

哈 尔 滨 工 业 大 学

## 硕士学位论文中期报告

题 目：基于机器视觉的雾化器装配质量检测算法  
研究

院       （系） 机电工程与自动化学院

学       科   机械工程

导       师   胡泓教授

研   究   生   王建坤

学       号   17S153652

中期报告日期 2019.03.07

研究生院制

# 目 录

1 课题主要研究内容及进度情况 .....	1
1.1 课题主要研究内容 .....	1
1.2 课题进度情况 .....	1
2 目前已完成的主要研究工作及结果 .....	1
2.1 构建数据集 .....	1
2.1.1 数据采集 .....	1
2.1.2 数据增强 .....	2
2.2 基于图像处理的检测算法研究 .....	3
2.2.1 检测目标定位及 ROI 设置.....	3
2.2.2 工件缺失检测 .....	4
2.2.3 棉芯缺失检测 .....	4
2.2.4 金属片缺失检测 .....	5
2.2.5 金属丝异常检测 .....	6
2.2.6 图像处理检测算法性能实验分析 .....	7
2.3 基于卷积神经网络的检测算法研究 .....	9
2.3.1 卷积神经网络 .....	9
2.3.2 模型选择 .....	10
2.3.3 模型训练 .....	11
2.3.4 卷积神经网络检测算法性能实验分析 .....	13
2.3.5 基于 Siamese 网络检测算法的研究 .....	13
2.4 检测软件与云端数据管理系统设计 .....	15
2.4.1 检测软件设计 .....	15
2.4.2 云端数据管理系统设计 .....	16
3 后期拟完成的研究工作及进度安排 .....	18
4 存在问题、困难及其解决方案 .....	18
4.1 存在问题与困难 .....	18
4.2 解决方案 .....	18
5 如期完成全部论文工作的可能性 .....	18

## 1 课题主要研究内容及进度情况

### 1.1 课题主要研究内容

本课题来源于深圳某自动化公司的“雾化器自动装配系统”项目，本文主要完成该系统中基于视觉的雾化器装配不良检测，包括检测算法的设计和检测软件的开发。

本课题的具体研究内容如下：

（1）构建雾化器装配不良图片数据集。研究雾化器的装配工艺流程和产品质量要求，划分雾化器装配不良的类别，采集图片数据，并研究图片数据增强的相关方法。

（2）基于传统图像处理的检测算法研究。分析各种装配不良类别的特点及检测要求，研究相应的图像处理检测算法。

（3）基于卷积神经网络的检测算法研究。研究基于深度学习的图像识别算法，设计符合工业检测特点的端到端的深度学习检测算法。

（4）检测软件系统的开发。设计相应的检测软件系统，包括检测软件、数据库服务器、检测数据云端管理系统。

### 1.2 课题进度情况

从去年7月份开始，在老师的指导和企业的支持下，本课题的各项工作都已陆续的展开。目前已经完成的工作如下：

（1）构建装配不良的图片数据集，共采集了382张图片，并对图像进行了适当的数据增强，增加样本数量，提高算法的鲁棒性。

（2）对于不同的装配不良类别，设计了相应的图像处理检测算法，并根据实际生产要求设计了完整的检测流程。

（3）完成了深度学习算法的学习，使用 MobileNet 网络结构搭建卷积神经网络算法，并进行模型的训练和算法效果的评估。

（4）基于 python 语言、opencv 图像处理库、MySQL 数据库管理系统、QT 框架和 Django 框架设计了检测软件和检测数据云端管理系统。

## 2 目前已完成的主要研究工作及结果

### 2.1 构建数据集

#### 2.1.1 数据采集

根据装配工艺流程和实际生产实验，本文把装配质量分为五种类别，分别为：

正常、工件缺失、棉芯缺失、金属片缺失和金属丝位置异常。我们在样机上进行了图片样本的采集，共采集了 382 张图片。各类别的样本数量见表 2-1。

表 2-1 各装配不良类别样本量

不良类别	正常	工件缺失	棉芯缺失	金属片缺失	金属丝异常
样本数	234	78	22	30	18

### 2.1.2 数据增强

为了更好的适应检测环境的变化，为深度学习算法提供更多的数据，有必要进行数据增强，增加样本的多样性和数量。

根据检测项目的特点，本课题提出了以下几种数据增强的方法：

（1）**位置偏移** 由于夹具和相机的相对位置会因为安装精度产生微小的偏移，随机对图片进行上下左右的微小偏移，可以提高算法对安装位置的适应性，位置偏移示例见图 2-1 b)。

（2）**亮度变换** 不同的生产环境及光源的差异都会对图像的亮度产生影响。为了使算法能够适应亮度的变化，有必要对图像进行适当的亮度变换增强，亮度变换示例见图 2-1 c)。

（3）**对比度变换** 装配检测是嵌入在流水线中的，不可避免会有一些振动，导致成像时对比度略有差别，对图像进行对比度变换增强可以增加算法的鲁棒性，对比度变换示例见图 2-1 d)。

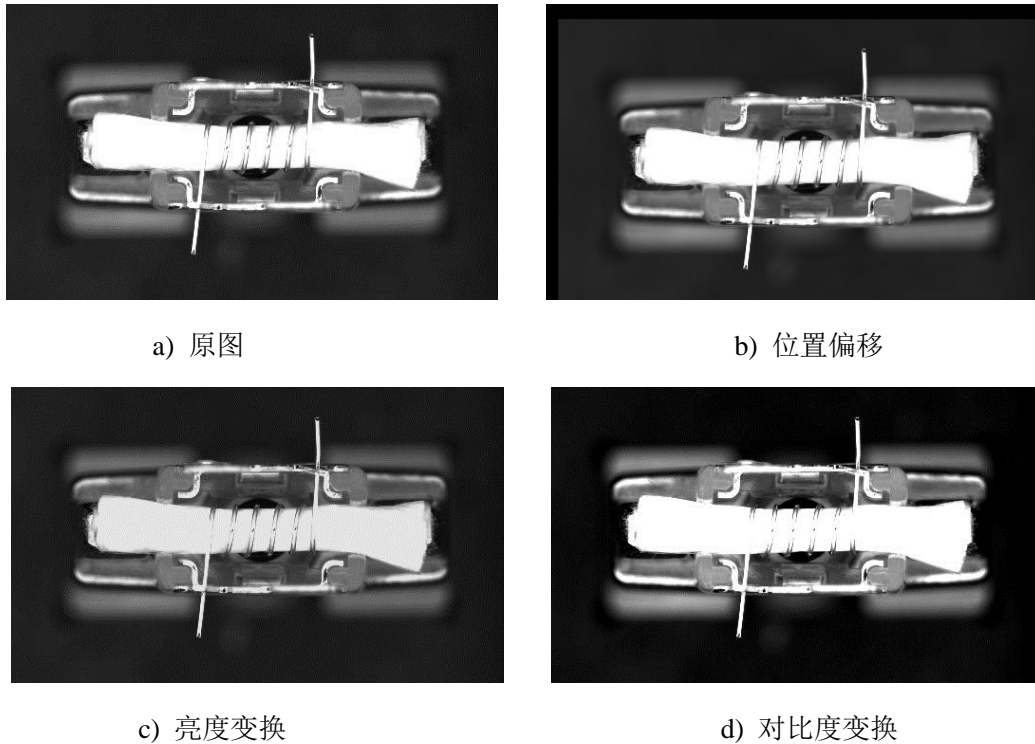


图 2-1 数据增强

## 2.2 基于图像处理的检测算法研究

### 2.2.1 检测目标定位及 ROI 设置

在对装配件进行检测时，首先要对检测目标进行定位。装配件是固定在夹具的里面，可以先定位夹具的位置，然后根据装配件与夹具的相对位置和需要检测的项目来设置 ROI 进行装配不良的检测。

