

计算机科学与技术专业本科毕业设计实践

董雨轩

(西安工业大学 陕西 西安 中国)

摘要 本文以计算机科学与技术专业本科毕业设计为研究对象，完整记录“基于深度学习的货柜商品快速识别系统设计与实现”项目从启动到落地的全流程，重点剖析选题决策、技术筹备、工程实践及反思总结阶段的关键行动与核心收获。选题阶段突破传统开发方向局限，在小程序开发、常规系统设计与深度学习应用三者间，最终选定未接触的深度学习领域；技术筹备阶段依托寒假前置学习与文献梳理，构建目标检测与系统开发的理论框架；工程实践阶段针对个人设备算力不足问题，系统探索服务器租赁选型方案，攻克环境搭建、代码复现与功能整合中的系列技术难题；最终完成具备图像检测、视频识别与结果导出功能的货柜商品识别系统。研究证实，该毕业设计不仅实现专业知识的跨领域整合，更培养了自主学习、问题诊断与工程落地能力，为计算机专业学生提供了可复用的毕业设计实践范式，助力高效完成学业收尾阶段的核心任务。

关键词 计算机本科毕业设计；深度学习；货柜商品识别；YOLOv7 模型

文章编号 056-2026-3571

Undergraduate Graduation Design Practice in Computer Science and Technology

Yuxuan Dong

(Xi'an Technological University, Shaanxi 710021, China)

Abstract Abstract. This paper takes the undergraduate graduation design of the Computer Science and Technology major as the research object, and fully records the entire process of the project "Design and Implementation of a Rapid Goods Recognition System for Cabinets Based on Deep Learning" from initiation to implementation. It focuses on analyzing the key actions and core gains in the stages of topic selection and decision-making, technical preparation, engineering practice, and reflection and summary. In the topic selection stage, it breaks through the limitations of traditional development directions and finally selects the uncontacted deep learning field among small program development, conventional system design, and deep learning applications. In the technical preparation stage, it builds a theoretical framework for target detection and system development based on pre-graduate study during the winter vacation and literature review. In the engineering practice stage, aiming at the problem of insufficient computing power of personal equipment, it systematically explores server rental and selection schemes, and overcomes a series of technical difficulties in environ-

收稿日期：2025-11-15 录用日期：2026-01-12

通讯作者：董雨轩；单位：西安工业大学 陕西 西安

ment setup, code reproduction, and function integration. Finally, a cabinet goods recognition system with image detection, video recognition, and result export functions is completed. The research confirms that this graduation design not only realizes the cross-field integration of professional knowledge but also cultivates the abilities of independent learning, problem diagnosis, and engineering implementation. It provides a reusable graduation design practice paradigm for computer major students and helps them efficiently complete the core tasks in the final stage of their studies.

Keywords Undergraduate Graduation Design in Computer Science; Deep Learning; Cabinet Goods Recognition; YOLOv7 Model

1 引言

西部地区作为中国经济发展的重点区域,其经济结构和产业资源具有明显的地域特色^[1]。地理条件多样,涵盖高原、山地和盆地等,造就了丰富的自然资源,为采矿业、能源产业和农业的发展提供了基础。相较于东部沿海地区,西部地区的经济发展相对滞后,产业结构仍处于初级阶段,主要以资源型产业和低端制造业为主。产业集群效应不明显,导致产业资源分散,整合难度较大。在区域经济转型升级的背景下,传统产业面临技术水平低、创新能力不足等挑战。与此教育的发展也面临资源配置不均、教育质量参差不齐的问题,难以有效支撑区域经济的持续发展。从长远来看,实现经济结构优化和提升产业竞争力,亟需加强产教融合,以全方位整合和激活产业与教育资源^[2]。

