二者可以独立选择。我们是 $\log_2 C$ 的值为从 $[-10, 15]$ 中选择任意一个整数，使 $\log_2 \sigma$ 的值为从 $[-20, 5]$ 中任意选择一个整数，然后以这一对值 (C, σ) 作为超参使用交叉验证的方法训练模型，知道尝试了所有满足选择条件的 (C, σ) ，找到结果最好的 (C, σ) 作为参数。

交叉验证的方法是先将原始数据集 D 划分为 K 个大小相似的互斥子集，本文实验中 K 取 5。即 $D = D_1 \cup D_2 \cup \dots \cup D_k$ ， $D_i \cap D_j = \emptyset$ ($i \neq j$)。每个

子集通过从 D 中分层采样得到，然后每次用 $k - 1$ 个子集的并集做训练，余下的那个子集作为测试集，这样就可以获得 k 组训练集/测试集，使用这 k 个训练集/测试集可以得到 k 个测试结果，使用这 k 个测试结果的均值来评价参数 σ 的好坏。

模型的训练样本即是 3.2 节描述的特征，我们首先分别单独使用图像梯度特征、HOG 特征、LBP 特征、GLCM 特征以及原始色彩空间作为输入样本，并比对了其结果，接着选择了效果较好的 HOG, 梯度以及色彩空间连接成一个特征作为特征空间输入训练模型训练不同的基学习器。

3.3.4 自适应增强算法

自适应增强算法（Adaboost）是 boosting 族算法中最具代表性的算法，也是 boosting 中效果较好且广泛使用的算法。本文使用这一算法来训练不同的 SVM 模型并将其组合成一个强学习器。该算法工作机制为先初始训练一个基学习器，再根据基学习器的表现对样本分布进行调整，使得该学习器中分类错误的样本得到更高的权重，然后基于调整权重后的样本分布训练下一个基学习器，如此重复进行，知道学习器数目达到实现指定的值 T ，最终将这 T 个学习器进行加权组合。

Adaboost 可以用“加性模型”的形式表达，数学表示形式为公式 (3-27)。该公式中， $H(x)$ 表示最终的模型输出， $h_t(x)$ 表示每一个基学习器的输出， α_t 表示基学习器的权重。该模型可以通过最小化指数损失函数求解不同模型的权重 α_t 。并且，Adaboost 算法在获得 H_{t-1} 之后调整样本分布，使得下一个基学习器 h_t 能够纠正 H_{t-1} 的错误，Adaboost 算法通过最优化这一过程来调整样本的权重。并最终将这些模型组合成一个强学习器。

$$H(x) = \sum_{t=1}^T \alpha_t h_t(x) \quad (3-27)$$

我们使用 SVM 组成 Adaboost 的基学习器，假设第 t 个模型 h_t 的分类误差为 $\varepsilon_t = 1/N [\sum_j w_j \delta(h_t(x_j) \neq y_j)]$ ，其中， N 表示样本的数量， w_j 为每个样本的权重。则有模型权重 α_t 以及样本权重的更新公式 (3-28) 和 (3-29) 可以用表示。公式中 $D_t(i)$ 表示每个样本的权重，表示归一化常数，使得 $D_t(i)$ 能表示一个

分布。

$$\alpha_t = \frac{1}{2} \ln \frac{1 - \varepsilon_t}{\varepsilon_t} \quad (3-28)$$

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t}, & \text{if } h(x_j) = y_i \\ e^{\alpha_t}, & \text{otherwise} \end{cases} \quad (3-29)$$

3.3.5 级联检测器

在实验过程中，人工标记过的正样本集 P (有缺陷的样本的集合)是少于负样本集 N (没有缺陷的样本集合)的，尤其是在经过了数据增强之后，进一步拉大了正负样本的差距。因此，在训练模型的时候我们并不是使用全部的训练样本训练一个 Adaboost 结束，而是将负样本分成 K 个大小与 P 相同的子集 $\{N_1, N_2, \dots, N_k\}$ ，其中 K 是由 P 和 N 的大小确定的， $K = \text{ceil}(|N|/|P|)$ ，其中 $|N|$ 表示集合 N 的元素数量， $|P|$ 表示集合 P 的元素数量， $\text{ceil}()$ 表示向上取整。划分子集的方式首先从 N 中随机的无放回的抽取 $|P|$ 个元素组成第一个子集 N_1 ，不断重复这一过程直到抽取了 $k - 1$ 个 $|P|$ 大小的子集，假设接下来 N 中仍剩余的元素个数为 N_{rmain} ，接着从已经抽取出来的 $k - 1$ 个子集中随机的抽取 $|P| - N_{rmain}$ 个元素，将这 $|P| - N_{rmain}$ 和 N 中剩余的元素组成第 K 个子集。接着我们分别用着 K 个子集和正样本集 P 组成训练集，使用前文描述的 Adaboost 的方式训练多个基础 SVM 模型，这样我们就可以的到 K 个用 Adaboost 集成 SVM 训练出来的强学习模型，我们使用级联^[2](Cascade) 的方式将其组合成一个最终模型。

前文中描述的 Adaboost 训练出来的组合的 SVM 预测结果会返回一个小数 p ，这个返回的结果可以被看做是否为缺陷的概率，我们可以设置一个阈值 $threshold$ ，当 p 大于 $threshold$ 时，认为该输入为一个缺陷，当 p 小于 $threshold$ 时， p 为正常零件。显然，如果我们将 $threshold$ 调高，那么就会有更低的误识率，相应的检测率就会变低，将 $threshold$ 调低，检测率会变高，但是误识率也会随之提高，这是一个矛盾的事情。而通过级联分类器，通过增加强分类器的个数，可以达到同时提高分类器检测率和降低误识率的效果。

级联分类器流程如图 3-9 所示，使用 Adaboost 组合 SVM 训练出 K 个不同的强分类器，每个强分类器为一个 $stage$ ，每个 $stage$ 中若分类器的个数为 n 。在检测的时候使用滑动窗口的方式将输入图片分成不同的窗口，首先将这所有

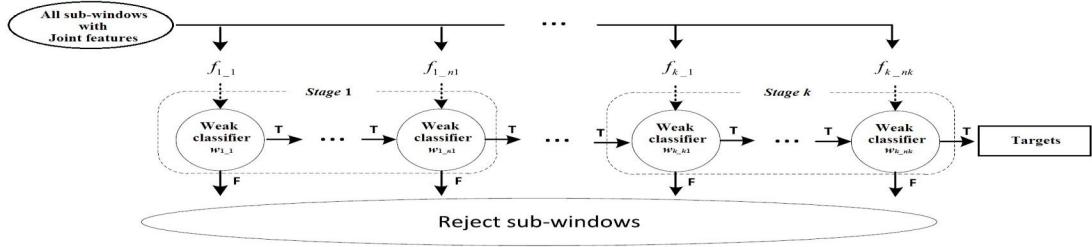


图 3-9: 级联分类器模型图

的窗口使用第一个 Adaboost 进行分类，即用 *stage 1* 分类，将所有判断为正常零件的窗口归到 *Reject sub-windows* 中，即认为其为正常零件，不在检测，将所有检测为缺陷零件的窗口送往下一个 *stage* 中，重复上一步继续划分出一部分窗口到 *Reject sub-windows* 中，直到通过所有的分类器都检测为缺陷，则认为这一窗口为缺陷，归入 *targets* 中，*targets* 中即保存了我们得到的认为是缺陷的零件。

