钢印号码与识别管理系统

1.研究背景及国内外研究现状

2.方案设计

2.1整体设计方案

2.2硬件设计方案

2.3 视觉部分硬件框架

3.算法研究

3.1算法说明

3.2 AI-OCR应用流程

3.3 AI-OCR算法原理

3.4 AI-OCR算法检测过程

3.5文本检测-原理简介

3.6 文本识别-原理简介

4.软件设计

4.1 软件框架设计说明

4.2 软件层级说明

4.3 软件框架设计说明

4.4 软件逻辑模块设计说明

4.5 数据保存及查询功能设计说明

5.系统实现及试验测试

5.1 软件操作界面说明

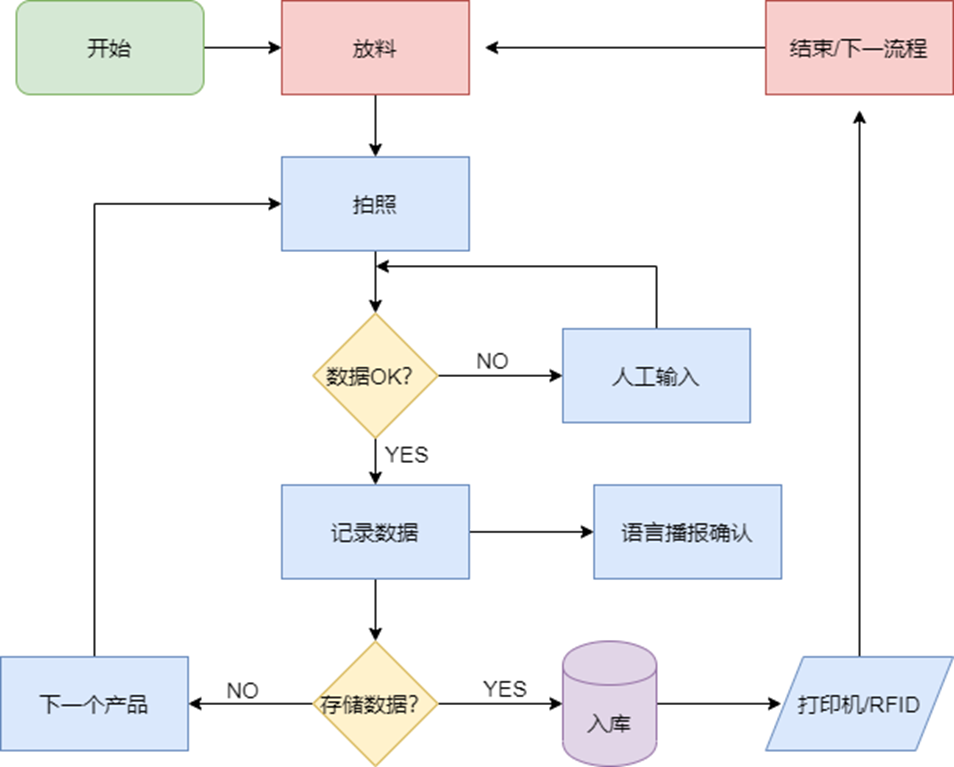
5.2 设计检出图片说明

5.3 实际仿真数据说明

2.方案设计

2.1整体设计方案：

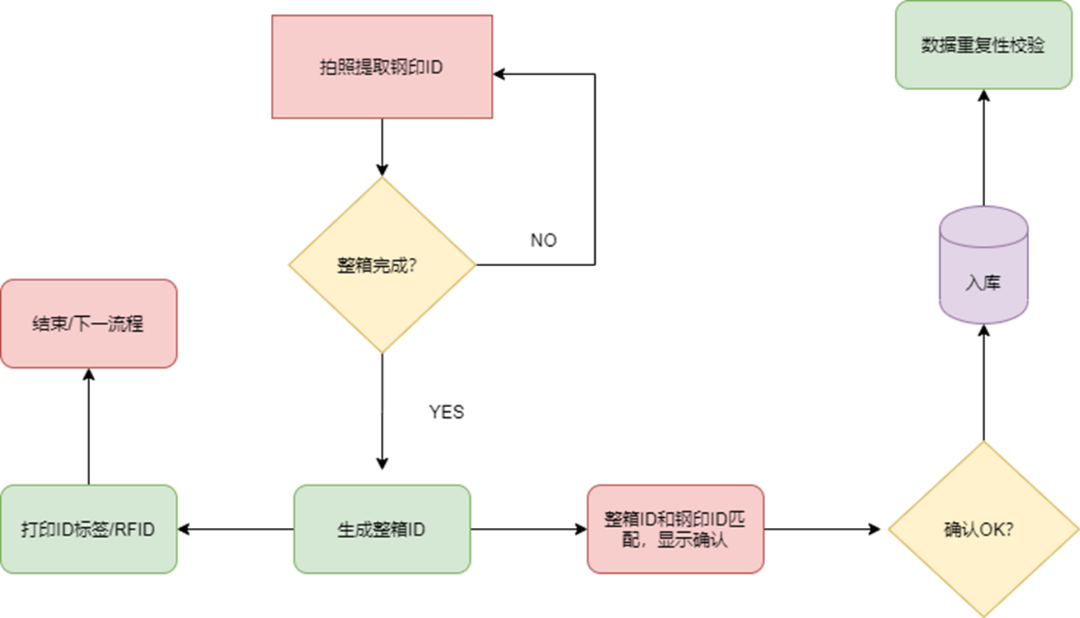
项目的方案设计先从动作流程开始，逐步的分解工艺步骤以及动作流程，弄清楚工艺流程图之后，可以很方便的对机械、软件、控制做出相应的指导程序，动作流程参考如下：