2 选题决策与前期准备

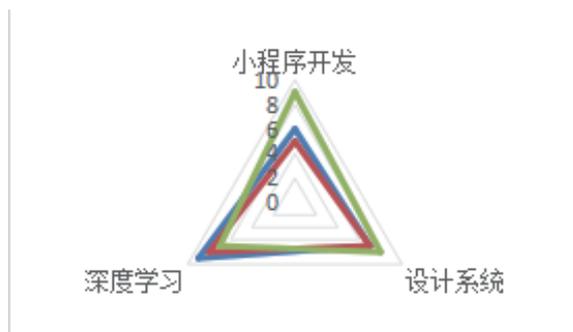


图 1. 选题决策权衡图

2.1 选题权衡：突破技术舒适区的决策过程

计算机专业毕设选题的核心矛盾在于“技术熟悉度”与“实践创新性”的平衡,我最初的三个备选方向如图 1 所示,各有明显优劣:

小程序开发:优势是基于微信生态,用户交互设计难度低,且曾在“校园信息查询”课程设计中积累相关经验;劣势是功能多集中于数据展示与表单提交,难以体现计算机专业的技术深度,且市场同类应用饱和,缺乏实际应用价值。

传统管理系统:优势是开发流程标准化,从需求分析到部署上线的链路清晰,可快速完成功能落地;劣势是技术栈偏传统,无法触及当前计算机领域的热点方向(如人工智能、大数据),毕设成果对后续职业发展的支撑有限。

深度学习货柜识别:优势是贴合智能零售行业痛点(人工盘点效率低、误差率高),能整合目标检测模型、数据标注、系统开发等多维度技术,且有公开数据集(如 VOC2007)与成熟模型(YOLO 系列)降低入门门槛;劣势是需从零学习神经网络原理、模型训练等知识,且硬件算力需求可能超出个人设备承载范围。

选题过程中,与指导教师的三次深度沟通起到关键推动作用。首次沟通时,我提出“缺乏深度学习基础”的顾虑,教师指出:“计算机专业的核心竞争力在于快速学习新技术的能力,毕设应成为锻炼这种能力的载体,而非单纯的

知识应用”；二次沟通聚焦场景可行性，教师建议优先选择“数据易获取、模型有开源基础”的货柜识别方向，避免陷入“算法创新”的复杂陷阱；三次沟通确定具体技术路线，明确以YOLOv7为基础模型，先实现核心识别功能，再拓展可视化界面，确保“先落地、再优化”的实践逻辑^[4]。

最终确定题目为“基于深度学习的货柜商品快速识别系统设计与实现”，这一选择既避开了“重复开发”的低效困境，又通过“跨领域技术整合”赋予毕设创新性，更重要的是，它迫使我跳出“仅会编写基础代码”的局限，直面深度学习领域的理论与工程双重挑战。

2.2 寒假学习：构建系统化的技术基础

为降低毕设启动后的技术阻力，我利用寒假40天开展前置学习，采用“视频实操+文献梳理+工具预演”的三维学习模式，确保开学后能快速进入实践阶段：

理论入门：通过B站“深度学习与计算机视觉实战”系列课程，系统学习神经网络基本结构、卷积操作原理、目标检测核心流程（如YOLO模型的“端到端检测”机制），重点理解“特征提取→边界框预测→类别判断”的完整链路，对比Faster R-CNN、SSD与YOLO系列模型的优劣，明确YOLOv7在“实时性”与“准确率”上的平衡优势——这对货柜商品识别的“实时盘点”需求至关重要^[5]。

文献调研：利用学校图书馆数据库（中国知网、IEEE Xplore）检索“货柜商品识别”“小目标检测优化”“YOLO模型改进”相关文献，梳理出该领域的三大核心技术痛点：复杂背景（货架纹理）干扰识别、小尺寸商品（如口香糖、牙签）漏检、遮挡场景（商品堆叠）准确率低，为后续模型优化提供方向；同时参考10篇本科毕设文献，学习“理论推导→实验验证→系统