由于我们所做的是使用精密仪器缺陷检测，一旦有缺陷的零件流出，往往会在实际应用中造成重大损失，相对于漏检，误检一个、几个甚至几十个零件所带来的损失往往微乎其微，因此，该环境要求我们尽可能的提高检测率，在检测率尽可能高的情况下降低误识率。因此，我们在设置阈值时，每一层的阈值都是能够达到最大检测率的值，然后将接受的窗口送往下一个分类器中，继续筛选，直到最终所有的分类器都认为这些窗口为缺陷，即认为这些窗口确实是缺陷。

3.4 实验结果和分析

本节首先描述实验环境，然后将描述我们所做的实验描述实验结果，实验结果主要包括两个部分，第一个部分为体征提取结果，第二个部分为使用前文描述的缺陷检测算法得到的缺陷检测结果。其中，除了我们提取出来的零件的法线图的主表面图作为输入，使用前文中描述的数据增强方法扩充了数据库并提取了前文中描写的特征，最后训练和测试了模型以外，我们还将设备获得的部分原始图片也做了和法线同样的处理，首先获取了去主表面图，接着使用数据增强的方法扩充了数据规模，并提取了主表面图相关特征，最后训练和测试了我们所提出的模型。这些原始图片作为输入的有没有加偏光镜和滤光膜东、南、西、北和顶部五组光源拍照的结果，这些实验也被用来当做法线

图输入算法的对比结果。

3.4.1 实验环境

本文算法所有实验环境都是一致的，我们在 Intel Core I7 – 7700 处理器的计算机上进行试验，系统为 Win7，处理器内存为 8GB。

同时本文算法中使用的语言主要有 matlab^[?]、C++、python 以及一些脚本语言。用到的库主要有 matlab^[?] 库、OpenCV^[?]、libsvm^[?]、TensorFlow^[?] 及其他相关库。

3.4.2 特征提取结果

我们使用 3.2 节中介绍的体征提取的方法对所有的数据提取了相应特征。在提取特征的时候，我们并没有先将其分割成小图片在提取，而是先提取整幅图片的特征，在使用与分割图片相同的方式直接从提取的特征中截取对应滑动窗口图片的特征。

以梯度特征为例，我们首先计算了所有图片的提取，然后以矩阵的形式保存对应的梯度信息，假设图片大小为 $N \times M$ ，对应像素点位置为 (x, y) ，那么计算之后的矩阵长和宽也为 $N \times M$ ，其对应位置 (x, y) 代表了对应图片 (x, y) 位置的梯度信息。因此，在使用滑动窗口将主表面图分割成训练所需图片时，只需要选则对应区域的梯度信息，并连接在一起即可以得到这一图片的梯度特征输入。

我们以一组图片为例，如图 3-10 所示为不同灯光组拍摄下的结果，以及提取法线图的结果。其中标注 *Top* 的为顶部光源拍照的出的结果，可以看出由于金属零件表面高光效果的原因，该图片中只有特别明显的缺陷和划痕才能看到，细节部分被完全掩盖了。标注了 *East*, *West*, *South*, *North* 的图分别代表了东部灯光组、西部灯光组、南部灯光组和北部灯光组拍照的到的结果，我们使用的相机分辨率较高，成像效果好，得到的四个方向光源的照片结果非常清晰。清晰的拍照结果有利有弊，好处是可以展示丰富的细节信息，即是非常微小的纹理、裂缝、划痕、污点等信息也不会丢失，同时，这也为分类和检测提供了困难，本身的色彩信息会对分类效果产生干扰，众多丰富的纹理细节更是增加了辨识缺陷的难度。标注了 *Normal* 的图片为最终提取的法线主表面图，该图中只保留了零件表面法线，能够很好地过滤无用纹理和色彩信息，同时由

裂缝、划痕等缺陷造成的表面不平整也可以由法线信息表示。

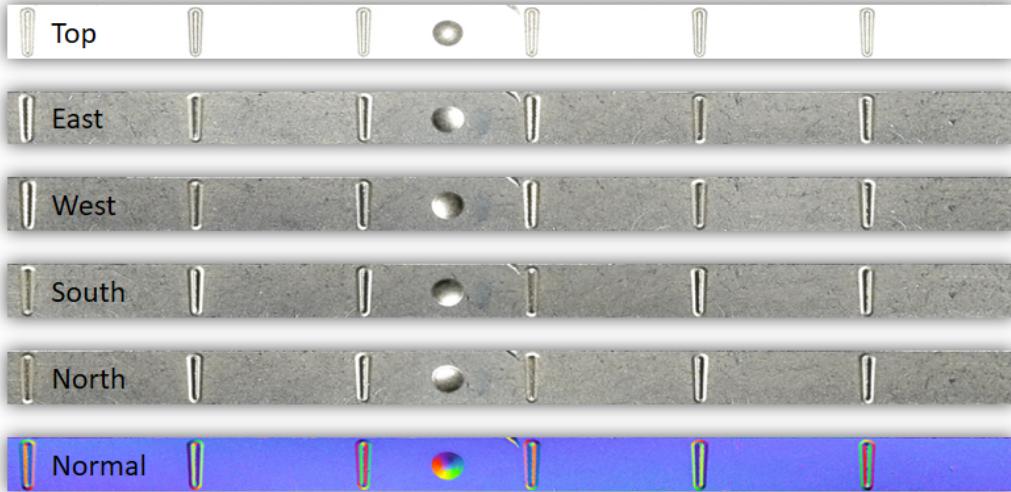


图 3-10: 不同光照下主表面结果图

为了更清楚的展示不同输入图片能够得到的细节信息，我们分别对这些输入使用 canny^[?] 算子提取了边缘信息。Canny 边缘检测是 John F.Canny 提出来的，是非常流行的一种边缘检测算法。在使用 canny 算子提取边缘信息的时候，一般只需要设定连个滞后性阈值 $threshold1$ 和 $threshold2$ ，其中值比较小的被用于边缘连接，较大的值用来控制强边缘的初始段。这里我们将两个阈值分别设为 100 和 300，canny 算子中是用的 sobel 算子卷积核大小使用默认值 3 即可。得到的结果如图 3-11 所示，该组图中命名规则和图 3-10 命名规则一样。从边缘提取之后的密度可以看出，*Top* 和 *Normal* 具有简单地边缘和实际的真正边缘比较接近，而不同方向光源下的到的结果基本相似，都有非常多的干扰信息，这些信息大多来自图片的色彩、纹理等和缺陷本身无关的信息。而 *Top* 中边缘信息太少，有些小缺陷的边缘并没有被提取出来，这在 *Normal* 中却可以清晰的看到。

上述特点还体现在提取的特征上面，如图 3-12 和图 3-13 组图片分别展示了 LBP 特征和 HOG 特征的结果，由于 *Top* 过滤掉了太多信息，无论是 LBP 特征还是 HOG 特征都会出现大量空白，得不到任何特征，而不同方向光源的特征提取结果相对过于密集，并且没有明显规律。而使用我们提取的法线结果，相对缓解了二者的问题。

