CAM2CART: VISION BASED PRODUCT SEARCH USING ESP32-CAM AND RASPBERRY PI

Project report submitted in fulfilment of the requirements for the degree of

Bachelor of Technology

in

Electronics and Communication Engineering

by

NAGIREDDY SAI KRISHNA - S200065

NAKKINA CHARISHMA PRIYA - S200354

SATTI NAGA SATYANARAYANA REDDY - S200542

VENNAPUSA DIVYANJALI - S200765

JUJJURI PRAVEEN KUMAR - S200928

Under the Guidance of

Mr T.S.Gagandeep, M.Tech (Ph.D),UGC-NET

Assistant Professor

Department of Electronics and Communication Engineering

Rajiv Gandhi University of Knowledge Technologies - Srikakulam

Certificate

This is to certify that the work entitled **"Cam2Cart: Vision-Based Product Search using ESP32-CAM and Raspberry Pi"** is a bonafide record of authentic work carried out by **NAGIREDDY SAI KRISHNA (S200065),NAKKINA CHARISHMA PRIYA (S200354),SATTI NAGA SATYANARAYANA REDDY (S200542),VENNAPUSA DIVYANJALI (S200765),JUJJURI PRAVEEN KUMAR (S200928)** under my supervision and guidance for the fulfilment of the requirement of the award of the degree of Bachelor of Technology in the department of Electronics and Communication Engineering at Rajiv Gandhi University of Knowledge Technologies - Srikakulam.

The results embodied in this work have not been submitted to any other university or institute for the award of any degree or diploma. This thesis, in our opinion, is worthy of consideration for the award of the degree of Bachelor of Technology in accordance with the regulations of the institute.

Mr T.S.Gagandeep
Assistant Professor,
Department of ECE,
RGUKT - Srikakulam.

Mr T.S.Gagandeep
Head of the Department,
Department of ECE,
RGUKT - Srikakulam.

Declaration

We hereby declare that the dissertation entitled **Cam2Cart: Vision-Based Product Search using ESP32-CAM and Raspberry Pi** submitted for the Bachelor of Technology Degree is our original work, and the dissertation has not formed the basis for the award of any degree, associateship, fellowship, or any other similar titles.

Place: Srikakulam
Date: November 18, 2025

NAGIREDDY SAI KRISHNA - **S200065**
NAKKINA CHARISHMA PRIYA - **S200354**
SATTI NAGA SATYANARAYANA REDDY - **S200542**
VENNAPUSA DIVYANJALI - **S200765**
JUJJURI PRAVEEN KUMAR - **S200928**

Acknowledgement

I would like to express my special thanks of gratitude to my project guide **Mr T.S.Gagandeep** sir, Assistant Professor, Department of Electronics and Communication Engineering , who gave me the golden opportunity to do this wonderful project on the topic **“Cam2Cart: Vision-Based Product Search using ESP32-CAM and Raspberry Pi”**, which also helped me in doing a lot of research and I came to know about so many new things I am really thankful to them.

Department of Electronics and Communication Engineering

Rajiv Gandhi University of Knowledge Technologies - Srikakulam

Abstract

The integration of artificial intelligence and embedded systems has enabled the creation of intelligent, automated solutions that can perform complex tasks efficiently. Cam2Cart: Vision-Based Product Search Using ESP32-CAM and Raspberry Pi is a novel system designed to identify and retrieve information about physical products using real-time image processing and computer vision. The project focuses on developing a low-cost and scalable model that can recognize products through camera input, process visual data, and display relevant information to users instantly. This concept aims to modernize retail environments by replacing conventional barcode or QR-based identification systems with vision-based automation.

In the proposed system, the ESP32-CAM module serves as the primary image acquisition unit, capturing product images in real time. These images are transmitted to the Raspberry Pi, which functions as the central processing and decision-making hub. The Raspberry Pi uses OpenCV, PYtesseract OCR,Numpy,QRcode Request,Python imaging libraries to perform image classification and object detection. Once the product is recognized, details such as name, price, and specifications are fetched from a predefined database or server. The processed output is displayed on a graphical interface or can be accessed through a mobile application, providing a seamless product search experience to users.