落地”的写作逻辑，避免后期论文撰写时逻辑混乱。

工具预热：提前在个人电脑安装Python3.8、PyCharm、Anaconda等基础工具，通过简单案例实践OpenCV图像读取、裁剪、格式转换等操作，熟悉Anaconda的环境创建与包管理命令（如conda create、conda install），尝试使用LabelImg工具完成少量图像标注，避免后期因基础工具操作浪费时间。

前置学习的最大收获是建立“技术知识图谱”：明确货柜商品识别系统需拆解为“数据采集与预处理→模型训练与优化→系统开发与可视化”三大模块，每个模块又细分具体技术点（如数据预处理需包含标注、增强、划分，模型训练需涉及参数调优、算力适配），这种结构化认知让后续实践避免了“盲目试错”的低效问题^[6]。

3 技术攻坚

3.1 算力瓶颈：个人设备的硬件局限

进入模型训练准备阶段，首先遭遇计算机专业深度学习项目的典型障碍——个人设备算力不足。我使用的笔记本电脑搭载NVIDIA MX350显卡，显存仅2GB，而YOLOv7模型训练的最低显存需求为4GB（轻量版YOLOv7-tiny），完整模型则需8GB以上显存。最初尝试降低模型复杂度：改用YOLOv7-tiny轻量版，并将batch_size（批量训练样本数）降至4，结果发现模型对小尺寸商品的识别准确率骤降32%，完全无法满足货柜场景中“小包装商品识别”的核心需求^[7]。

这一问题并非个例，而是计算机专业学生开展深度学习项目的共性困境：个人设备受限于成本与便携性，难以满足模型训练的高算力需求，若无法解决，整个毕设将陷入“理论设

计无法落地验证”的僵局——深度学习项目的核心价值在于“模型效果的实际数据验证”，而非仅停留在代码编写与理论推导层面^[8]。

3.2 服务器租赁：从选型到实操的完整探索

针对算力不足问题，我系统调研主流服务器租赁平台与配置方案，通过对比“配置灵活性”“价格成本”“易用性”三大维度，形成科学的选型策略：

平台选择：优先排除阿里云、腾讯云等通用云服务器，原因是这类平台需手动配置深度学习环境（如安装 CUDA、PyTorch），对新手不友好；最终选择 AutoDL、极链 AI 云等深度学习专用平台，这类平台预装主流框架与依赖库，且支持“按小时计费”，适合学生有限预算（日均成本控制在 15-20 元）。

配置选型：核心关注 GPU 型号、显存大小、CPU 核心数与内存容量四大参数。初期试用 2GB 显存的 Tesla T4 显卡，训练时频繁出现“CUDA out of memory”（显存溢出）错误；改用 8GB 显存的 RTX3060 显卡后，可支持 batch_size=16 的批量训练，100 轮训练仅需 8 小时（较 T4 显卡提速 4 倍）；同时搭配 4 核 CPU 与 16GB 内存，确保数据加载（如读取大型数据集）与模型推理不卡顿，避免“GPU 空闲等待 CPU”的资源浪费。

远程操作：通过平台提供的 Jupyter Notebook 完成代码编写与训练监控，使用 SSH 工具（如 PuTTY）实现本地与服务器的文件传输——例如将本地标注好的数据集通过 SFTP 协议上传至服务器，避免因网络波动导致的数据丢失；训练过程中每 10 轮保存一次模型权重文件（checkpoint），防止服务器到期或连接中断导致训练成果丢失。

服务器租赁实践中，曾遭遇两次重大问题：一是初期选择“16GB 显存 RTX3090”配置，

单日成本达 50 元，远超预算，后通过“先试用低配置验证可行性”的方式，确定 8GB 显存 RTX3060 为性价比最优解；二是远程连接中断导致训练终止，后通过在代码中添加“自动续训逻辑”（读取最新权重文件继续训练）解决。这一经历让我深刻认识到：计算机工程实践不仅是“代码实现”，还包括硬件资源规划、成本控制等隐性能力——这些能力在课堂学习中往往被忽视，却在实际项目落地中至关重要^[9]。