对于夹具位置的定位，由于图像的背景为黑色，通过式 (2-1) 对图像进行二值化，分割出夹具和装配件的区域。二值化的结果如图 2-2 b) 所示。

$$g(x,y) = \begin{cases} 255 & f(x,y) > t \\ 0 & f(x,y) \leq t \end{cases} \quad (2-1)$$

式中  $g(x,y)$  —— 图像二值化后  $(x,y)$  处的像素值；

$f(x,y)$  —— 灰度图  $(x,y)$  处的像素值；

$t$  —— 二值化阈值；

由于装配件的金属丝会超出夹具的范围，对夹具进行定位时应该对其进行处理。通过对图像进行开运算可以消除突出的金属丝，开运算是先对图像进行腐蚀后膨胀的操作，作用是消除细小物体。其中，腐蚀是去除白色区域的边缘地带，膨胀是扩张白色区域的边缘。开运算结果如图 2-2 c) 所示。

通过对图像进行轮廓查找，可以得到夹具的外包矩形轮廓。轮廓查找结果如图 2-2 d) 所示。

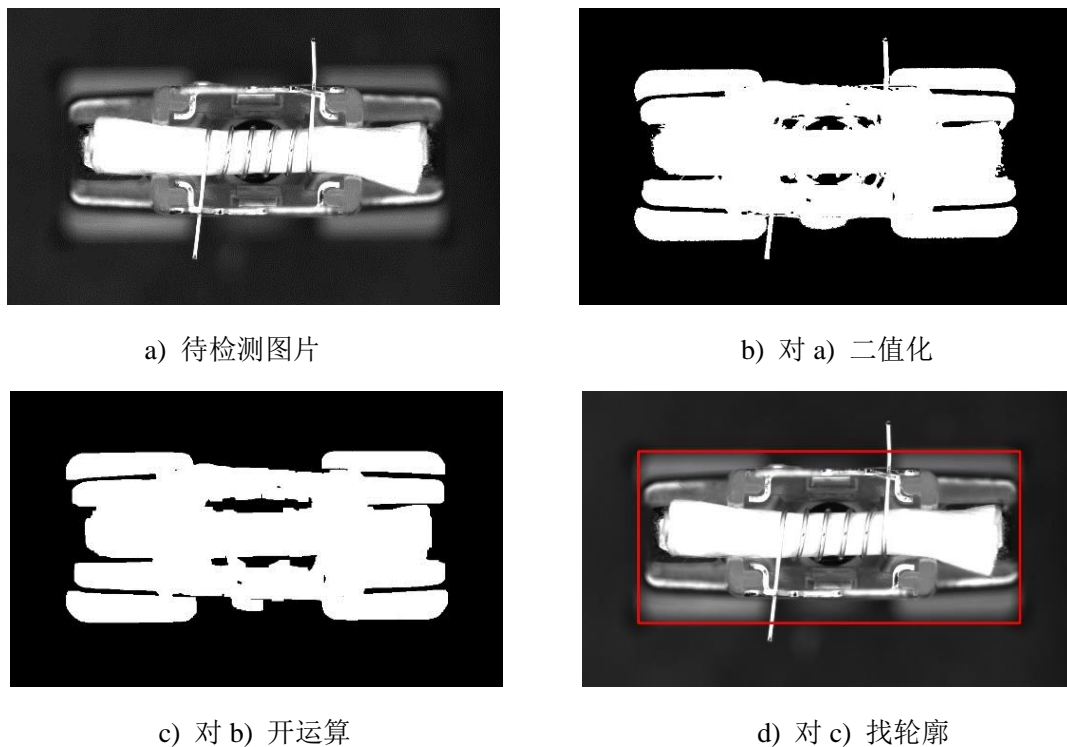


图 2-2 目标定位

在工业检测中，对于不同的检测项目通常采用设置检测的感兴趣区域（ROI）来进行检测。根据检测要求和不良样本的分析，本文总共设置了八个 ROI，如图 2-3 所示。根据从左到右、从上到下的顺序，2、7 用来进行金属片缺失检测，4、5 用来进行棉芯缺失检测，1、3、6、8 用来进行金属丝异常检测。

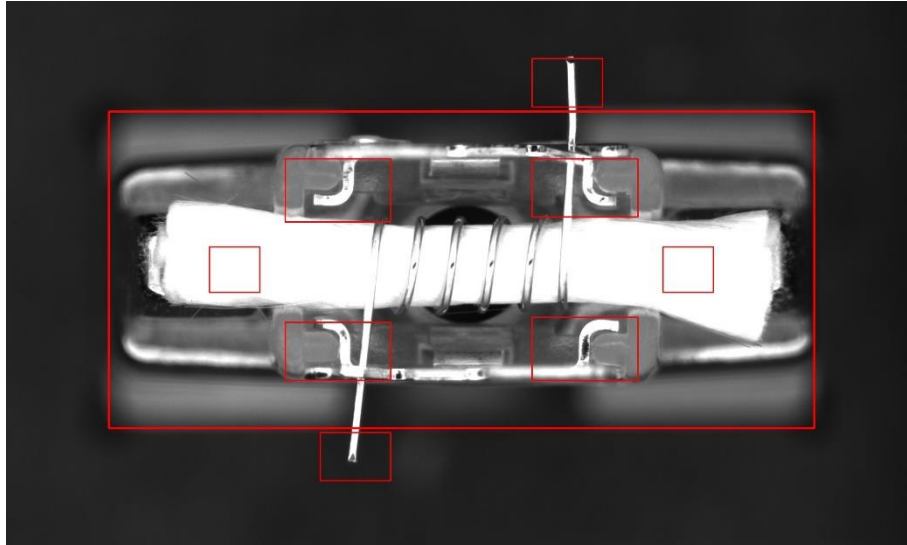


图 2-3 ROI 设置

### 2.2.2 工件缺失检测

装配件在装配过程中需要经过多次的夹取，可能会出现夹取失败或者在夹取过程中丢失的情况，导致在检测工位上没有装配件只有夹具。

由于夹具是由多个部分组成，在进行夹具轮廓查找时不能得到完整的轮廓，而是检测到多个小轮廓。本文根据这个特点来进行工件缺失的检测，通过设置轮廓面积阈值进行轮廓筛选，可以过滤小面积的轮廓，如果存在工件缺失的情况，在进行轮廓查找是就找不到轮廓，因此可以根据目标定位时轮廓查找的结果来判断工件是否缺失。工件缺失检测如图 2-4 所示。