One of the major advantages of Cam2Cart lies in its cost-effective and efficient hardware-software integration. The ESP32-CAM, with built-in Wi-Fi and compact camera capabilities, enables real-time image transmission with minimal power consumption. The Raspberry Pi's computational capabilities make it ideal for running lightweight machine learning models locally without requiring cloud computing. This design ensures faster response times, increased reliability, and data privacy. Furthermore, the system's modular architecture allows for easy expansion and customization to support new features such as gesture-based interaction, product recommendation, or inventory tracking.

In summary, this project demonstrates how embedded systems, IoT connectivity, and computer vision can merge to build a practical and intelligent solution that bridges the gap between the physical and digital shopping experience.

Table of Contents

List of Figures

1. Introduction

In the modern world, the rapid evolution of technology has transformed how people interact with retail and e-commerce systems. Traditional shopping methods often require manual searching, barcode scanning, or entering product names into databases. These approaches, while functional, are time-consuming and lack automation. To address this challenge, vision-based systems that leverage artificial intelligence and embedded devices are becoming increasingly popular. The proposed project, Cam2Cart: Vision-Based Product Search Using ESP32-CAM and Raspberry Pi, aims to simplify the product identification process by using computer vision and image recognition. This system enables users to identify products through visual input captured by a camera, making shopping and inventory processes faster, more efficient, and more intelligent.

The Cam2Cart system is designed around two key hardware components—ESP32-CAM and Raspberry Pi. The ESP32-CAM acts as an image-capturing device that takes real-time pictures of the products placed in front of it.The Raspberry Pi performs feature extraction, object detection, and classification to identify the product and retrieve related details such as name, price, and description. This information is then displayed to the user through a connected interface. By integrating machine learning and embedded systems, Cam2Cart bridges the gap between physical products and digital databases.

The main objective of this project is to develop a low-cost, efficient, and intelligent system that eliminates the need for manual product searches or barcode-based identification. It emphasizes automation, user convenience, and real-time processing. Unlike barcode scanners, which require physical tags, Cam2Cart relies on the visual features of objects, making it more adaptable for diverse product categories. Additionally, the system demonstrates how affordable components like ESP32-CAM and Raspberry Pi can be used to implement advanced vision-based tasks without depending on expensive computing resources. This makes the project highly suitable for small-scale retail environments, educational purposes, and prototype development for smart retail systems.

1.1. Background and Motivation

In recent years, the demand for automation and intelligent systems has grown rapidly across multiple domains, especially in retail and e-commerce. Traditional retail systems rely heavily on manual operations such as barcode or QR code scanning, which require human intervention and physical tagging of products. Although effective, these systems are limited in scalability and convenience. With the advent of artificial intelligence (AI) and computer vision, it has become possible to automate product identification by analyzing visual data directly, thereby eliminating the need for manual scanning or labeling. This evolution has paved the way for vision-based product recognition systems that can intelligently identify items using real-time image processing and pattern recognition techniques.

The Cam2Cart project emerges from the idea of simplifying product identification using low-cost embedded systems. In many small-scale retail environments, deploying high-end computer vision systems or cloud-based image recognition solutions is not economically feasible. However, with the introduction of powerful yet affordable microcontrollers and single-board computers such as the ESP32-CAM and Raspberry Pi, it is now possible to implement intelligent vision systems locally. The ESP32-CAM offers built-in Wi-Fi and camera functionality, while the Raspberry Pi provides enough computational capability to perform machine learning and image processing tasks efficiently. Combining these two devices forms an effective, compact, and cost-efficient solution for vision-based product search.

The motivation behind developing Cam2Cart lies in addressing the limitations of conventional retail search systems. Current barcode-based systems are not adaptable to untagged or unregistered items, and manual searches are inefficient, particularly in environments with large product catalogs. A vision-based solution offers greater flexibility, as it can recognize products by their appearance rather than relying on predefined codes. This capability not only enhances customer convenience but also reduces dependency on database maintenance and physical labeling. Furthermore, with real-time image recognition, users can simply show a product to the camera and instantly retrieve detailed information, mimicking a smart, human-like assistant in the retail process.

1.2. Problem Defination

In the modern retail environment, efficient product identification and management are essential for improving customer experience and operational productivity. Traditional retail systems still rely on barcode or QR code scanning methods, which, although effective, require physical labels, manual scanning, and constant maintenance. These methods are often prone to human error, slower in operation, and not suitable for untagged or visually similar products. In large supermarkets or inventory-based environments, such limitations lead to delays, misidentification, and reduced efficiency. As a result, there is a growing need for a more automated and intelligent system that can recognize and retrieve product details based purely on visual appearance.