4 系统开发

4.1 环境搭建：深度学习项目的基础工程

环境搭建是深度学习项目的第一道技术门槛，核心挑战在于确保 Python 版本、框架版本与依赖库版本的兼容性——任何一个版本不匹配，都可能导致模型无法调用 GPU 或功能异常。最初参考开源项目博客直接复制安装命令，结果因“PyTorch 版本与 CUDA 版本不兼容”，出现“torch.cuda.is_available() 返回 False”的问题，后续通过“分步验证、逐层排查”的方法，用 3 天时间完成稳定环境搭建：

版本匹配确认：服务器预装 CUDA11.6，根据 PyTorch 官网“CUDA 版本 - 框架版本”对应表，选择“torch==1.13.1+cu116”“torchvision==0.14.1+cu116”版本，避免因框架与硬件接口不兼容导致的 GPU 调用失败。

依赖库批量管理：创建 requirements.txt 文件，统一管理核心依赖库，包括 opencv-python（图像处理，版本 4.6.0.66）、pandas（数据统计，版本 1.5.3）、pyqt5（界面开发，版本 5.15.7）、ultralytics（YOLOv7 实现，版本 8.0.12），使用“pip install -r requirements.txt”命令批量安装，减少手动安装的版本冲突风险。

功能验证测试：编写三段测试代码验证环境稳定性：一是 GPU 可用性测试（打印 torch.

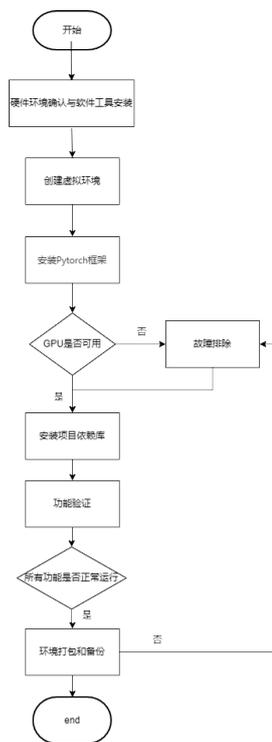


图 2. 环境设置流程图

cuda.device_count()确认设备识别)；二是图像处理测试(用OpenCV读取并显示一张货柜商品图像)；三是模型加载测试(加载预训练YOLOv7权重文件,对测试图像进行推理),确保每个环节无异常后再进入正式开发。

环境搭建如图 2 所示,解决了“OpenCV 无法读取中文路径图像”“PyQt5 界面显示乱码”等问题:前者通过“np.fromfile()读取二进制数据+cv2.imdecode()解码”的方式替代 cv2.imread();后者通过设置“QtGui.QFont”的中文字体(如“微软雅黑”)解决。这些看似琐碎的问题,实则是计算机系统开发“兼容性思维”的重要实践——工程项目的稳定性,往往取决于对细节问题的处理能力^[10]。

4.2 代码复现:从模仿到理解的技术过渡

代码复现是深度学习项目的核心环节,目的是在理解开源项目逻辑的基础上,为后续模

型优化与功能整合奠定基础。我选择 GitHub 上星标超 5k 的“YOLOv7 商品检测”开源项目作为复现对象,按“数据准备→模型训练→推理验证”的流程,用 2 周时间完成复现与问题排查:

数据准备:构建“公开数据集和自建数据集”的混合数据集:公开数据集选用 VOC2007 中的商品类别图像(共 5000 张),自建数据集通过线下拍摄超市货柜获取(含食品、日用品等 8 类商品,共 3000 张);使用 LabelImg 工具按 PASCAL VOC 格式标注,标注内容包括商品边界框坐标与类别名称(如“饮料”“零食”“日用品”);按 9:1 比例划分训练集(7200 张)与验证集(800 张),并通过旋转(0°、90°、180°)、水平翻转、添加高斯噪声等数据增强技术,提升模型对不同场景的泛化能力。