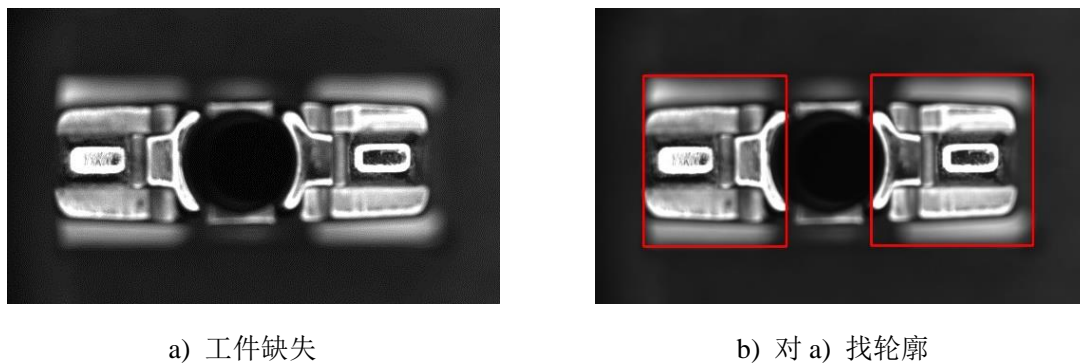


图 2-4 工件缺失检测

### 2.2.3 棉芯缺失检测

由于棉芯是由金属丝包裹着，为了不使金属线圈变形，上料的夹取力较小，

容易导致在搬运的过程中丢失。

对于棉芯缺失检测，本课题采用的是像素统计法，棉芯检测 ROI 样本如图 2-5 所示。棉芯在成像时呈亮白色，如果存在棉芯，则棉芯检测 ROI 应该都为白色像素。通过统计棉芯检测 ROI 中的白色像素数量比例来判断棉芯是否缺失。如果比例小于设定的阈值则判定为棉芯缺失，反之。

统计白色像素数量的过程是：对图片进行二值化，设置像素值阈值，对于高于该阈值的像素点认定为白色，其像素值置为 1，否则置为 0。对整张图片进行求和即可得到图像的白色点数。

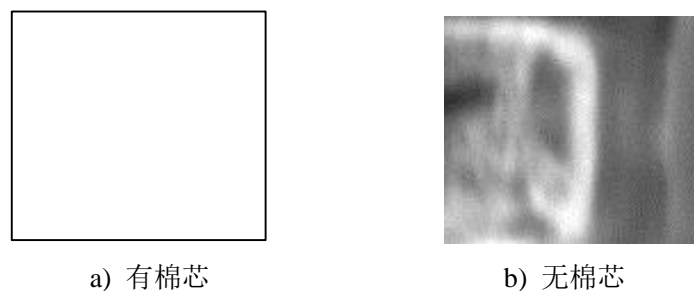
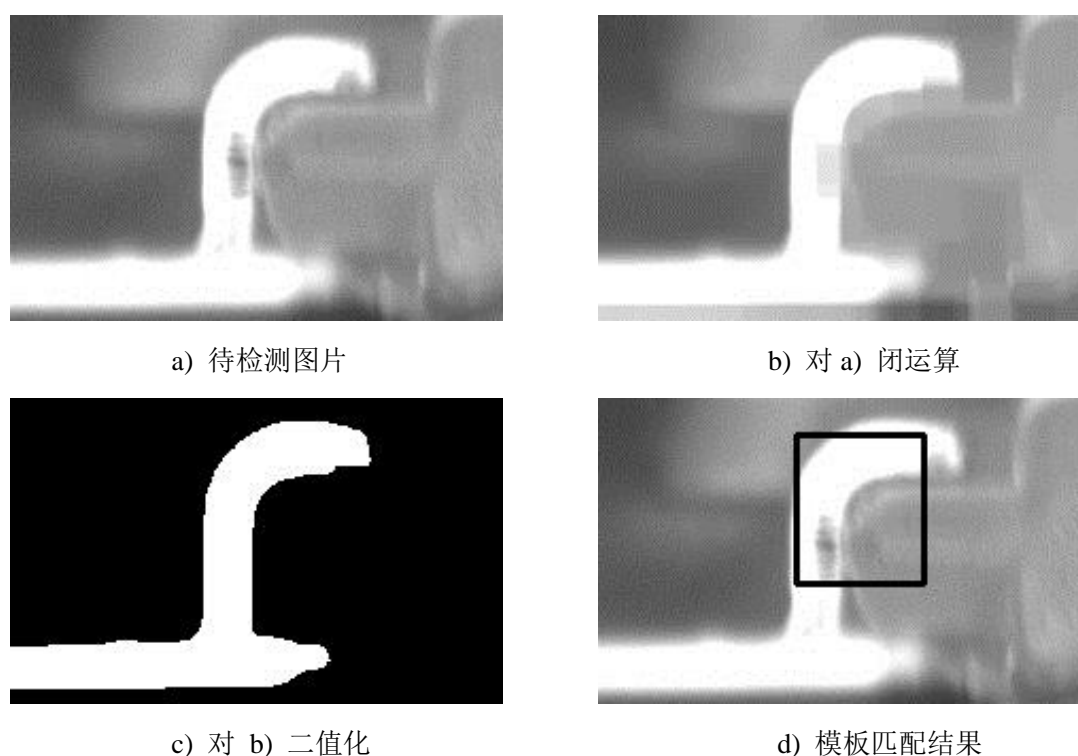


图 2-5 棉芯缺失检测

#### 2.2.4 金属片缺失检测

对于金属片的检测，本课题采用模板匹配。金属片有个卡爪，只要能检测到卡爪就能判定为金属片是存在的。由于成像的原因，在金属片卡爪上可能会出现一些偏暗的区域，需要对图片进行预处理。

本课题采用闭运算进行预处理，闭运算是先膨胀再腐蚀的操作，作用是消除小孔洞。对金属片检测 ROI 进行闭运算的结果见图 2-6 b)。





e) 模板图片 1



f) 模板图片 2

图 2-6 金属片检测

进行模板匹配前先对图片进行二值化,使得除了金属片以外的区域变为黑色,见图 2-6 c),这样能使得模板匹配不受背景的干扰,提高模板匹配的准确性和稳定性。我们所使用模板图片也是经过二值化操作的,两个不同地方的模板图片如图 2-6 e)、f)所示。