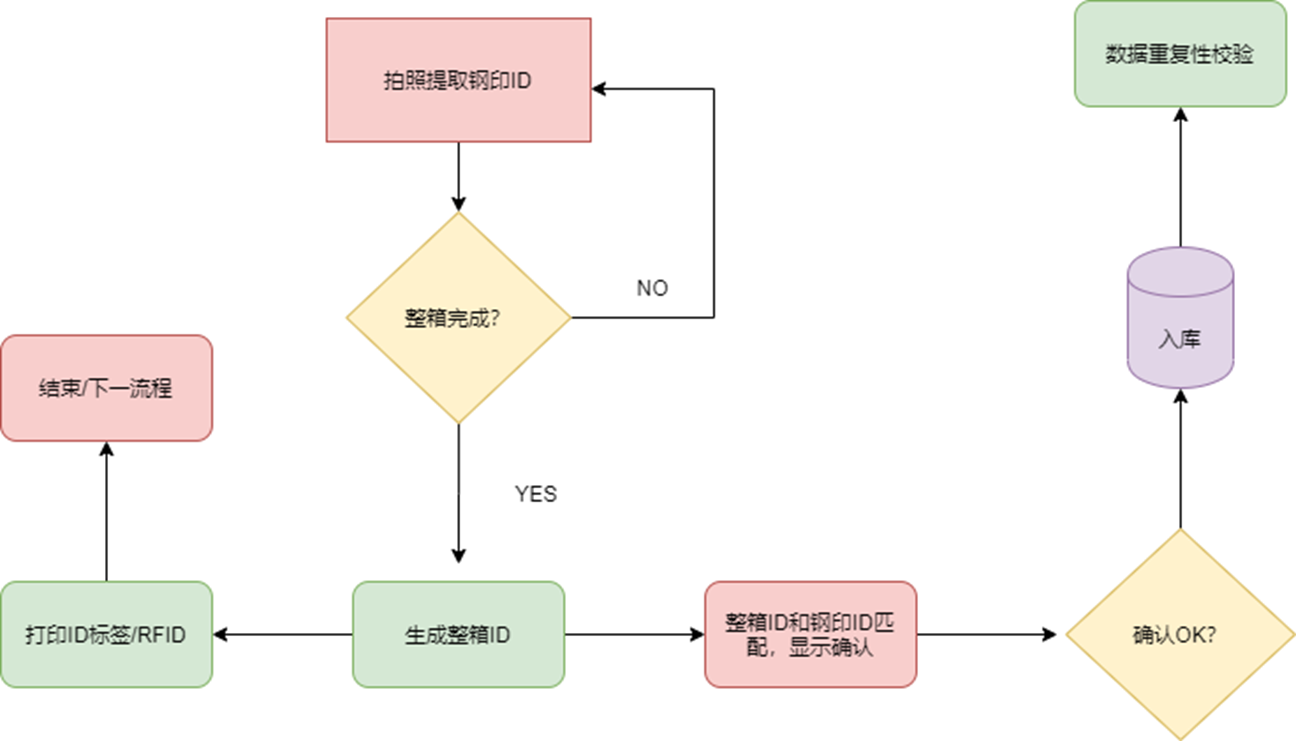
2.1.1动作流程说明

2.1.2数据保存流程说明

本次的项目不只是一个操作的程序，其中还设计到数据的保

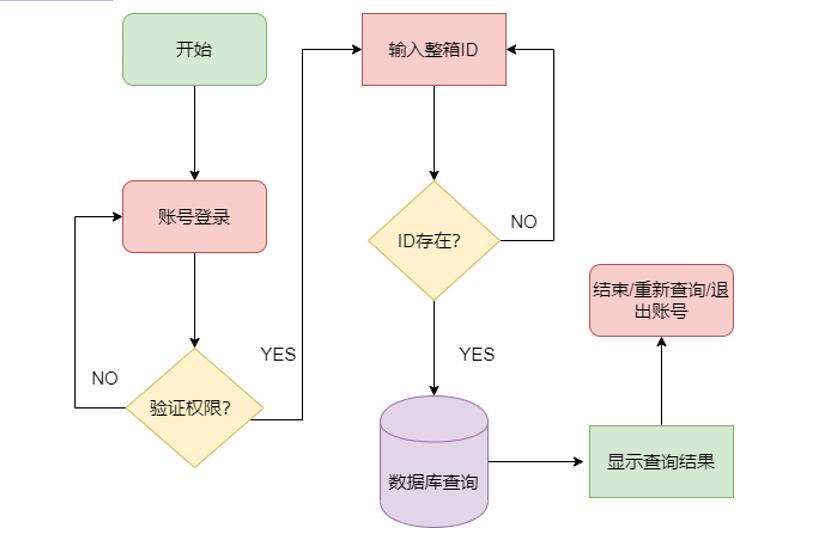
存与查询系统，为了方便程序的开发，需要在前期对所有的数据流做个详细的分析，确保软件开发的合理性以及准确性





2.1.3数据查询流程

数据保存以后，所有信息会保存在本地电脑里面，出于安全的考虑，数据的查询与归档需要做出权限的管理，确保信息安全可靠。



2.2硬件设计方案

2.2.1 站立式操作硬件方案

2.2.1.1此方案主要由一个操作台组成，所有功能集中在操作台上面，包括枪支放置平台，显示器，语音播报器，打印机等等。