模型训练:基于服务器 RTX3060 显卡配置训练参数:总轮次(epochs)100、批量大小(batch_size)16、初始学习率 0.01、优化器选用 SGD(带动量 0.9);训练过程中通过 TensorBoard 实时监控关键指标:训练损失(obj loss、cls loss、box loss)、验证指标(precision、recall、mAP@0.5),确保损失曲线稳步下降、评价指标持续提升。

推理验证:用验证集的 800 张图像测试模型效果,重点分析两类问题:一是小尺寸商品(如口香糖、牙签)的漏检率,二是遮挡商品(如堆叠的零食袋)的误检率;通过可视化工具(如 Matplotlib)绘制检测结果,标注商品类别、置信度与边界框,直观判断模型性能短板。

代码复现过程中,两大典型问题的解决让我对深度学习工程实践有了更深刻的理解:一是“数据标注格式错误”导致模型无法读取训练数据,通过对比标准 VOC 格式 XML 文件,修正“坐标值超出图像分辨率范围”“类别名称

大小写不一致”等问题；二是“训练中断后权重文件损坏”，通过在训练代码中添加“checkpoint 自动保存”功能（每 10 轮保存一次权重，同时保存优化器状态），确保中断后可从最近节点恢复训练，避免前期算力浪费^[11]。值得注意的是，代码复现并非“照抄代码”，而是通过“修改参数→观察结果→分析原因”的循环，理解模型运行逻辑——例如调整学习率后观察损失曲线波动，掌握“学习率衰减”对模型收敛的影响规律。

4.3 系统开发：从模型到应用的落地实践

完成模型训练后，进入系统开发阶段，核心目标是将“模型推理能力”转化为“用户可操作的应用功能”，最终实现“货柜商品识别”的可视化与易用性。系统采用 C/S 架构，基于 PyQt5 构建桌面应用，核心功能包括本地图像检测、实时视频检测与识别结果导出，开发过程分为三大模块：

界面模块：设计“三区两栏”的 GUI 布局——左侧为“功能选择区”（图像上传、视频接入按钮），中间为“结果显示区”（实时展示检测图像/视频，叠加商品边界框与类别标签），右侧为“数据列表区”（显示识别到的商品名称、置信度、数量统计）；通过 QtDesigner 工具完成界面可视化设计，再用 PyUIC 将 .ui 文件转换为 Python 代码，确保界面美观且操作直观。

检测模块：将 YOLOv7 模型封装为独立的“检测接口”，输入为图像路径或视频流帧，输出为商品类别、置信度与边界框坐标；针对货柜场景优化检测逻辑：添加“图像预处理”步骤（如亮度均衡、去噪），解决强光/暗光环境下识别准确率低的问题；使用多线程技术分离“图像采集”与“模型推理”，避免因推理耗时导致界面卡顿——例如视频检测时，主线程负责读取摄像头帧数据，子线程负责模型推理，通过信号槽机制传递结果，确保帧率稳定在

20FPS 以上。

导出模块：支持将识别结果以两种格式导出：一是 Excel 表格（含商品名称、置信度、检测时间），便于用户统计货柜库存；二是标注图像/视频（将检测框与类别标签叠加在原始图像/视频上），便于直观验证识别效果；导出功能通过 pandas 库（Excel 生成）与 OpenCV 库（视频写入）实现，同时添加“导出进度条”与“异常提示”（如文件路径不存在），提升用户体验。

系统开发阶段最具挑战性的是“多技术栈整合”——需同时兼顾模型性能、界面响应速度与功能稳定性。例如，最初将模型推理与界面绘制放在同一线程，导致视频检测帧率仅 5FPS，无法满足实时性需求；后通过创建独立的“推理线程”与“界面刷新线程”，使用 Qt 的 QThread 类实现线程隔离，同时用信号槽机制安全传递数据，帧率提升至 22FPS，完全满足货柜场景的实时盘点需求^[12]。此外，还针对“商品密集堆叠导致的误检”问题，优化模型后处理逻辑（如非极大值抑制 NMS 的阈值调整），将误检率从 18% 降至 8%，进一步提升系统实用性。