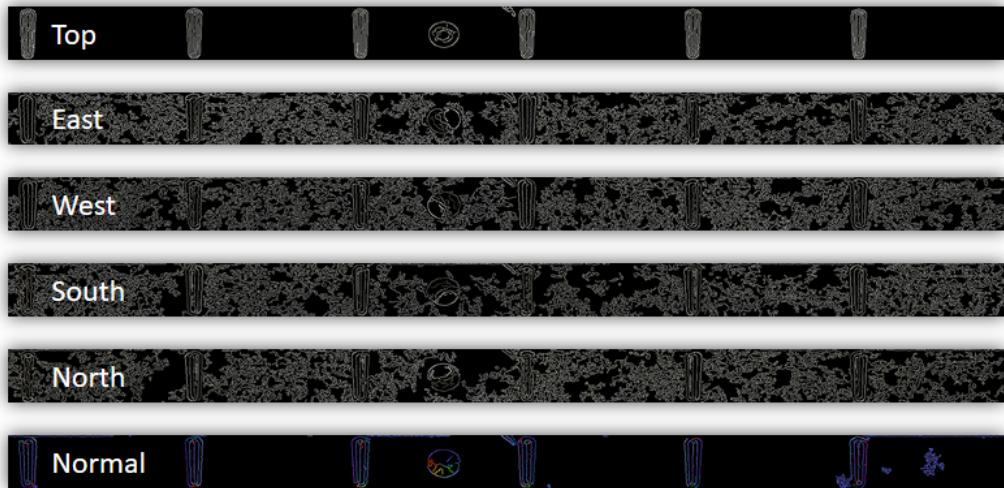


图 3-11: 不同光照下边缘检测结果图

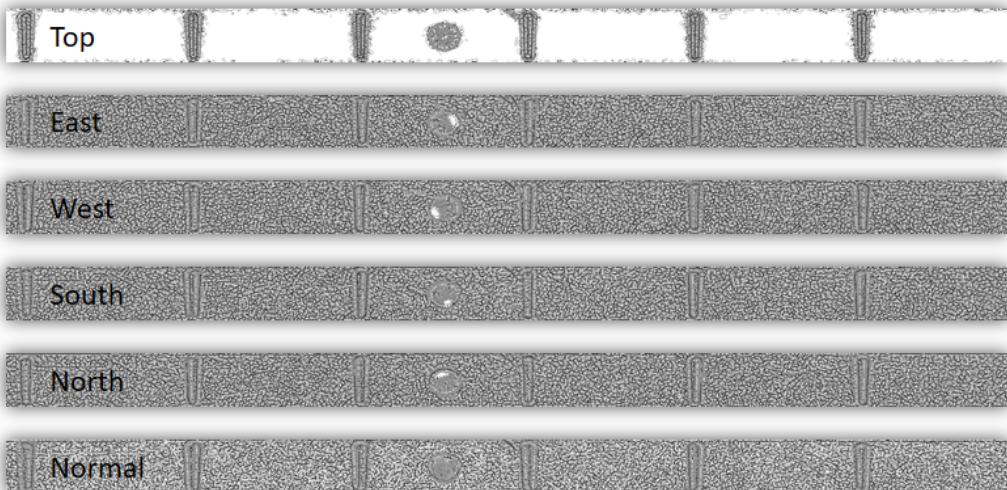


图 3-12: 不同光照下 LBP 特征提取结果图

3.4.3 传统方法缺陷检测结果

本小节将介绍我们使用的基于传统方法进行缺陷检测的结果。数据分割和增强以及提取特征算法的参数使用前文中具体描述算法的地方所确定的参数，接着我们使用不同的原始图片提取的不同特征组合训练模型，得到检测结果。

对于结果的评判标准，我们使用错误率 (*Error rate*, 记为 *Ero*)，精度 (*Accuracy*, 记为 *Acc*)，查全率 (*Recall*, 记为 *Rec*) 和查准率 (*Precision*, 记为

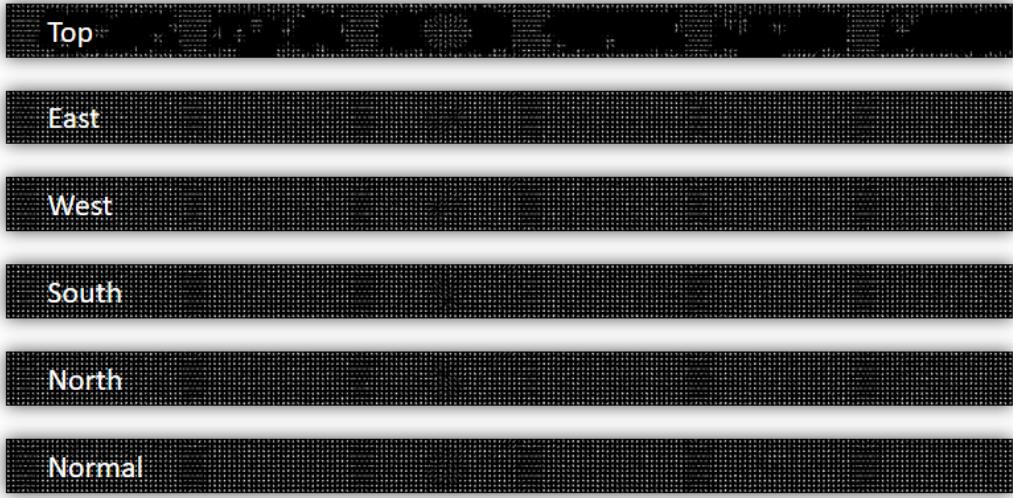


图 3-13: 不同光照下 HOG 特征提取结果图

Pre) 四个指标来衡量模型的好坏。假设总共用来测试的零件有 N 个, 其中正样本即有缺陷的样本数量为 N_p 个, 负样本即没有缺陷的样本个数为 N_n 个, 显然有 $N = N_p + N_n$ 。其中, N_p 个正样本中分类正确的个数为 TP 个, 分类错误的个数为 FN 个, N_n 个负样本中分类正确的个数为 TN 个, 分类错误的个数为 FP 个。我们可以使用公式 (3-30),(3-31),(3-32),(3-33) 表示。

$$Ero = \frac{FN + FP}{N} \quad (3-30)$$

$$Acc = \frac{TP + TN}{N} \quad (3-31)$$

$$Rec = \frac{TP}{TP + FN} \quad (3-32)$$

$$Pre = \frac{TP}{TP + FP} \quad (3-33)$$

首先, 我们简单地使用 HOG+Adaboost 的组合来训练出一组分类器。这组分类器的训练样本为不同灯光组条件下得到的数据, 经过数据预处理, 每一组数据我们得到了 564 张有缺陷的正样本, 以及 3164 张无缺陷的负样本, 基于正负样本的比例, 我们使用前面描述的方法将其训练成 6 个 Adaboost 分类器, 直接取所有分类器的平均值, 并设置阈值为 0.8 来筛选最终候选结果。其典型分类结果如图 3-14 所示, 其中黄色的矩形框给出了检测出的缺陷结果。其中 (c), (d), (e), (f) 分别代表了东西南北四组光源条件下训练得到的典型结果, 因

为南北和东西两组结果非常相似，因此我们展示的时候，在 (c) 和 (e) 中选择了一个有两个缺陷的零件得到的结果，(d) 和 (f) 选择了与 (c) 和 (e) 与其不同的一个没有缺陷的零件结果图。从图中可以看出，有缺陷的两幅图中，并不是所有的缺陷都检测了出来，并且混杂了大量的非缺陷窗口，该算法很难将缺陷和非缺陷的部分分割开来。(b) 选择了和 (c),(e) 相同的零件，代表了使用顶部光源拍照得到的结果，非常清楚的看到它得到的缺陷框非常的少，过滤掉了绝大部分噪声信息，但与此同时，它也过滤掉了一些虽然较小但是很明显的缺陷。(a) 代表了使用法线图训练和测试得到的结果，使用 HOG+Adaboost 很多时候就可以得到比较好的结果，在这副有缺陷的图中黄色框圈出了所有的缺陷信息，并且没有任何误检。