显示器

语音播报器

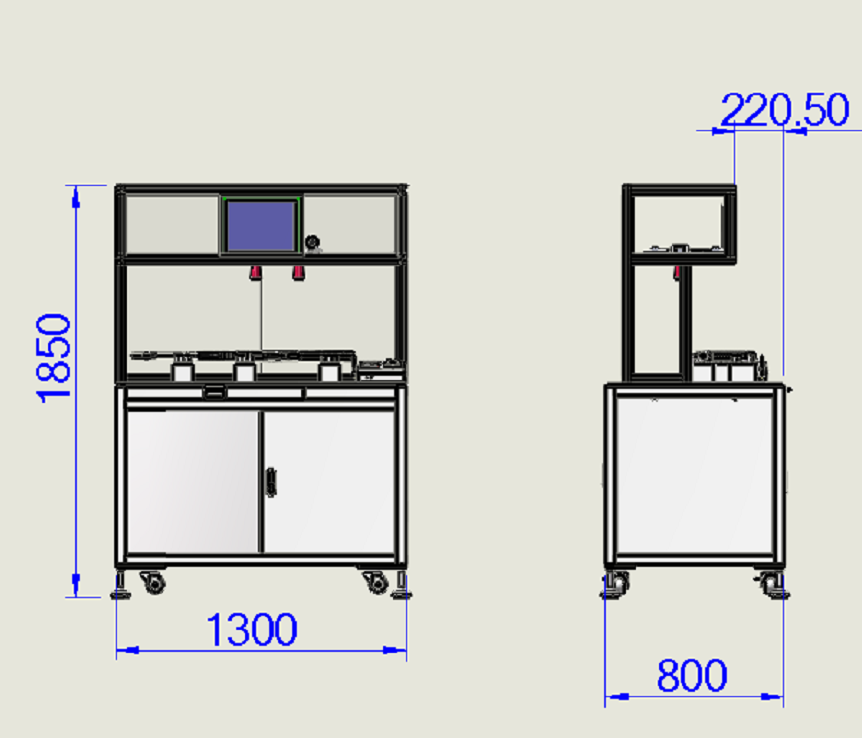
柔性放置平台

待读码枪支

打印机

已读码枪支

2.2.1.2整体设计方案：硬件设计方案



2.2.1.3机台说明：

1. 外形尺寸：长1300mm\*宽800mm\*高1850mm

2. 输入电压：单相220V 50HZ；功率800W

3. 安全要求：配漏电开关

4. 设备机架为工业灰（RAL2006）

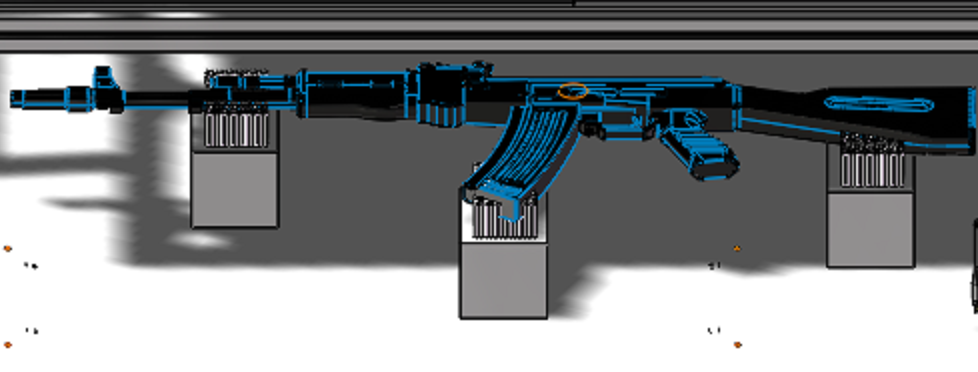
5. 金属部件进行防锈处理，外观光洁、美观。无毛刺、无尖锐突起、无镀层脱落、无锈斑等不良。