模板匹配方法有很多,本课题选择了归一化平方差法。归一化平方差公式见式 (2-2),这种方法具有简单快速的特点。归一化平方差法的原理是,首先计算模板图像和待检测图像的像素值平方差,然后对结果进行归一化处理,使其取值范围区间变换为[0,1],值越小越匹配,模板匹配结果如图 2-6 d) 所示。

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2}{\sqrt{\sum_{x', y'} T(x', y')^2} \sqrt{\sum_{x', y'} I(x + x', y + y')^2}} \quad (2-2)$$

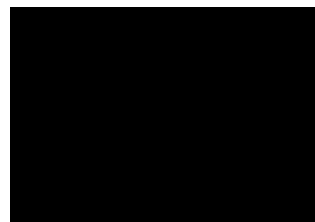
式中  $T(x', y')$  —— 模板图在  $(x', y')$  处的像素值;  
 $I(x + x', y + y')$  —— 待匹配图在  $(x + x', y + y')$  处的像素值;

### 2.2.5 金属丝异常检测

金属丝异常有两种情况,一种是金属丝过短,一种是由于金属丝没有压好导致的位置偏离。对于第一种情况,采用的是像素统计法来判断,与棉芯检测的方法类似,这里不再赘述,金属丝过短检测 ROI 样本图如图 2-7 所示。



a) 有金属丝



b) 无金属丝

图 2-7 棉芯缺失检测

对于第二种情况,本文通过测量金属丝与金属片卡爪竖直边的距离来判定。当距离偏离正常范围时即判定为异常。分析图片,金属丝与金属片卡爪处会有大量亮白色区域,其它区域大部分呈灰色。准确的找到两个亮白色的区域就可以进行距离的测量。

首先对待检测的图片进行预处理,包括闭运算和二值化。闭运算消除黑色孔洞,二值化操作分割出亮白色区域。预处理结果如图 2-8 所示。



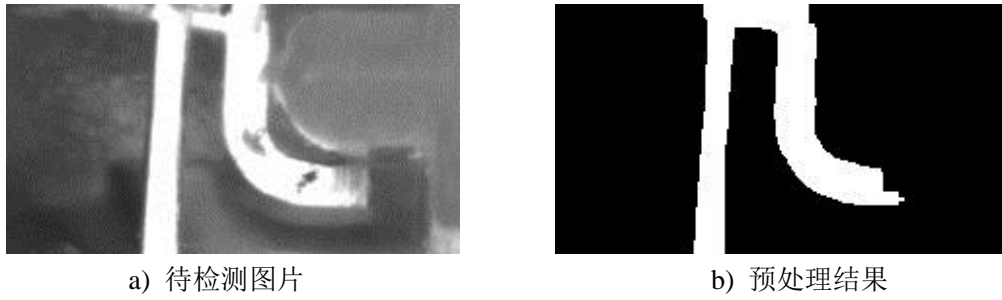


图 2-8 金属丝距离测量预处理

对经过预处理后的图片，统计每一列白点的数量进而绘制曲线。在金属丝和金属片卡爪竖直边处会有大量的白点，出现两个极大值。因为图片存在其它白色区域和噪点导致曲线不够平滑，难以找出正确的极大值。

为了准确的找出极大值，本文对统计数列进行了一些预处理。第一步是对数列的数值进行过滤，使小于某一个数值的数值变为 0，得到金属丝和金属片卡爪竖直边之间的区域。第二步进行数列的平滑，找到两个极大值之后，就可以测出二者的距离。统计数列预处理过程曲线化如图 2-9 所示。

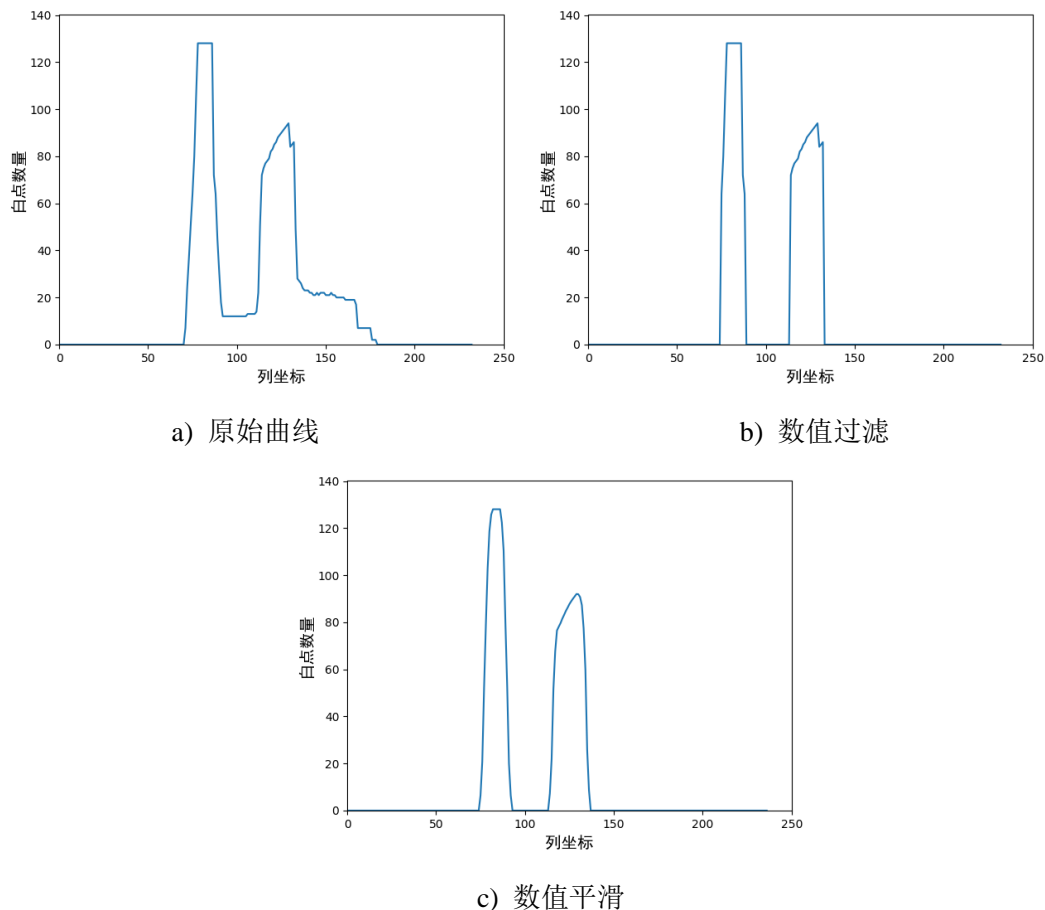


图 2-9 曲线处理

### 2.2.6 图像处理检测算法性能实验分析

工业检测中评价检测算法的性能的两个重要指标是检出率和误检率。检出率

指该类别被正确检测出来的比例，误检率指非该类别检测为该类别的比率。

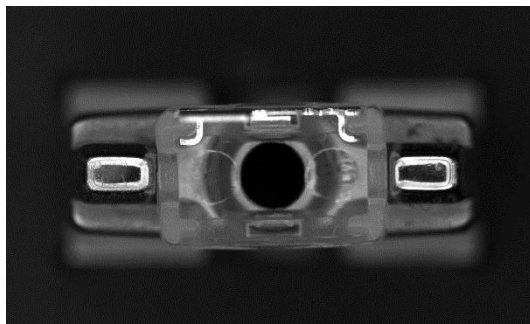
使用本文提出的基于传统图像处理的检测算法对所采集的图片数据集进行检测性能测试实验。测试的样本数量见表 2-1，各种不良的检出率和误检率见表 2-2。