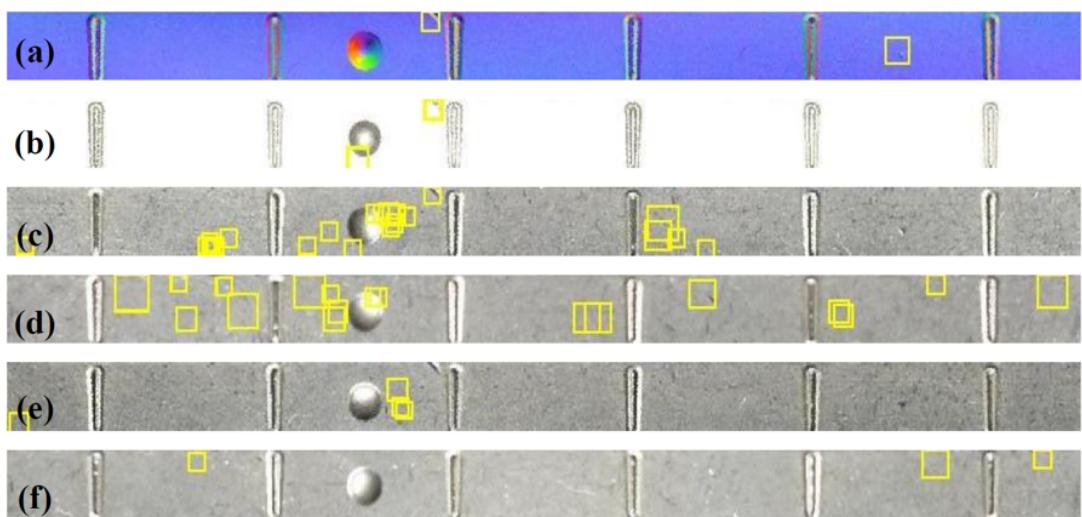


图 3-14: 基于传统方法检测常规结果示意图

检测结果如表 3-1 所示，可以明显看到东南西北四个灯光组的结果非常差，任何一项评价指标都毫无亮点，而顶部灯光组在 *Recall*, *Precision* 上略有起色，但却远不及使用法线图的结果，因此接着在训练不同特征的分类器时，我们仅仅使用法线图作为输入，并没有继续测试不同灯光组下模型的表现。

在进行特征组合的时候，我们首先分别单独使用梯度特征、HOG 特征、LBP 特征、GLCM 特征 Haar-like 特征、以及本身色空间的特征作为输入，训练不同的模型，结果如表 3-2 所示。其中使用梯度特征、HOG 特征以及本身色空间 (*RGB*) 特征得到的结果要明显好于使用 LPP 特征、GLCM 特征以及 Haar-like 特征得到的结果，我们分析认为梯度特征、HOG 特征本身包含了梯度

表 3-1: 不同光照条件下检测结果表

输入类型	Ero	Acc	Rec	Pre
东部灯光组	0.398	0.602	0.213	0.607
西部灯光组	0.374	0.626	0.229	0.625
南部灯光组	0.336	0.664	0.268	0.714
北部灯光组	0.336	0.664	0.275	0.750
顶部灯光组	0.177	0.823	0.453	0.857
法线图	0.124	0.876	0.553	0.929

表 3-2: 不同输入特征检测结果表

输入类型	Ero	Acc	Rec	Pre
Haar-like	0.199	0.801	0.418	0.821
LBP	0.156	0.844	0.490	0.893
Gredient	0.137	0.863	0.526	0.911
HOG	0.073	0.927	0.684	0.964
GLCM	0.183	0.817	0.440	0.786
HOG+Gredient+RGB	0.032	0.968	0.824	1.000

信息，反应了图像中像素值的变化大小，而在法线图中，像素值的变化正反应了零件表面的高低起伏变化情况。并且，提取出来的像素值本身即包含了法线的信息，反应了零件表面的法线情况，因此可以直接将其看做是一项非常好的特征，在分类中得到比较好的结果。有趣的事，测试的时候我们发现，有时候 HOG 特征训练出来的模型反而不如直接使用梯度信息得到的结果好，我们认为，主要是本身法线信息已经是比较好的特征了，再次提取梯度信息并继续扩展到更抽象的信息可能本身会损坏包含到信息。所以接着我们直接使用 HOG, Gredient, RGB 这三个特征作为组合输入，训练模型，取得了较好的结果，其结果也在表 3-2 中显示了。

最后，我们测试了不同分类器的检测速度，从表 3-3 可以看出，其中速度最快的是使用 haar-like 特征检测，由于其简单地结构以及积分图算法的优化速度最快，其次是梯度特征，LBP 特征都要比 haar-like 特征略慢；HOG 特征是梯度特征的基础上优化组合，速度也要比梯度特征慢，但是并不会慢多少；GLCM

表 3-3: 不同特征检测速度表

特征种类	Haar-like	LBP	Gredient	HOG	GLCM	HOG+Gredient+RGB
检测速度	11ms	17ms	16ms	22ms	486ms	25ms

特征提取最复杂，速度最慢；而我们使用的梯度特征加上 HOG 特征最后组合上颜色空间信息，由于 HOG 特征在计算的时候本身就需要求梯度信息，而色彩空间本身就存在，因此并不比 HOG 特征慢多少，但是却能够显著改善分类器的效果。

第四章 基于深度学习的缺陷检测

除了手工特征加传统分类算法检测以外，我们还是尝试了是用最新的深度学习方法来对金属零件缺陷进行检测。自 2006 年由 Hinton 等人提出了深度学习^[?]的概念，深度学习已经被广泛应用于图像处理、计算机视觉等诸多领域，并取得了令人惊讶的成就。并且，随着硬件设备的发展，尤其是 GPU 的出现和发展，计算性能得到了快速提升，是的深度学习不仅仅是活跃在学术界，更是在工业界中显现出非凡的潜力。本章节主要介绍了本文如何针对这一问题选择和使用深度学习，以及如何对模型进行优化。首先会介绍深度学习的一些基本概念，接着会介绍著名的反向传播算法，正是该算法使得基于神经网络的深度模型能够得实现和推广，然后我们将介绍本文实验所使用的网络结构以及数据预处理方法和模型训练实施的过程。

4.1 卷积神经网络

卷积神经网络 (*Convolutional Neural Networks*, 简称 *CNN*) 模型是深度学习的一个非常成功的算法，并且能够组合成多层网络结构，通过一定方法训练。总的来说，卷积神经网络是一种层次模型，在计算机视觉领域，其输入一般是原始图像，通过卷积 (*convolution*) 操作，池化^[?](*pooling*) 操作和非线性激活函数 (*non-linear activation function*) 映射等一系列操作堆叠在一起，逐渐从原始图像中将高维语义信息提取出来，并最后通过全连接层和输出层连接，将目标任务（分类、回归等）形式化为目标函数，并使用损失函数 (*loss function*) 计算预测值和真实值之间的损失 (*loss*)，使用反向传播算法^[?](*back-propagation algorithm*) 将损失从后向前传播，更新每层参数，并在更新后再次训练向前传播，如此反复直到模型收敛。

一个好的卷积神经网络模型得益于良好的网络结构设计，而一个复杂而精妙的网络结构往往由诸多基本结构组成，本节将描述构成神经网络的一些重要基本结构以及一些基本的概念，这些结构概念将在接下来的网络结构以及实验等部分中用到。

4.1.1 卷积层

卷积层是卷积神经网络中的基础操作，包括最后网络最后起分类作用的全连接层在工程实现时也是由卷积操作替代的。所谓卷积，就是使用一个固定大小的卷积核，通过滑动窗口的形式和图像对应区域做内积（逐个元素相乘再求和）的操作。此时的卷积核类似于图像处理中常见的滤波操作，其中卷积核为一个固定大小的矩阵，这些矩阵的值即为这一卷积核的权值。一般我们会使用正方形的卷积核，并且取奇数大小，常用的有 $1 \times 1, 3 \times 3, 5 \times 5$ ，卷积核的大小也对应了滑动窗口的大小。没做完一次卷积操作，卷积核即移动一个位置，移动的大小记为卷积步长(*stride*)，因为图像是二维的，为了卷积整幅图像，需要卷积核向两个方向移动，因此卷积步长也是一个二维结构，一般也是取 1×1 。