6. MTBF平均无故障间隔:>200小时

7. MTTM平均维护时间：小于30分钟

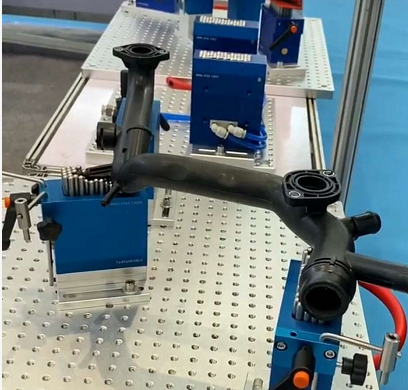
2.2.1.4 详细硬件设计说明

柔性固定方案

说明：

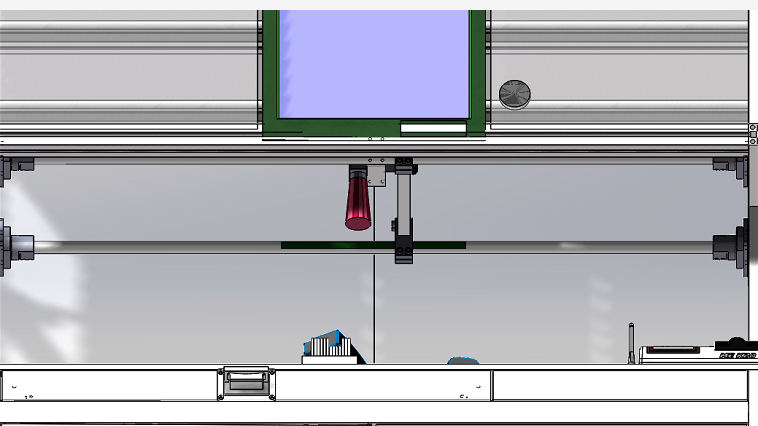
1.底板采用多孔设置，固定座可依据需要进行位置调整。

2.柔性夹具设计，可满足不同形状枪支，不用定制设计夹具。



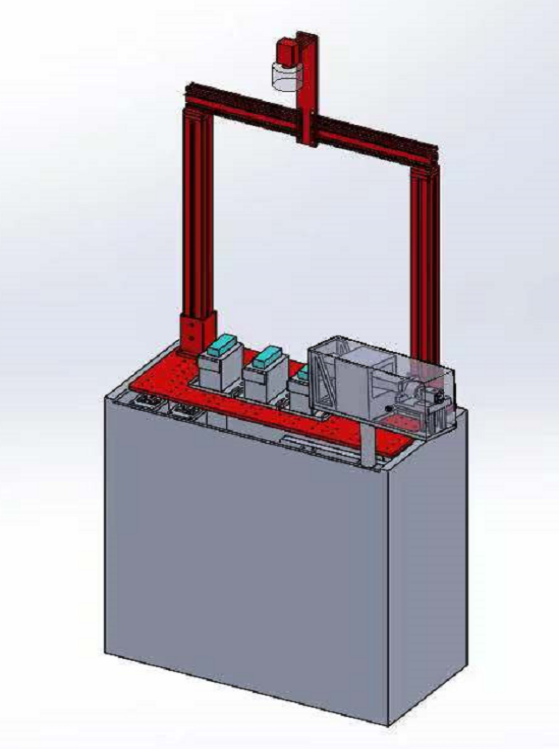
2.2.1.5 光源组成部分说明

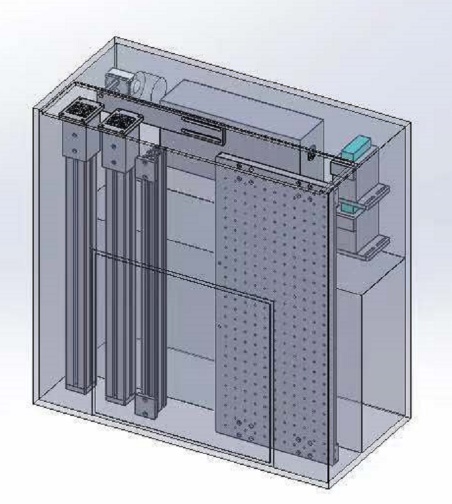
相机/光源设计采用一体式安装方式，手动调整位置：高度和水平位置。



2.2.2 站立式操作硬件设计---简易化+组合化方案设计

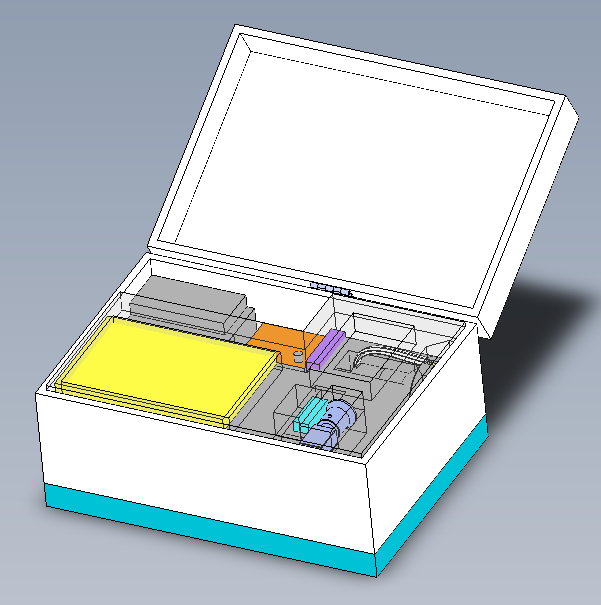
此方案最大的好处是体积小，体积相比第一版缩小60%；另外一点就是组合化安装，通过将不同功能元件的拆卸，所有的机械部件可以全部装在一个箱体内。