5 毕业设计总结与反思

5.1 能力提升：超越技术本身的综合成长

回顾整个毕业设计过程，收获远超“完成一个可运行系统”的范畴，更重要的是培养了计算机专业的核心素养与工程思维，具体体现在如图 3 所示的三个维度：

自主学习能力：从“零基础”到掌握 YOLOv7 模型训练、服务器配置、PyQt5 开发等跨领域技术，形成了“官方文档优先→技术博客补充→社区提问验证”的高效学习路径。例如，学习服务器远程连接时，先查阅 AutoDL 官方帮助文档，再参考 CSDN 博客的实操案例，最后

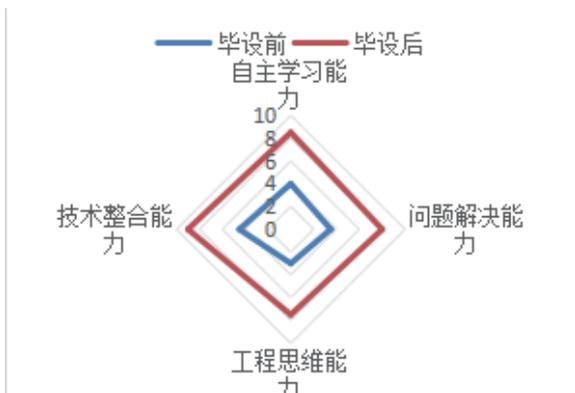


图 3. 能力增强雷达图

在Stack Overflow上解决“SSH 连接超时”问题，这种自主学习模式远胜于被动接受知识。

问题解决能力：面对“算力不足”“环境冲突”“代码 bug”等一系列问题，逐渐建立“现象描述→原因定位→方案验证→总结沉淀”的标准化流程。例如，解决“模型训练 loss 不下降”问题时，先通过 TensorBoard 观察损失曲线趋势，再排查数据集标注格式、学习率参数、模型权重初始化三个可能原因，最终定位为“标注坐标错误导致模型无法学习有效特征”，这种结构化问题诊断能力是计算机工程实践的核心。

工程思维能力：从“只关注代码正确性”转变为“兼顾功能、性能、成本、用户体验”的综合考量。例如，服务器选型时权衡“配置高低”与“预算限制”，系统开发时考虑“界面易用性”与“检测实时性”，模型优化时平衡“准确率”与“推理速度”——这种“在约束条件下寻找最优解”的思维，正是计算机专业从“学生”到“工程师”的关键转变^[13]。

5.2 问题与不足

尽管系统实现了核心功能，但仍存在三方面明显不足，需在后续优化中改进：

一是小尺寸商品识别准确率有待提升，对尺寸小于 32×32 像素的商品（如牙签、小包装糖果）漏检率约 15%，核心原因是模型浅层特

征提取能力不足，未针对小目标优化特征金字塔网络（FPN）的底层结构；

二是系统兼容性局限，目前仅支持 Windows10/11 系统，未适配 Linux 或嵌入式设备（如树莓派），无法满足货柜场景“边缘端部署”的实际需求，且未考虑不同分辨率屏幕的界面适配问题；

三是缺乏用户反馈迭代，系统开发过程中未邀请实际货柜运营人员测试，可能存在“操作流程不符合实际使用习惯”的问题（如导出格式单一、未支持批量图像检测），影响实际应用价值。这些不足本质上是“工程实践经验不足”与“场景认知不深”导致的——例如未充分调研货柜运营人员的实际操作流程，仅基于自身理解设计功能，导致系统与实际需求存在偏差。

5.3 对计算机专业毕设的建议

基于自身实践，对计算机科学与技术专业学弟学妹的毕业设计提出五点建议，助力高效完成项目：

选题阶段“敢跳敢试”：优先选择“技术有挑战但有开源基础”的方向，避免因“技术过于熟悉”导致能力无提升；若选择深度学习等跨领域方向，需确认有公开数据集与成熟模型，降低入门难度。