表 2-2 传统图像处理算法检测性能表

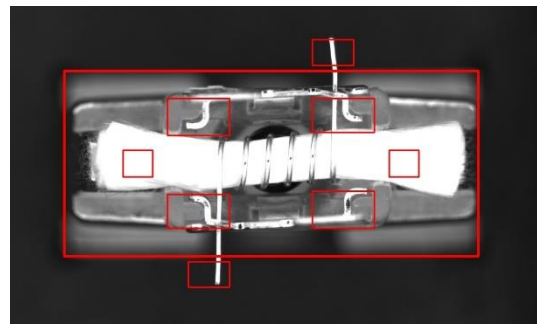
类别	检出率	误检率
正常	96.16%	-
工件缺失	100%	0%
棉芯缺失	72.72%	0%
金属片缺失	100%	0.85% / 2/234
金属丝异常	100%	2.99% / 7/234

基于传统图像处理的算法有大量的参数需要设置，不同的参数设置会得到不同的结果。由于工业检测中通常对检出率的要求比较严格，本文确定参数的原则是在保证高检出率的条件下，尽可能的降低误检率。

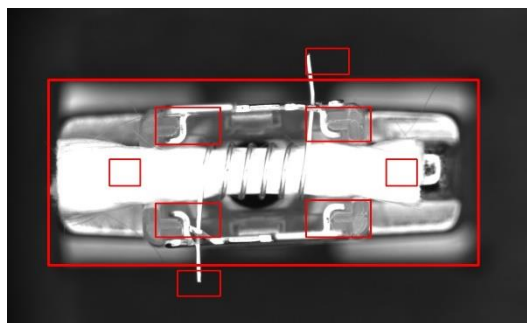
棉芯漏检原因是漏检的样本被检测为工件缺失。因为个别没有棉芯的图片偏暗，轮廓查找失败。金属片误检原因是 ROI 不准确（金属卡爪的区域不完整）。由于不同的工位夹具的成像尺寸不一样，装配件在夹具中的位置会有旋转，导致装配件在夹具中的相对位置有一些偏差。金属丝误检原因，一方面是由于 ROI 不准确，另一方面是因为 ROI 中可能包括一些干扰物，导致测距不准确。漏检样本示例见图 2-10。



a) 棉芯漏检样本



b) 金属片误检样本



c) 金属丝异常误检样本

图 2-10 漏检误检样本图

## 2.3 基于卷积神经网络的检测算法研究

### 2.3.1 卷积神经网络

卷积神经网络（CNN）是一种深度学习算法，主要应用于图像识别。卷积神经网络的重要组成部分有卷积、激活函数和分类函数。

(1) **卷积** 卷积的作用是提取图像特征。对图像进行卷积运算的原理是，让卷积核在图像上以一定的步长进行滑动，计算每个窗口区域的卷积值。不同的卷积核大小、卷积核个数和滑动步长可以得到不同的图像特征。卷积分为二维卷积和三维卷积，计算公式分别见式（2-3）和式（2-4）。

$$H(i, j) = \sum_m \sum_n F(m, n) G(i - m, j - n) \quad (2-3)$$

$$H(i, j) = \sum_k \sum_m \sum_n F_k(m, n) G_k(i - m, j - n) \quad (2-4)$$

卷积神经网络通常由多层卷积组成，需要大量的运算，往往很难用于实时检测。为了减少算法的运算量，学者们提出了深度分离卷积，在保证模型准确性偏差不大的情况下大幅降低算法的运算量。深度分离卷积是将正常的卷积拆分成深度卷积和逐点卷积，拆分原理如图 2-11 所示。

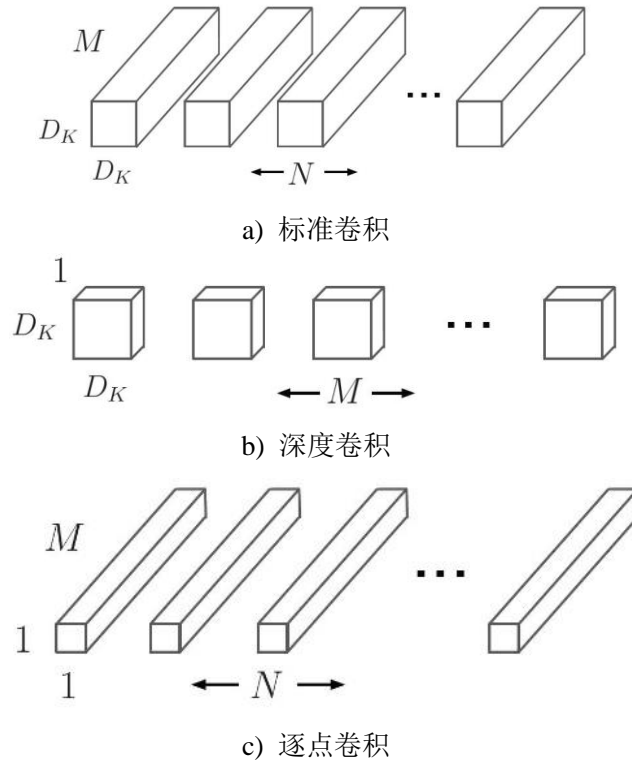


图 2-11 深度可分离卷积

假设有一个  $D_F \times D_F \times M$  的特征图。标准的卷积层包括  $N$  个  $D_K \times D_K \times M$  三维卷积核，每一个卷积核需要对输入特征图的所有通道进行卷积运算得到输出