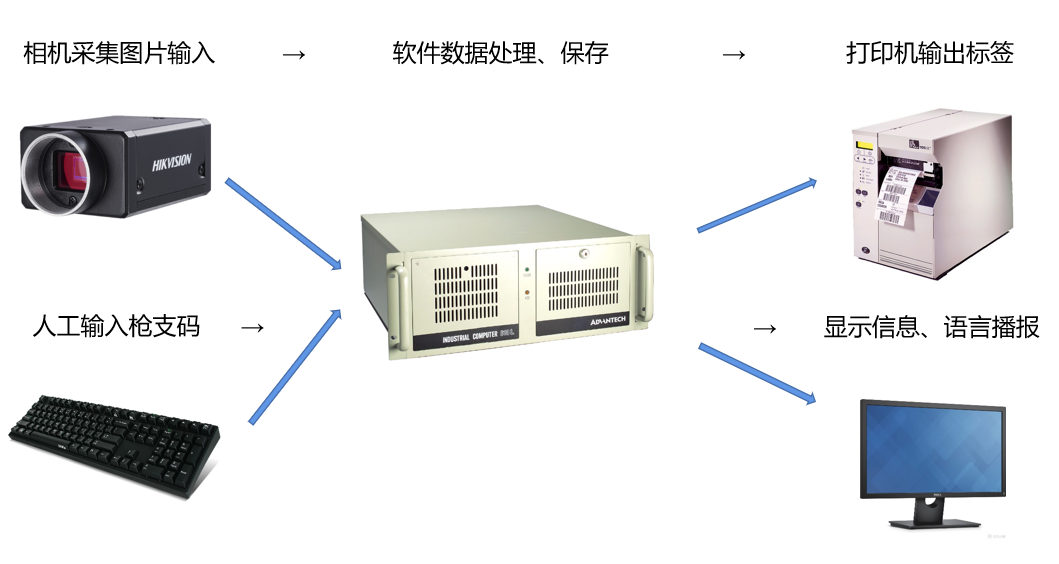
2.2.3 拉箱式操作硬件设计

相比第一版，本次方案的主体结构减少了90%，拉箱将主要的零件整体安装在一体，内容包括：电脑、相机、光源、底座等等。

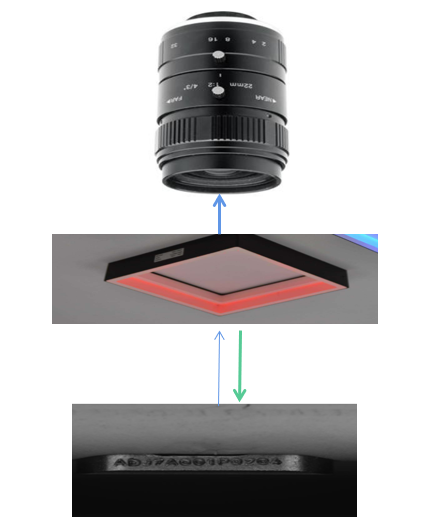


箱体具有高强度，非常方便野外使用。所有的电源目前只使用一个插口，连接起来非常方便。

2.3 视觉部分硬件框架



光学系统：



1、采用工业定制光源，特定形状和波段，有利于金属产品的字符凸显；工业专业设计保证使用寿命

2、采用高清低畸变镜头，保证产品的成像效果

3、采用大品牌相机，优秀的电子电路设计低噪声

4、世界优秀厂家感光sensor

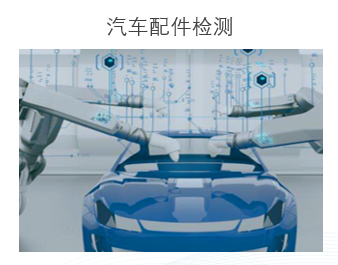
3、算法研究

3.1算法说明：

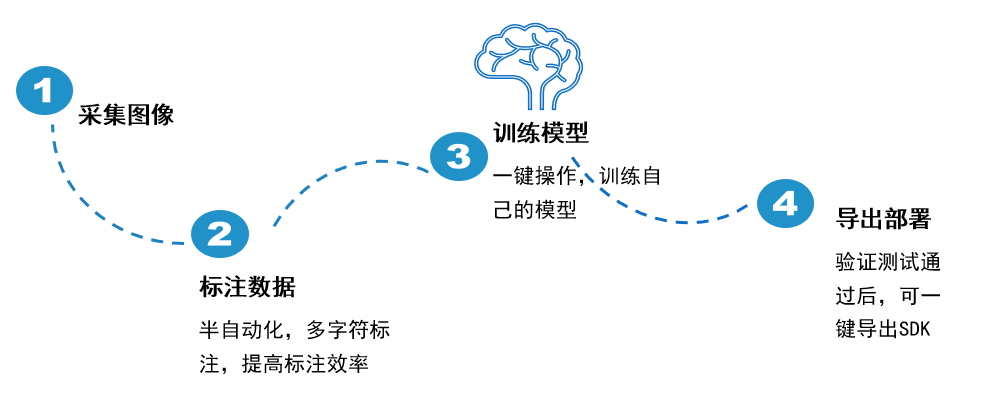
本次项目的算法主要是基于AI的OCR算法，该算法应用场景丰富，存在日常生活的方方面面







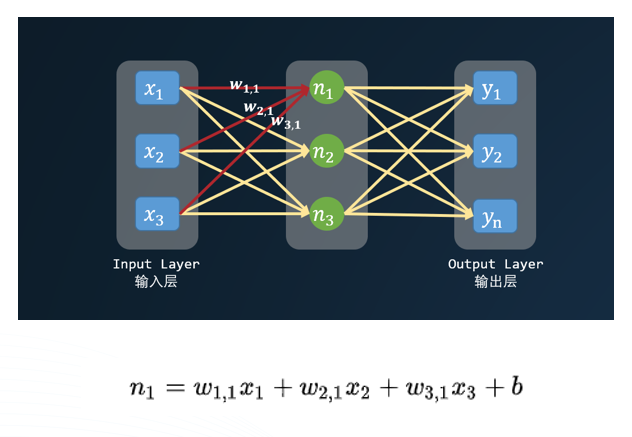
3.2 AI-OCR应用流程



* 采集图像就是获取包含我们需要识别完整字符串的图像
* 标注数据是深度学习算法特有的步骤，就是需要人工标注出字符所在的位置以及字符内容（标签信息）。
* 训练模型就是用我们之前标注的数据包含一一对应的图片和标签信息来训练模型，该模型可以用来识别同类型的其他新的图像数据。