不难看出，卷积是一种局部操作，通过一定大小的卷积核作用域图像的局部区域可以得到图像的局部信息，通过层层卷积，高层的卷积核可以扫过的信息会覆盖原始图片的更大区域，提取图像更高层的信息。与此同时，往往图片的底层特征与其在图片中的位置是无关的，因此我们可以在使用同一个卷积核扫描整幅图像时使用同一个权值，即这张图片每个位置都会被同样的卷积核扫过，实现权值共享。这大大减少了卷积层的参数，预防了过拟合。卷积核的权值通过发向传播算法最优化得到。

4.1.2 激活函数

激活函数层又称为非线性映射层，其引入正是为了增加整个网络的表达能力(非线性)。如果没有激活函数层，那么整个网络仍可以看作是若干线性操作的堆叠，只能起到线性映射的作用。直观的看，激活函数模拟了神经元的特点：接收一组输入信号并产生输出，激活函数也是这样，接受一个输入，这个输入往往是卷积层卷积之后得到的值，激活函数会根据这一输入产生一个输出信号，因此通常激活函数紧跟着卷积层使用，常常被合并到一层。

从定义来看，集会所有的连续可导函数都可以作为激活函数，但是目前常见的多是分段性和具有指数形状的非线性函数，本小节将介绍其中最基础也是最常见的有 *Sigmoid*^[?], *tanh*^[?], *ReLU*^[?] 激活函数，在实际使用过程中，常见的激活函数多达十几种，而这些激活函数往往以上述三种激活函数为基础，是对其做出一定改进优化的结果。

4.1.2.1 Sigmoid 函数

Sigmoid 是使用范围最广的一类激活函数，它是具有指数函数形状的激活函数。该函数的定义如公式 (4-1) 所示。

$$f(x) = \frac{1}{1 + e^{-x}} \quad (4-1)$$

其函数形状如图 4-1 所示，从图中可以明显的看出来，经过 *Sigmoid* 型函数作用后，输出响应的值域被压缩到 $[0, 1]$ 之间。但是，对于大于 5(或小于 -5) 的值，无论多大(或多小)都会被压缩到(或 0)，此部分的梯度会接近 0，会导致在误差反向传播过程中导致该区域的误差很难甚至根本无法传播，进而导致模型无法训练，这一现象被称为梯度的“饱和效应”。梯度消失问题虽然存在，但是可以被新的优化方法环节，如 Batch Normalization^[?]。*Sigmoid* 有其有利的一面，它在物理意义上最为接近生物神经元，并且其 $(0, 1)$ 范围的输出可以表示一个概率分布，或者用于输入的归一化。

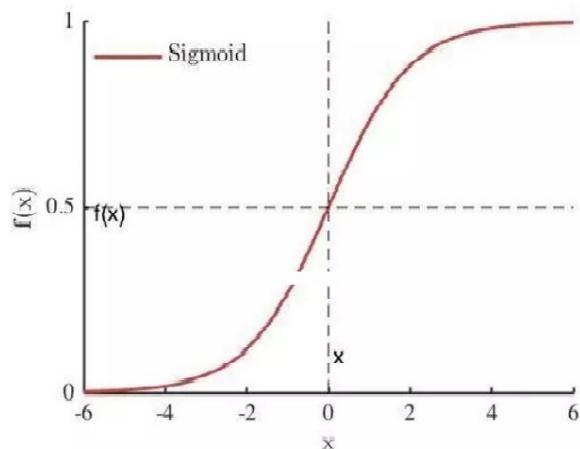


图 4-1: *Sigmoid* 激活函数示意图

4.1.2.2 Tanh 函数

Tanh 激活函数又被称为双正切函数，其形状与 *Sigmoid* 函数类似，都是将输出压缩到了 $(0, 1)$ 的范围，其形式如公式 (4-2) 所示。

$$\tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (4-2)$$

其函数形状如图 4-2 所示，不难看出， $tanh$ 函数同 $sigmoid$ 形状极其相似，同样在 x 比较大或者比较小的时候会出现梯度消失的现象，但是该函数输出的均值比 $sigmoid$ 函数更接近 0，在训练的时候收敛速度更快。

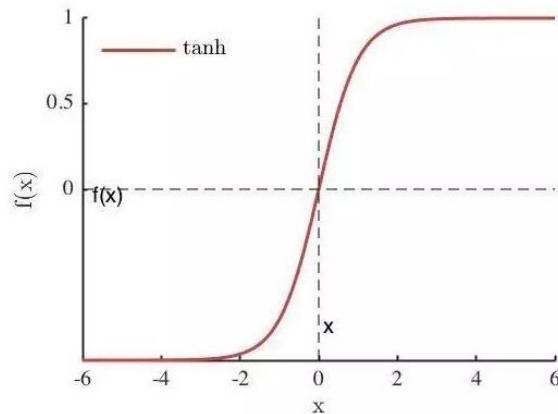


图 4-2: $tanh$ 激活函数示意图

4.1.2.3 修正线性单元 ($ReLU$)

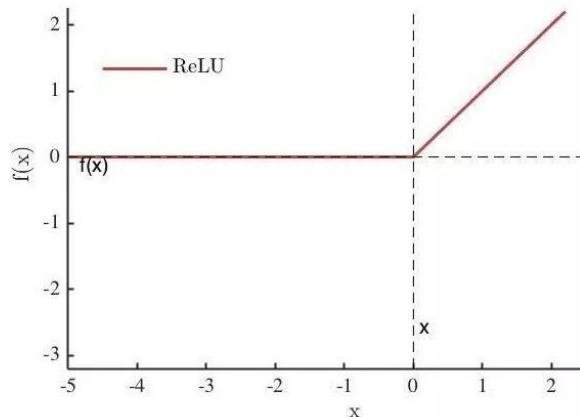
$ReLU$ ^[?] 是由 Nair 和 Hinton 在 2010 年引入神经网络的，现已成为深度卷积神经网络中最为常用的激活函数之一，也由此产生了很多变种被广泛应用于各种模型中。 $ReLU$ 函数实际上是一个分段函数，其定义如公式 (4-3) 所示。

$$ReLU(x) = \max\{0, x\} \quad (4-3)$$

与前文所述两个激活函数相比， $ReLU$ 函数的梯度在 $x \geq 0$ 时为 1，反之为 0(如图 4-3 所示)；在 $x \geq 0$ 部分该函数完全消除了梯度消失的情况，并且，在计算复杂度上更加简单，在实际应用中能够更快的收敛。 $ReLU$ 的主要缺陷是在 $x < 0$ 的时候，梯度为 0，对这部分的卷积结果无响应，它们一旦变为负值将再无法影响网络的训练。当然，这一缺陷可以通过 *Parametric Rectified Linear Unit (PReLU)*^[?] 来改善。

4.1.3 池化层

池化层一般采用平均池化(*average-pooling*)或者最大池化(*max-pooling*)。和卷积操作类似，它使用一个滑动窗口，取滑动窗口中像素值最大值(最大池