特征图的一个通道， $N$  个卷积核需要  $D_K \times D_K \times M \times N \times D_F \times D_F$  的计算量。深度可分离卷积把标准的卷积层拆分为两步，第一步使用  $M$  个  $D_K \times D_K$  的二维卷积核分别对输入的特征图进行卷积运算，每个二维卷积核分别和对应的输入通道进行卷积运算，计算量为  $D_K \times D_K \times M \times D_F \times D_F$ ；第二步使用  $N$  个  $1 \times 1 \times M$  的二维卷积对第一步的输出特征图进行卷积运算，计算量为  $M \times N \times D_F \times D_F$ 。这两种方法都能得到相同尺寸的输出。根据公式 (2-5) 计算二者计算量的比率，其中卷积神经网络中的卷积核个数通常较多。

$$rate = \frac{D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F}{D_K \times D_K \times M \times N \times D_F \times D_F} = \frac{1}{N} + \frac{1}{D_K^2} \approx \frac{1}{D_K^2} \quad (2-5)$$

由上式可见，如果使用  $3 \times 3$  的卷积核，深度分离卷积的计数量约为标准卷积层计算量的九分之一。

**(2) 激活函数** 激活函数 (Activation Function) 通常是作用在卷积运算之后的，负责将卷积的输出按照激活函数关系进行映射。通过使用激活函数，使得卷积神经网络具有了非线性，进而使得卷积神经网络可以逼近更复杂的非线性函数，增加了网络的学习能力。目前卷积神经网络常用的激活函数有 Sigmoid 见式 (2-6) 和 ReLu 见式 (2-7)。

$$y = \frac{1}{1 + e^{-x}} \quad (2-6)$$

$$y = \max(0, x) \quad (2-7)$$

**(3) 分类函数** 分类函数是作用在整个卷积神经网络的最后面，对计算的结果进行归一化得到每个类别的概率。分类任务可以分为二分类和多分类。二分类通常使用 Sigmoid 函数进行分类，得到其中一个类别的概率，通过概率和为 1 的关系可以求得另一类别的概率。多分类通常使用 softmax 函数进行分类，得到每个类别的概率，softmax 函数见式 (2-8)。

$$S_i = \frac{e^{V_i}}{\sum_i^C e^{V_i}} \quad (2-8)$$

式中  $V_i$  —— 上一层第  $i$  个类别的计算结果；  
 $C$  —— 总类别数；

### 2.3.2 模型选择

根据工业检测的特点，本课题选择了 MobileNet 卷积神经网络模型。MobileNet 是一个基于深度可分离卷积的模型，跟其它模型相比，在准确率相差不大的情况下它的计算量远远少于其它模型。MobileNet 的具体网络结构参数见表 2-3。

表 2-3 MobileNet 网络结构

网络层/步长	卷积核形状	输入尺寸
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5 × Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool $7 \times 7$	$7 \times 7 \times 1024$
FC / s1	$1024 \times 5$	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 5$

模型参数中的 Conv 表示标准的卷积，Conv dw 表示深度分离卷积。最后一层的分类类别数根据具体任务确定。

### 2.3.3 模型训练

卷积神经网络模型的参数需要通过数据训练确定。模型训练的两个主要部分是损失函数和优化算法。

(1) **损失函数** 损失函数是用来衡量预测值与真实值的差别。损失函数值越小说明预测值与真实值之间的差别越小，模型学习的结果越好。损失函数对模型的学习效果有很大的影响，选择合适的损失函数可以得到较好的效果。在图像识别领域通常使用交叉熵损失函数，见公式 (2-9)。从公式中可以看出，当预测值越接近于真实值时损失函数值就会越小，反之。

$$L(\hat{y}, y) = -y \log \hat{y} + (1 - y) \log(1 - \hat{y}) \quad (2-9)$$

式中  $y$  —— 真实值；  
 $\hat{y}$  —— 预测值；

(2) 优化算法 优化算法是根据损失函数值和一定规则进行网络参数更新的算法，网络参数的更新公式见式 (2-10)。优化算法的选择决定着模型的训练时间和收敛效果。常用的优化算法有梯度下降、动量梯度下降、RMSprop、Adam 等。本文选择的是 Adam 优化器，它结合了动量梯度下降和 RMSprop，是一个已经被广泛的应用并证明有效的优化器。

$$\omega = \omega - \alpha \frac{\partial L}{\partial \omega} \quad (2-10)$$

式中  $\omega$  —— 参数；  
 $\alpha$  —— 学习率；  
 $L$  —— 损失函数；

学习率也是一个重要的参数，分为固定和动态两种方式。动态学习率的好处是在训练模型的前期可以设置较大的学习率加快训练的速度，在后期学习率会不断的衰减可以稳定的收敛到更优解。动态学习率有多种，本文使用的是阶梯下降，即每隔一定的迭代次数减小一次学习率。

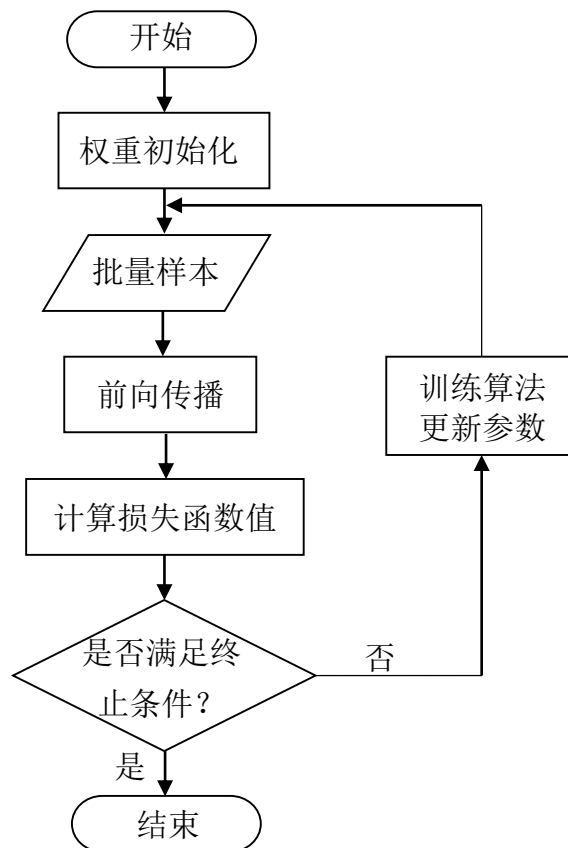


图 2-12 模型训练流程图

（3）模型训练 模型的训练是一个不断迭代的过程，可以通过设置损失函数阈值或训练的最大迭代次数来结束。训练流程图见图 2-12。

#### 2.3.4 卷积神经网络检测算法性能实验分析

深度学习算法的预测结果通常被划分为四类：真正例（TP）、假正例（FP）、真反例（TN）、假反例（FN）。正例被预测为正例是真正例，反例被预测为正例是假正例，反例被预测为反例是真反例，正例被预测为反例是假反例。常用的评估指标有准确率（Accuracy）、精确率（Precision）、召回率（Recall）、F1 系数，公式分别见式（2-11）、式（2-12）、式（2-13）和式（2-14）。