* 导出部署是指当我们的模型有很高的识别率后，可以将该模型通过SDK集成软件系统中。

3.3 AI-OCR算法原理

根据我们采集的几百张带有字符图片及人工标签的标注信息由算法工程师去调试网络的各种参数(不用的检测需求要用不同的网络，调整的参数也不尽相同)，从而生成模型。其本质就是不断地改变网络权重（w1，w2，w3）的大小，减小神经网络输出的误差值。 当误差值较小时够得到一个多层神经网络的模型，即可对未知输入进行有效的预测。



3.4 AI-OCR算法检测过程

整个字符识别的算法过程分为两步：

* 字符定位：首先通过基于深度学习的文本检测算法，定位出字符串所在区域。
* 字符识别：将定位的字符区域输入到字符识别网络，进行字符识别。



经定制算法后可实现多种产品多种字体的兼容

深度学习原理介绍

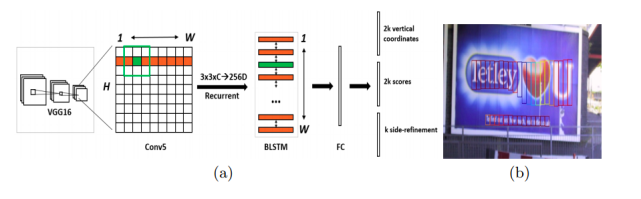
目前，基于深度学习的场景文字识别主要包含两种方法，第一种是分为文件检测和文字识别两个阶段，第二种则是通过端对端的模型一次性完成文字的检测和识别，我们采用的是一种方式。

1.文字检测：文字检测就是要检测到图片中文字所在的区域，其核心就是要区分文字和背景，比较常见的文本检测算法包含CTPN、TextBoxes、EAST等。

2.文字识别：通过文字检测对图片中的文字区域进行定位后，然后对区域中的每一个字符进行识别，常见的有CNN+softmax,CNN+RNN+attention[6]、CNN+stacked CNN+CTC[7]等。