图 4-3: *ReLU* 激活函数示意图

化层)或者平均值(平均池化层),以此对卷积操作之后得到的特征进行降维。从操作的角度来讲,池化层可以看做是一个用 p -范数(p -norm)作为非线性映射的“卷积”操作,当 p 趋近于正无穷时就是最常见的最大池化层。与卷积层不同的是,池化层不需要设定参数,使用时只需要指定池化层类型、池化操作核的大小和池化操作的步长即可。从图像处理的角度来看,池化层可以被视为一种“降采样”,一般在卷积操作之后得到的特征可能会包含较多冗余信息,池化层的引入可以对输入的特征进行降维和抽象,同时在一定程度上预防过拟合,更加方便优化。虽然池化层不是卷积神经网络中必须的操作,但是由于其良好的性质,往往会在每一个卷积层之后都使用池化层对特征降维。

4.1.4 全连接层

全连接层往往在整个卷积神经网络的最后一层或几层,起到“分类器”的作用。在这一层,每一个神经节点都会与上一层中的所有节点相连,因此全连接层往往含有较多参数。如果卷积层、激活函数和池化层起到将原始数据映射到隐层特征空间中的话,那么全连接层就起到将特征空间中的特征表示到样本的标记空间中的作用。实际应用中全连接层可以通过卷积操作实现,对于前一层为全连接层,可以通过 1×1 的卷积核实现,对于前面一层为卷积层,则可以通过与前一层特征维度大小一致的卷积核来做全局卷积。

4.1.5 损失函数

损失函数又被称为目标函数，使用来衡量该预测值和真是样本标记之间的误差。使用网络结构从输入层经过各种卷积、池化以及全连接操作之后，会对输入做出预测，预测值和真实标记之间的误差的大小即可用损失函数表示，这一误差又被用来发向传播，更新权重。在目前的卷积神经网络中，交叉熵损失函数和 l_2 损失函数分别是类别问题和回归问题中最常用的目标函数。当然，针对不同的问题特性，也有越来越多的损失函数被提出来可供选择。

$$L_{loss} = -\frac{1}{N} \sum_{i=1}^N (y_i \log \hat{y}_i + (1 - y_i) \log (1 - \hat{y}_i)) \quad (4-4)$$

本文将缺陷检测定义为一个二分类问题，使用 *sigmoid* 函数做分类，因此我们使用交叉熵损失函数 L_{loss} 。其公式定义如 (4-4) 所示。其中 N 表示样本的数量， L_{loss} 表示所有的损失。 y_i 表示样本本来的标签，当样本为缺陷样本时， y_i 值为 1，当样本为正常样本时， y_i 值为 0。模型的输出记为 \hat{y}_i ，表示该输入为一个缺陷的概率，检测的时候我们设置一个阈值 $threshold$ ，当 $\hat{y}_i \geq threshold$ ，认为这是一个替换函数缺陷，当 $\hat{y}_i < threshold$ ，认为是正常零件。

4.1.6 优化方法

有了目标函数之后就可以对网络进行优化，深度卷积神经网络通常采用随机梯度下降类型的优化算法对模型进行训练调整参数。随着深度学习的发展，人们也研究出越来越多的有效的网络训练优化算法，并且，在现有的深度学习框架中也提供了这些优化策略的实现，可以很方便的使用。本节将介绍其中几种最常用的优化算法。

4.1.6.1 随机梯度下降

随机梯度下降算法 (*Stochastic Gradient Descent*, 简称 SGD) 是神经网络中最经典也是最基础的基本算法。该算法每次批处理训练时计算网络的误差并对其进行反向传播，然后根据误差的一阶梯度信息对参数调整，参数的更新策略可以用公式 (4-5) 表示。

$$w = w - \eta \cdot dw \quad (4-5)$$

其中， dw 表示误差对参数 w 的导数即梯度，其值完全依赖于当前数据在

目标函数上的误差， η 表示学习率，即根据梯度学习参数的快慢，是 SGD 算法中唯一的超参数，表示了当前批的梯度对网络整体的参数更新的影响程度。SGD 算法收敛效果稳定，但是收敛速度较慢，并且，在训练的时候极容易困在鞍点，即 $dw = 0$ 的点，走不出去，导致模型收敛在局部最优解中，无法完全优化。并且，在选择学习率的时候，过大的学习率可能导致模型在最后不收敛而是来回震荡，过小的学习率又会影响收敛速度，因此本身学习率如何设定也是一个复杂的问题，并且会影响模型的效果。虽然可以多次随机初始化参数训练多次，对比选择效果最好的网络参数来改善 SGD 算法容易困在鞍点的问题，但这同时增加了大量的训练时间，同时没办法改变该算法本身收敛速度慢的缺点。

4.1.6.2 基于动量的随机梯度下降法

受物理学研究的启发，提出了基于动量^[?](momentum) 的随机梯度下降用于改善 SGD 更新时可能产生的震荡现象，以及容易困在鞍点等问题。该算法通过前几轮训练积累的“动量”信息辅助参数更新，更新策略可以表示为公式(4-6)的形式。

$$\begin{aligned} v &= \mu \cdot v - \eta \cdot dw \\ w &= w + v \end{aligned} \tag{4-6}$$

公式中， μ 为动量因子，表示动量对整体梯度更新的影响程度， v 表示动量，在梯度方向相同的方向逐渐增大，在梯度方向改变的方向逐渐变小， η 为学习率，表示梯度 dw 对动量的影响。基于动量的随机梯度下降法除了可以抑制震荡，还可以在网络训练的中后期趋于收敛，并且在参数的鞍点附近来回震荡时帮助其跳出拒不限制，找到更优的网络参数。一般 μ 取 0.9 就可以。

4.1.6.3 RMSProp 算法

RMSProp(*root mean square prop*) 算法^[?]可以针对学习率做动态的调整。这一算法的更新策略如(4-7)所示。

$$\begin{aligned} Sdw &= \beta \cdot Sdw - (1 - \beta)dw^2 \\ w &= w - \alpha \frac{dw}{\sqrt{Sdw} + \epsilon} \end{aligned} \tag{4-7}$$

上式中加入了 w 的二阶导数更新 Sdw , 在更新 w 的时候使用用 w 的梯度除以 Sdw 的平方根作为学习对象, 这使得对于不同的 w 可以有不同的学习率。其中 ε 只是为了防止分母变为 0, 本身对算法的意义不大, 一般置为 10^{-6} , β 为衰减因子, 消除算法对全局学习率 α 的依赖, 较大的 β 会促进网络更新, 较小的 β 会抑制网络更新, 一般可以取 0.9, α 可以取 1。

4.1.6.4 Adam 算法

Adam^[?] 算法用梯度的一阶矩估计和二阶矩估计动态调整每个参数的学习率, 并且经过偏置校正后, 每一次迭代学习率都有一个确定的范围, 可以使得参数更新更加平稳。算法实现步骤如

1. 初始化 $Vdw = 0, Sdw = 0$;
2. 每一次迭代, 使用当前批样本计算 dw , 即参数的导数。并使用公式 (4-8) 计算和调整参数 w 。

$$\begin{aligned} Vdw &= \beta_1 \cdot Vdw - (1 - \beta_1)dw \\ Sdw &= \beta_2 \cdot Sdw - (1 - \beta_2)dw^2 \\ Vdw^{corrected} &= \frac{Vdw}{1 - \beta_1^t} \\ Sdw^{corrected} &= \frac{Sdw}{1 - \beta_2^t} \\ w &= w - \alpha \frac{Vdw^{corrected}}{\sqrt{Sdw^{corrected}} + \varepsilon} \end{aligned} \tag{4-8}$$

可以看出, Adam 算法也需要制定参数, 其中 ε 同样是为了防止分母为 0, 使用 10^{-6} 即可。 β_1 为第一矩参数, 可以使用 0.9, β_2 为第二矩参数, 可以使用 0.999。该算法既考虑了动量又针对不同的参数调整为不同的学习率, 常常具有收敛速度快, 效果好的优点, 往往是模型训练的优先选择。本文实验即使用 Adam 算法对模型进行优化。