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (2-11)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2-12)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2-13)$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2-14)$$

使用卷积神经网络检测算法对我们数据集进行检测，算法的检测性能如表 2-4 所示。

表 2-4 卷积神经网络算法检测性能表

类别	精确率	召回率/检出率	F1	误检率
正常	98.63%	99.31%	98.97%	-
工件缺失	100%	100%	100%	0%
棉芯缺失	100%	100%	100%	0%
金属片缺失	94.44%	97.14%	97.14%	0.69%
金属丝异常	100%	83.33%	90.91%	0%

可以看出，对比基于传统的图像处理检测算法，卷积神经网络算法的分类准确率和误检率都得到了提升，但是检出率会有一点降低。主要原因是卷积神经网络算法是平等的对待每一种类别。后续将对检出率做一些优化。

#### 2.3.5 基于 Siamese 网络检测算法的研究

上述基于卷积神经网络的分类方法对于数据集中已有类别的检测效果比较好。但在工业检测中通常是不能预知所有的不良情况，如果采用上述算法进行检测，当出现一个数据集中没有的类别时，算法也会强制的为其预测类别，导致分类错误。本课题基于此提出了基于 Siamese 网络的检测算法。

Siamese 网络是一种相似性度量方法，被成功的应用在人脸识别。Siamese 网络对于样本数量少的识别、分类任务有比较好的效果。传统分类方法是需要确切的知道每个样本属于哪个类，需要针对每个样本有确切的标签。Siamese 网络从数据中去学习一个相似性度量，用这个学习出来的度量去比较和匹配新的未知类

别的样本。

Siamese 网络的主要原理是：通过一个函数将输入映射到目标空间，在目标空间使用简单的距离（欧式距离等）进行对比相似度。在训练阶段，最小化属于相同类别的一对样本的损失函数值，最大化属于不同类别的一堆样本的损失函数值。给定一个映射函数 $G_W$ （卷积神经网络）， $W$ 代表函数中的所有参数。Siamese 网络的目的就是去找一组参数 $W$ ，使得当输入 $X_1$ 和 $X_2$ 属于同一个类别的时候，相似性度量是一个较小的值，当输入 $X_1$ 和 $X_2$ 属于不同的类别的时候，相似性度量较大。这个系统是用训练集中的成对样本进行训练。这里的除了需要可微外不需要任何的前提假设，因为针对成对样本输入，这里两个相同的函数 $G_W$ ，拥有一份相同的参数 $W$ ，即这个结构是对称的，所以将它叫做 Siamese 网络。Siamese 网络结构如图 2-13 所示。

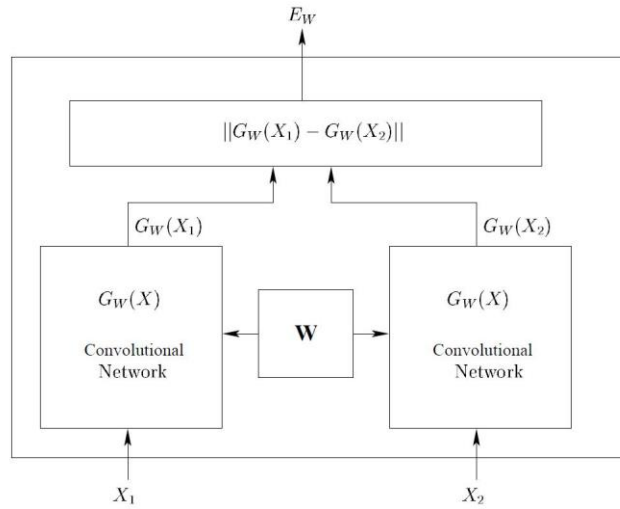


图 2-13 Siamese 结构

相似度量函数有多种，本文拟采用的是 Contrastive loss，公式见式(2-15)。该损失函数可以很好的表达成对样本的匹配程度，也能够很好用于训练提取特征的模型。

$$(W, (Y, X_1, X_2)) = \frac{1}{2N} \sum_{n=1}^N Y D_W^2 + (1 - Y) \max(m - D_W, 0)^2 \quad (2-15)$$

$$D_W = \|X_1 - X_2\|_2 = \left( \sum_i^N (X_1^i - X_2^i)^2 \right)^{1/2} \quad (2-16)$$

公式中 $D_W$ 代表两个样本特征 $X_1$ 和 $X_2$ 的欧氏距离， $Y$ 为两个样本是否匹配的标签， $Y = 1$ 代表两个样本相似或者匹配， $Y = 0$ 则代表不匹配， $m$ 为设定的阈值。

如果两个样本特征的欧式距离越小则表明两者属于同一类别的概率越大。当 $Y = 1$ 时（即样本属于同一类别），欧式距离变大，相似度函数值也变大，增加了损失值；当 $Y = 0$ 时（即样本不属于同一类别），欧式距离变小，相似度函数值变大，增加了损失值；通过优化算法来减小损失值可使模型的预测效果变好。



## 2.4 检测软件与云端数据管理系统设计

### 2.4.1 检测软件设计

检测软件需要实现检测算法、检测信息管理和图形界面的结合。根据软件实现的功能，检测软件可以分为四大部分，分别是：图像处理检测算法，卷积神经网络检测算法，图形用户界面，数据库。

(1) **图像处理检测算法** 本课题采用 OpenCV 图像处理库设计图像处理检测算法，由于装配不良的类型有多种，本课题采用检测顺序是：工件缺失检测，棉芯缺失检测，金属片缺失检测，金属丝异常检测。