3.5文本检测-原理简介

文本检测算法的主体网络采用CTPN的文本检测算法，由Faster RCNN改进而来，且结合了CNN与LSTM深度网络，支持任意尺寸的图像输入，并能够直接在卷积层中定位文本行。



CTPN是由检测小尺度文本框、循环连接文本框、文本行边细化三个部分组成，具体实现流程为：

1.使用VGG16网络提取特征，得到conv5\_3的特征图；

2.在所得特征图上使用3\*3滑动窗口进行滑动，得到相应的特征向量；

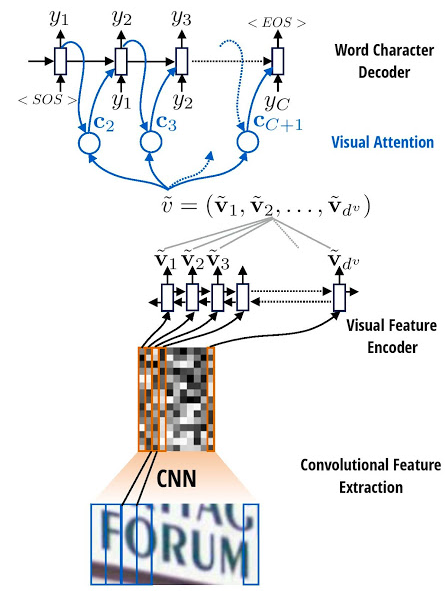
3.将所有特征向量输入BLSTM，学习序列特征，然后连接一个全连接FC层

4.最后结果输出层输出结果

CTPN是基于Anchor的算法，在检测横向分布的文字时能得到较好的效果，而且引入BLST进一步提高了其检测能力。

3.6 文本识别-原理简介

识别网络算法采用的是CNN+RNN+attention[6]的算法本方法是基于视觉注意力的文字识别方法。



其主要过程分为三步：

1.模型首先在输入图片上运行滑动CNN以提取特征

2.将所得特征序列输入到堆叠在CNN顶部的LSTM进行特征序列的编码。

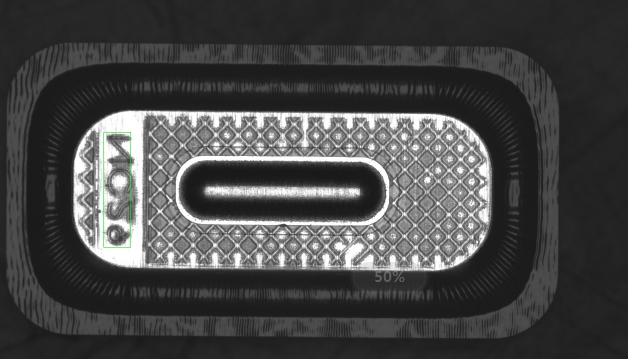
3.使用注意力模型进行解码，并输出标签序列。

3.7 字符定位效果展示

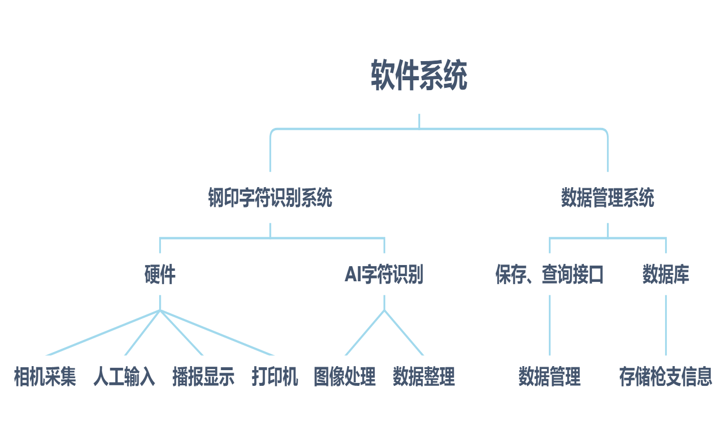


图片左侧是算法自动定位字符位置，右侧是人工标注。

红色区域是算法定位效果。



NC29



4．软件设计

4.1 软件框架设计说明：

4.2 软件层级说明

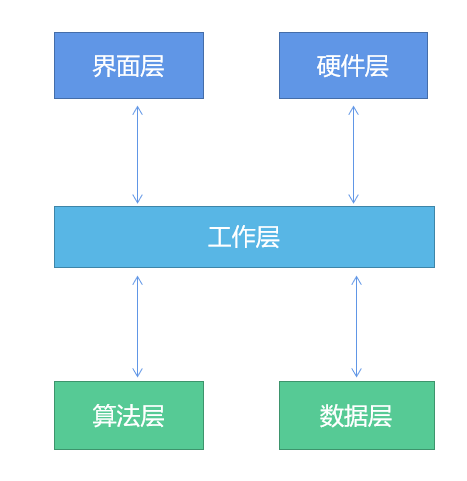
1、界面层：用户界面，负责软件和用户互动实时显示图像与数据信息；

2、硬件层：连接相机和打印机等硬件设备，采集图像和打印标签；

3、工作层：负责整体业务逻辑，信息交互；