4.2 网络结构

基于效率和数据量的考量, 本文基于 VGG-16^[?] 来对零件做分类。VGG 网络是 2014 年由 Karen Simonyan 和 Andrew Zisserman 等人提出, 在该类模型中所有的卷积层都使用 3×3 的滤波器, 并且步长都为 2, 所有的池化层都使用 2×2

的池化窗口，步长也为 2。并且，VGG 将卷积层的概念提升到了卷积组 (group) 的概念，每个卷积组中含有几层卷积层，模型中包含五个卷积组，前两个卷积组中各自包含两个卷积层以及一个最大池化层，后三个卷积组中每个包含三个卷积层，并同样在卷积层后面加了一个最大池化层，最后在这五个卷积组后面加了三层全连接层用来分类。其结构图如图 4-4 所示。

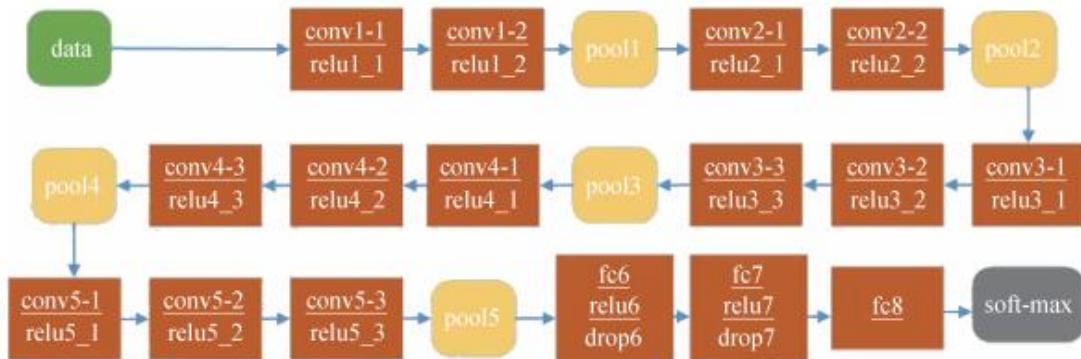


图 4-4: VGG-16 模型结构示意示意图

该网络中每一个卷积层都随之使用 *ReLU* 激活函数，同时在训练阶段，会在 *fc6* 和 *fc7* 中使用 *dropout* 的方法来预防过拟合。*Dropout* 方法是在训练的时候，向后传输的过程中对神经元以一定的概率随机失活，从网络中丢弃。这里的随机失活只是暂时的，每次 *mini-batch* 都会以固定的概率重新失活神经元。*Dropout* 一般取 0.5 可以达到较好的结果。事实上，2015 年 Sergey Ioffe 和 Christian Szegedy 提出了 *Batch Normalization* 的方法，可以大大加快神经网络的训练速度，并且可以抛弃掉 *dropout* 操作。*Batch Normalization* 是在网络的每一层输入的时候，又插入了一个归一化层，然后再将数据送入网络的下一层，该归一化层是一个可学习的、有参数的网络层，可以将输入数据归一化至均值为 0，方差为 1 的分布，从而减少神经网络中层与层之间的连接。虽然理论上 *Batch Normalization* 可以被应用于神经网络的任何神经元上，但是实际操作过程中我们一般将其应用在卷积层之后，激活函数之前。

本文代码实现基于 TensorFlow 框架，该框架下有实现好的各种基础经典框架，我们只需要根据我们的需要对现有框架进行微调即可。TensorFlow 中的 VGG 网络保存了使用 ImageNet-1000 训练好的网络参数，并且整个网络结构是适用于这一数据集的，分类对象共有 1000 类，而我们在做缺陷检测时，只是一个二分类问题，我们首先去除了基本 VGG 结构中的 *dropout*，并加入了 *Batch Normalization*，接着改了最后用来分类的 *softmax* 函数，变为 *sigmoid* 函数。

数，使其能够简单地变为一个二分类问题。

4.3 数据预处理和数据增强

同基于传统方法的缺陷检测一样，在应用深度学习对零件缺陷进行检测时，数据的预处理也是非常重要的一环。在进行数据预处理的时候，同基于传统的方法的缺陷检测一样，首先，我们也提取了主表面并将其展开成矩形的形状。方法和第3章所使用的方法一样。

接着，在数据增强阶段，为了获得更多的数据，我们也使用相同的方法对数据做分割，产生更多的固定大小的数据，接着对这些数据做增强，使用亮度调整、镜像、旋转等操作在此基础上得到更多的数据，并获得旋转不变性，平移不变性等较好的数据属性。

但是，VGG 网络模型输入是 224×224 ，同 3.1.2.1 节的中所描述的图像分割方法不同的是，我们使用滑动窗口划分出的数据大小是 112×112 ，并使用双三次插值算法将其放大到 224×224 。在实际训练网络的时候，很多分类和检测的过程都会为了尺度不变性做一定操作，比如在训练阶段并不是直接使用划分后的窗口图片，而是对其随机放缩，在一定的范围随机选择一个数当做放缩系数，对图片进行放大或者缩小，然后采用一定的方法比如图像补全、随机裁剪获取放缩后的符合输入要求的区域，然而请注意本实验并没有采取这样的方法，原因有二，首先，生产过程中零件的缺陷是有明确定义的，有些公司将宽度或者深度大于 5mm 的划痕或者裂纹等认定为缺陷，有些企业要求更高或者更低，这就使得很多缺陷对其缩小之后即不在称为一个缺陷，很多非缺陷的部分对其放大之后会变成缺陷，因此，使用随机缩放而没有标注的情况下反而会影响分类器的效果。其次，在做基于传统方法的缺陷检测时我们也同样没有对尺度不变性做处理，为了对比传统学习方法和深度学习方法本身，在训练和测试深度学习模型的时候，我们也并不针对尺度不变性做处理。

4.4 模型训练

有了网络结构之后就可以对模型进行训练和测试了。在训练阶段，首先我们要对所有的参数进行初始化，其中网络权重使用 VGG 在 ImageNet 中预训练好的权重作为初始化。优化的目标函数为前面介绍神经网络时定义的针对这

表 4-1: 深度学习和传统方法结果对比

模型种类	Ero	Acc	Rec	Pre
HOG+Gredient+RGB	0.032	0.968	0.824	1.000
VGG-16	0.112	0.888	0.580	0.938

一问题的损失函数。深度学习训练过程并不需要交叉验证，我们将所有的测试集都用来做训练。在训练的时候使用 *mini – batch* 梯度下降的方法，使用 Adam 对模型进行优化。其中 *mini – batch* 的大小设置为 128。并且，在训练的时候，我们将除了最后一层全连接层以外所有的层数参数全部冻结，只训练最后一层知道收敛，接着训练倒数第二层，这样逐层训练网络，得到收敛结果，可以有效防止因为数据集数量不够导致的直接全部训练模型无法收敛的问题。

同时，由于本身样本不平衡的问题，我们对神经网络也同样适用 Cascade 的方法训练出不同的深度学习模型并强强联合组合成一个更强的分类器。

4.5 实验结果

本小节将分析使用基于深度学习的缺陷检测算法得到的结果，并且与传统算法进行了对比。实验使用的深度学习框架是 TensorFlow。该框架本身是一个采用数据流图，用于数值计算的开源软件库。并且，TensorFlow 本身支持 windows，它具有灵活的架构可以在多种平台上展开计算，现已被广泛应用于学术界和工业界中。