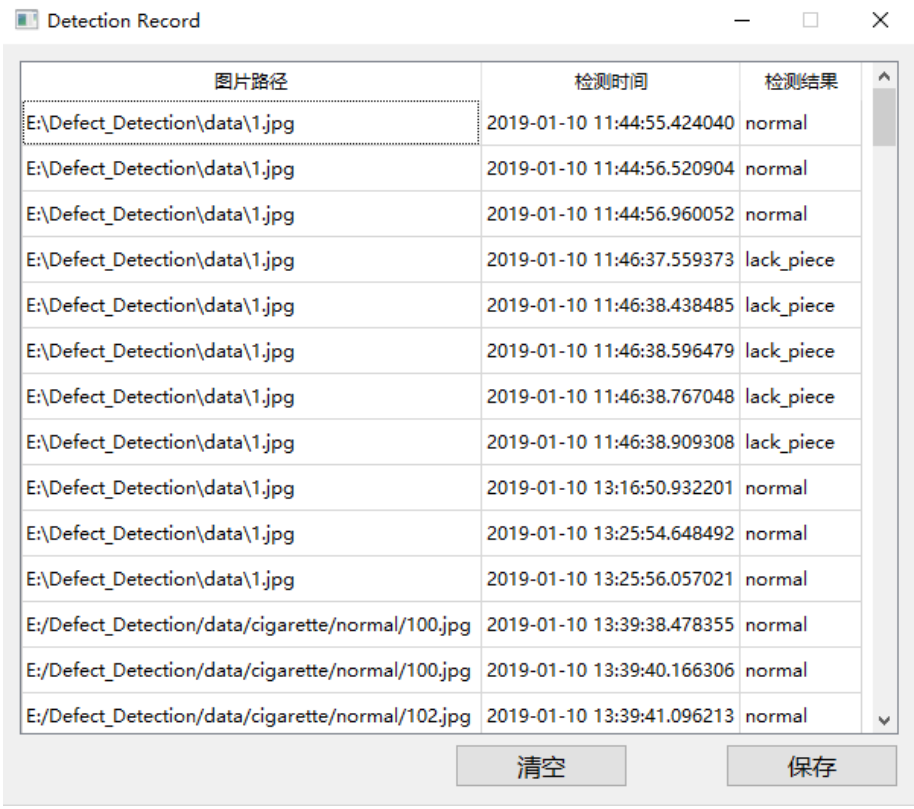
(2) **卷积神经网络检测算法** 本课题采用 TensorFlow 深度学习框架设计卷积神经网络算法。TensorFlow 是谷歌公司开发的开源框架，目前已经是深度学习领域最受欢迎的开发框架。使用卷积神经网络算法进行检测时，需要先加载网络权重文件初始化模型。为了使软件的操作更加方便，选择在软件启动的时候进行相关的初始化。

(3) **数据库** 本课题采用 MySQL 数据库管理系统来存储和管理检测信息。MySQL 数据库管理系统是一种关系型数据库管理系统（RDBMS），使用结构化查询语言 SQL 进行数据库管理。具有简单、小巧和免费的特点，十分适用于本课题的需求。检测软件在每一次检测结束时会生成一条检测记录保存到数据库中，包括检测的图片路径、检测结果和检测时间。



图 2-14 检测软件主界面

(4) 图形用户界面 本课题采用 QT 软件界面设计框架开发图形用户界面，主要功能有检测算法选择、检测对象选择、检测结果显示、软件运行信息提示、检测记录查询与导出。检测软件主界面如图 2-14 所示。检测记录查询与导出界面如图 2-15 所示。



图片路径	检测时间	检测结果
E:\Defect_Detection\data\1.jpg	2019-01-10 11:44:55.424040	normal
E:\Defect_Detection\data\1.jpg	2019-01-10 11:44:56.520904	normal
E:\Defect_Detection\data\1.jpg	2019-01-10 11:44:56.960052	normal
E:\Defect_Detection\data\1.jpg	2019-01-10 11:46:37.559373	lack_piece
E:\Defect_Detection\data\1.jpg	2019-01-10 11:46:38.438485	lack_piece
E:\Defect_Detection\data\1.jpg	2019-01-10 11:46:38.596479	lack_piece
E:\Defect_Detection\data\1.jpg	2019-01-10 11:46:38.767048	lack_piece
E:\Defect_Detection\data\1.jpg	2019-01-10 11:46:38.909308	lack_piece
E:\Defect_Detection\data\1.jpg	2019-01-10 13:16:50.932201	normal
E:\Defect_Detection\data\1.jpg	2019-01-10 13:25:54.648492	normal
E:\Defect_Detection\data\1.jpg	2019-01-10 13:25:56.057021	normal
E:\Defect_Detection\data\cigarette\normal\100.jpg	2019-01-10 13:39:38.478355	normal
E:\Defect_Detection\data\cigarette\normal\100.jpg	2019-01-10 13:39:40.166306	normal
E:\Defect_Detection\data\cigarette\normal\102.jpg	2019-01-10 13:39:41.096213	normal

图 2-15 检测记录界面

#### 2.4.2 云端数据管理系统设计

随着工业智能化时代的来临，工业生产中的检测数据和数据挖掘变得越来越重要，设计相应的云端数据管理系统是十分有必要的。

本课题采用 Django 框架进行云端系统的开发。Django 是一个开放源代码的 Web 应用框架，由 Python 写成。Django 是一个基于 MVC 构造的框架。但是在 Django 中，控制器接受用户输入的部分由框架自行处理，所以 Django 里更关注的是模型（Model）、模板(Template)和视图（Views），称为 MTV 模式。

(1) 模型 即数据存取层。处理与数据相关的所有事务，包括如何存取、如何验证有效性、包含哪些行为以及数据之间的关系等。模型与数据库的连接采用对象关系映射 (ORM, object-relational mapping)，它是以 Python 类形式定义数据模型，可以采用面向对象的思想操作数据库，同时也支持原始的 SQL 语句。

(2) 模板 即表现层。处理与表现相关的操作，如何在页面或其他类型文档中进行显示，模板是可继承的。

(3) 视图 即业务逻辑层。存取模型及调取恰当模板的相关逻辑。视图

是模型与模板的桥梁。

本课题设计的云端数据管理系统包括两个模块。一个是检测记录查询模块，一个是不良图片查询模块。检测记录查询模块可以根据时间段查询检测记录，并且能查看相应的图片，检测记录查询模块界面见图 2-16。不良图片查询模块，可以根据时间段和不良类型查询相应的不良图片，方便进行观察和总结，不良图片查询模块界面见图 2-17。



图 2-16 检测记录查询



图 2-17 不良图片查询

### 3 后期拟完成的研究工作及进度安排

后续需要完成的工作主要是改进和优化卷积神经网络算法，完善检测软件和云端数据管理系统。

具体进度安排表如下：

时间	进度安排
2019.03—2019.05	改进卷积神经网络算法，尝试修改损失函数、使用 Siamese 网络、优化训练策略。
2019.06—2019.07	完善检测数据的数据库和云端管理系统。
2019.08—2019.09	优化相应的检测算法，调试相关检测软件。
2019.10—2019.12	整理研究成果，撰写硕士学位论文，准备答辩。

### 4 存在问题、困难及其解决方案

#### 4.1 存在问题与困难

目前存在的主要问题和困难是卷积神经网络算法虽然在分类的整体准确率上有所提高，算法的鲁棒性也更好，但漏检率还远远不能达到工业检测的要求。

#### 4.2 解决方案

针对目前存在的问题和困难，提出以下解决方案：

（1）查阅相关的文献和解决方案。

（2）改进卷积神经网络算法，包括损失函数、网络结构和训练策略。在保证分类的整体准确率偏差不大的情况下，降低检测的漏检率。

### 5 如期完成全部论文工作的可能性

本课题目前已经完成了小型雾化器装配不良图片的采集、相应图像处理算法的设计、检测软件的初步开发。目前正在进行基于卷积神经网络的检测算法的研究。后期还会进一步对算法和检测软件进行修改和优化。

综上所述，在老师的指导和同学的帮助下，我相信能够如期完成论文全部工作，并取得一定的研究成果。