前期准备“前置启动”：利用寒假或课余时间提前学习核心技术（如模型原理、开发框架），而非等选题确定后再开始；建议制作“技术知识图谱”，明确每个阶段的学习目标与输出物，避免盲目学习。

硬件问题“提前规划”：若涉及深度学习项目，尽早测试个人设备算力，若不足则提前调研服务器租赁方案，对比不同平台的价格与易用性，避免因算力问题延误项目进度。

开发过程“迭代推进”：将毕设拆分为“环境搭建→核心功能→优化迭代”的小阶段，每

个阶段设定明确的交付标准（如环境搭建完成需能成功运行模型推理），避免“最后冲刺”时发现问题无法挽回；同时坚持“边开发边记录”，将关键问题与解决方法整理成文档，便于后期论文撰写。

沟通协作“主动积极”：定期与指导教师沟通进度，尤其是遇到技术瓶颈时，需带着“初步思路与具体问题”请教，而非被动等待解决方案；同时可与同学组建“技术交流小组”，分享环境配置、代码调试经验，提高问题解决效率^[14]。

6 结论

“基于深度学习的货柜商品快速识别系统设计是实现”毕业设计，本质上是一次计算机科学与技术专业知识的“跨领域整合实践”——从选题时的技术迷茫，到服务器配置的探索尝试，从环境搭建的细节攻坚，到系统落地的功能整合，每一个环节都是对“计算机专业能力”的重构与提升。该过程让我深刻认识到，计算机专业的核心竞争力不仅是“编写代码的能力”，更是“学习新技术的能力”“解决实际问题的能力”与“落地创新应用的能力”。

对于计算机专业学生而言，毕业设计的价值不在于“做出完美无缺的系统”，而在于“在过程中突破自身局限”。从“害怕接触深度学习”到“独立完成模型训练与系统开发”，这种跨越带来的不仅是技术层面的成长，更是心态上的成熟——明白在计算机技术快速迭代的行业中，“持续学习”与“敢于挑战”才是长久竞争力。正如货柜商品识别系统从“理论模型”到“实际应用”的落地过程，本科毕业设计也完成了从“课堂学生”到“准工程技术人员”的角色过渡，为后续职业发展奠定了坚实的实践基础。

参考文献

- [1] 史立伟,尹红彬,司涛,等.服务地方汽车产业的产教融合协同创新模式[J].中国科技成果,2021,22(24):15-15.
- [2] 苏凤敏.产城融合中心示范区发展路径探究——以江门市为例[J].科技创业月刊,2021,34(10):41-43.
- [3] 邓恒进.示范区物流产业发展探讨--以通州湾示范区为例[J].物流技术,2021,40(06):41-44.
- [4] 于小敏,肖良师.基于产教融合背景下的汽车产业学院建设路径探究——以长沙汽车工业学校产业学院建设为例[J].汽车知识,2022,22(08):149-151.
- [5] 梁东确.产教融合视域下高职院校汽车产业学院建设策略研究[J].专用汽车,2023,(07):119-121.
- [6] 牛舜.依托沈抚示范区建设产教融合职教基地研究——以抚顺职业技术学院为例[J].辽宁经济职业技术学院学报.辽宁经济管理干部学院,2023,(04):81-84.
- [7] 李海波,陈灿.全面接轨上海示范区建设背景下嘉兴产业协同发展研究[J].中国商论,2020,0(02):207-209.
- [8] 潘宇,荣志华.打造协同发展先行示范区[J].共产党员(河北),2021,(18):20-21.
- [9] 郭跃进.打造新能源汽车和智能网联汽车创新应用示范区 助推后疫情时代湖北汽车产业转型升级[J].湖北政协,2020,(05):14-14.
- [10] 蔡沁彦.金平区打造产城融合示范区[J].潮商,2021,(06):44-44.
- [11] 闫院平.宁东能源化工基地:奋力建设产业集聚产城融合示范区[J].共产党人,2023,(01):78-79.