同基于传统特征的缺陷检测算法一样，我们同样使用错误率 (*Errorrate*, 记为 *Ero*)，精度 (*Accuracy*, 记为 *Acc*)，查全率 (*Recall*, 记为 *Rec*) 和查准率 (*Precision*, 记为 *Pre*) 四个指标来衡量模型的好坏。其结果如表 4-1 所示，我们意外的法线，这一检测结果竟然比不上使用 HOG, Gredient, RGB 特征训练的传统模型的到的结果，我们分析认为有两点原因，首先是在模型训练的时候，为了更好匹配 VGG16 框架，我们将输入调整到了 224×224 ，虽然我们使用的滑动窗口大小为 112×112 ，但由于本身缺陷就很小，这一窗口对于缺陷本身还是太大了，是的缺陷占据窗口的比例太小，其次是模型网络较深，不断地对特征进行抽象，缺陷本身占比较小，更是使得缺陷所对应的特征在较深层所占比重微乎其微，甚至被忽略。

由于本身训练数据的不同，并且在 TensorFlow 中我们使用了 GPU，而在基

于传统缺陷检测的时候我们并没有使用 GPU，因此，我们并不对二者的速度进行比较。但是，显然，基于传统的检测算法已经取得了比较好的速度，而深度学习由于其本身较深的网络参数和较多的网络参数本身速度有限。

综合本章实验结果以及第四章实验结果，我们认为，虽然深度学习非常火爆，并且在非常多的领域取得了令人瞩目的成果。但是传统的机器学习方法仍然有其发挥作用的地方，并且在有些领域无论在速度上还是效果上都可以取得优势，尤其是在对精度要求比较高的精密仪器缺陷检测领域，我们不能忽视传统算法的力量和优势。

第五章 总结与展望

本文就零件表面缺陷检测这一问题，针对传统方法的目标检测手工设计特征困难，效果不佳以及基于深度学习的目标检测方法模型复杂度高、耗时严重并且对小目标效果差这一现象，提出了基于法线图的目标检测方法。这一方法首先设计了一套提取目标法线图的方法，接着将法线图应用到传统方法和深度学习的检测方法中，设计和实验了一套速度快、效果好的解决零件表面缺陷的方法。

5.1 本文工作总结

本文提供了一种获取零件表面法线信息的方法，并且，在提供了这一方法的同时，本文设计和实现了一套相配合的硬件设备，该设备通过机、LED 灯带、滑轨、偏光镜和滤光膜等不同硬件设备的组合搭配，能够得到材质不同方向光源条件下拍摄得到的照片。我们的算法针对这一组原始数据，首先对照片做白平衡校正、色彩校正、畸变校正、光线补偿校正等针对这一目标的校正算法，弥补了硬件设备本身的不足和缺陷，提高了输入数据的质量以及最终算法的效果。最后，我们使用我们设计的算法能够得到平滑、精细的法线图。

在将法线图应用的传统的检测方法中，我们不是单纯的使用现有的目标检测方法，而是针对最终要求，首先处理了零件法线图，去除了拍照过程中必须的背景，提取了零件主表面信息。接着，就缺陷检测要求中的缺陷大小规定以及精密零件缺陷数据少这一问题，我们分析了不同的数据增强方法的可行性，采用了合适的数据增强方法，扩充数据库。在设计和提取手工特征的时候，我们实验了不同的图像特征，并首次将法线图本身作为输入，并对不同的特征进行组合，测试和对比了不同特征的效果以及速度。在设计和训练模型的时候，我们针对精密零件难以获得缺陷以及使用滑动窗口分割后数据更加不平衡的问题，首先训练不同的 SVM，使用 Adaboost 的方法将其组合成强分类器，并使用 Cascade 组合了多个 Adaboost 训练出来的模型，不仅可以改善数据不平衡的问题，同时针对缺陷检测中漏检造成的损失远大于漏检这一问题做出了优化，

取得了良好效果。同时，我们不仅对比了不同特征的效果，同时将法线图实验的结果和不同灯光条件下拍照得到的照片作为输入得到的结果进行了对比，并分析了问题所在。

同样，本文还将法线图应用到深度学习中，训练了一个深度学习的目标检测模型，并且，将深度学习模型的检测结果与我们的基于传统的检测方法做了对比，分析了二者的各自优缺点以及原因。本文还探讨了不同的目标检测方法的各自优缺点，深入研究了不同的手工特征以及深度学习中的各种概念和模型。最终得到适用于这一问题得较好的检测模型。

5.2 未来工作展望

本文虽然取得了一定进展，但是仍有很多可以改进的地方。

首先是基于传统的缺陷检测算法，我们虽然尝试了不同特征并将其进行组合，但是没有原始图片做相应的处理，没有将我们之后的模型应用到普通的图片输入中。并且，在分类过程中，缺陷也有非常多的种类，目前我们只是简单地将这些缺陷全部认为是缺陷，变为一个单一的二分类问题，而没有对每一类缺陷进行划分，这在之后的工作中可以做进一步改进。

再次对于深度学习算法，目前有很多端到端的目标检测算饭比如 Yolo、SSD 等，或者是 R-CNN 这种首先获取推荐候选区域，过滤不必要的滑动窗口这类方法，但是考虑到这些模型的复杂度和数据量的规模，我们并没有进行深入研究和测试，我们采用的仍然是首先用滑动窗口分割图片，然后分类的方法，所以这一问题有待研究。其次，我们并没有针对去缺陷目标过小这一问题对深度学习模型提出优化，这是一个非常好的可以优化的地点，我们打算在接下来的工作中结合深度学习中目标检测算法针对这一问题继续优化。

致 谢

从选择了读研，到如今论文完成之际，时光转眼走过三年。这一千天中，有数不清的欢笑与汗水，收获与感动。也有太多的人和事情需要我去感谢！

首先，最需要感谢的是我的导师郭延文教授，感谢郭老师的悉心指导与敦敦教诲。在论文的完成中，郭老师不仅在选题上给予了我悉心的指导，更是在实验和撰写过程中给出了独特的意见和帮助。作为一名教授，郭博士在学术上拥有极高造诣，是我学术道路上的引路人和指导者，作为一名老师，郭老师以严谨的态度和勤奋的精神以身作则的树立了我工作、学习和生活上的典范。从老师那里，我得到不仅仅是宝贵的知识财富，更是受益一生的做人道理。

我还要感谢周文喆师兄和吕高建师姐，我们设计和实现了获取物体表面材质的设备和算法，工程量庞大，仅凭我一人之力可能难以完成，是和他们的通力合作下，才完成了这一工作任务，在和他们的合作中我也学到了很多的知识，获得了快速的成长。要感谢张扬师兄，在将法线信息应用到目标检测的过程中，他以渊博的学术见解和深厚的编程功底为我提供了极大的帮助。感谢同级的同门于宵和黄凯同学，我们同一年级入学，一起学习和生活了诸多时间，并在最后的论文阶段互帮互助，顺利完成了各自的论文，感谢他们带给我的帮助、快乐和感动。

感谢刘明明、朱捷、潘飞、强玉庭、张宏杰、贺敬武五位博士师兄在三年来的帮助和指导，感谢马晗师兄的热心帮助，感谢张可心师妹、张云峰师弟、陈钊民师弟、韩旭师妹，感谢张慧明师弟、陈玉念师妹、罗曼琳师妹、徐春雷师妹，感谢能够相遇在实验室的所有人，感谢你们。

感谢我的舍友孙佳俊、宋仁杰，三年的舍友情弥足珍贵，感谢在南京大学遇到的熊宇、王铖燕等所有的朋友们，是和你们的相遇填满了我研究生生涯的宝贵时光，感谢我遇到的所有的人。

最后，我要感谢我的家人，感谢家人给予我的关心与照顾，是他们的支持和陪伴，让我有勇气和力量一路走下去，感谢你们，我爱你们。

感谢！