4、数据层：负责数据库管理，对数据的保存和查询的功能；

5、算法层：负责字符识别的算法和图像处理



4.3 软件框架设计说明：



操作界面主要功能：

1.实时图片显示。

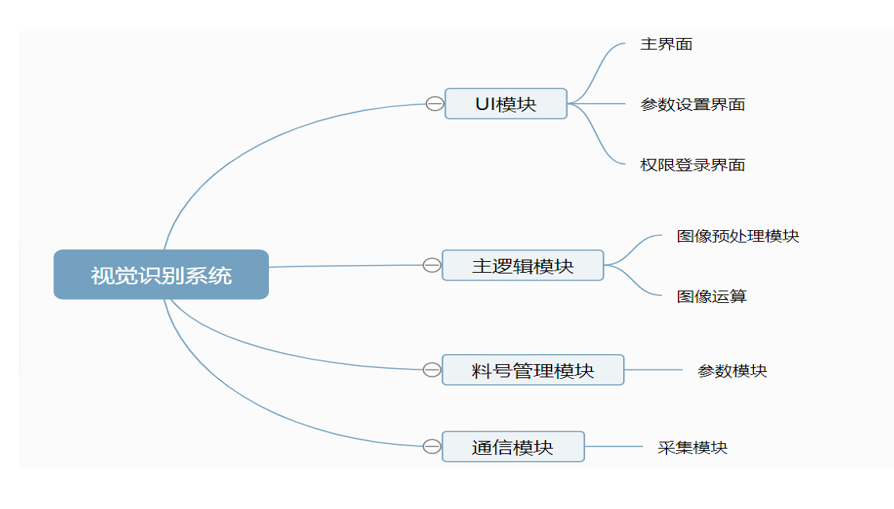
2.数量统计。

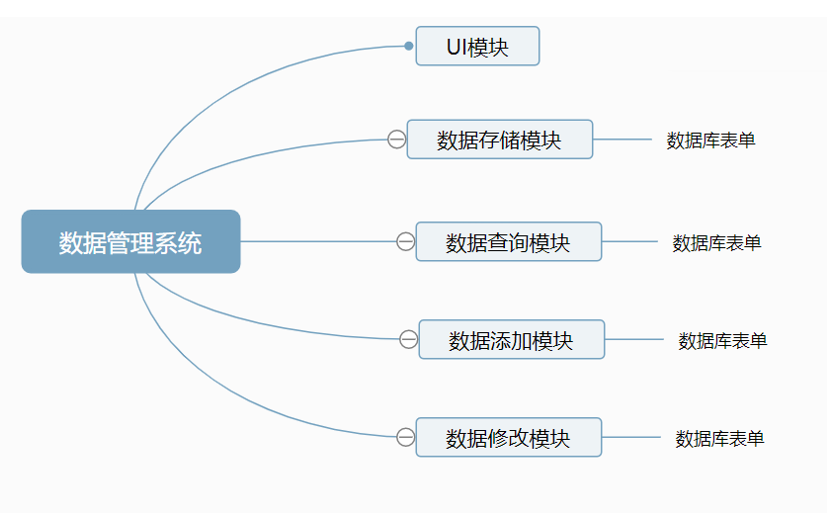
3.实时ID显示。

4.ID查询功能

……

4.4 软件逻辑模块设计说明：





4.5 数据保存及查询功能设计说明：

5.系统实现及试验测试

5.1 软件操作界面说明

5.1.1 **枪支信息智能管理系统——主页**

|  |
| --- |
|  |

5.1.2 账号管理-登录页面

|  |
| --- |
|  |

5.1.3 账号管理页面

|  |
| --- |
|  |

5.1.4 编码检视-登录页面

|  |
| --- |
|  |

5.1.5编码检视-箱装入库管理页面

|  |
| --- |
|  |

5.1.6 编码检视-单装入库管理页面

|  |
| --- |
|  |

5.1.7 检视页面

|  |
| --- |
|  |

5.1.8 信息管理-查询登录页面

|  |
| --- |
|  |

5.1.9 信息管理-信息查询界面

|  |
| --- |
|  |

5.1.10 信息管理-信息出库界面

|  |
| --- |
|  |

5.1.11 信息管理-出库管理页面明

|  |
| --- |
|  |

5.2 文件输出说

|  |
| --- |
|  |

|  |
| --- |
|  |

5.3 设计检出图片说明

|  |
| --- |
| C:/Users/11789/AppData/Local/Temp/picturecompress_20210929002455/output_1.pngoutput_1 |

5.4 实际仿真数据