The core problem addressed in this project is the absence of an affordable, vision-based product search mechanism that can perform real-time product recognition using camera input instead of manual scanning. Existing solutions are either costly or dependent on cloud-based machine learning systems, which are not ideal for small-scale or offline operations. The goal of this project, Cam2Cart, is to design and implement a low-cost embedded system capable of visually recognizing products and displaying relevant information instantly. The system must work with minimal hardware resources while maintaining accuracy and speed in product identification. This makes it highly suitable for small businesses, retail stores, and prototype smart cart applications.

1.2.1. Challenges in Existing Systems

Despite advancements in technology, current retail systems face several limitations that restrict automation and real-time adaptability. Barcode-based systems require that every product be tagged and properly aligned for scanning, which is inefficient in fast-paced retail environments. Damaged, misplaced, or missing barcodes can lead to identification failures. Additionally, manual searching for product details through user interfaces consumes time and increases workload for both customers and employees. These limitations highlight the need for a smarter, vision-based approach that can identify products directly through image capture.

Another challenge is the cost and complexity of existing vision-based recognition

systems. Most AI-driven retail solutions rely on cloud computing or expensive high-performance hardware, which small retailers cannot afford. These solutions also depend on constant internet connectivity, making them unsuitable for local or offline operations. By using low-cost embedded devices like ESP32-CAM and Raspberry Pi, the Cam2Cart system aims to overcome these limitations by enabling local processing, real-time recognition, and seamless communication between hardware modules. This approach minimizes cost, reduces latency, and allows for scalability and customization according to user needs.

In conclusion, the Cam2Cart project defines the problem as the need for a real-time, low-cost, and efficient vision-based product identification system that reduces human effort and increases automation in retail settings. By leveraging embedded vision and machine learning, the proposed system provides a scalable solution to the long-standing challenges of product search and inventory management. It represents a step toward smart retail automation where convenience, speed, and intelligence converge into a single integrated platform.

1.3. Objectives of the Projects

The main objective of the Cam2Cart project is to design and implement an intelligent, low-cost, and vision-based product search system using embedded platforms such as ESP32-CAM and Raspberry Pi. The project aims to bridge the gap between physical products and digital databases by utilizing real-time image processing and machine learning techniques. Unlike traditional systems that depend on barcode or QR code scanning, Cam2Cart identifies products purely based on their visual features. This approach enhances user convenience, operational efficiency, and the overall shopping experience, making it suitable for modern smart retail applications.

The proposed system uses the ESP32-CAM as a real-time image capture device that transmits product images to the Raspberry Pi, which serves as the main processing unit. The Raspberry Pi runs machine learning algorithms and computer vision models using OpenCV,Pytesseract to analyze and classify the product. The recognized product information—such as name, price, and description—is then retrieved from a database and displayed on an interface or mobile device. Through this automated pipeline, the project seeks to eliminate manual input, reduce human dependency, and increase the accuracy of product identification.

Specific Objectives

To accomplish the above goal, the project is designed with the following specific objectives:

1. To develop a vision-based product recognition system that can accurately identify products using images captured by the ESP32-CAM.
2. To implement efficient image processing and machine learning algorithms on the Raspberry Pi for real-time classification and feature extraction.
3. To design a communication interface between ESP32-CAM and Raspberry Pi using Wi-Fi for fast and reliable image transmission.
4. To create a product information database containing item details such as product name, price, and category for retrieval upon recognition.
5. To develop a user-friendly interface that displays product information in an organized and intuitive manner.
6. To evaluate system performance in terms of recognition accuracy, processing time, and hardware efficiency.
7. To ensure scalability and adaptability so that the system can be extended for various applications like smart carts, inventory tracking, and automated billing systems.

Expected Outcomes

Upon successful completion, the Cam2Cart system is expected to provide an automated and efficient product identification platform capable of real-time operation with minimal hardware cost. The system will demonstrate how embedded computing and AI can work together to solve practical retail problems. It will serve as a prototype for future smart retail and IoT-based applications by showing that even low-power devices can perform intelligent visual recognition tasks. Moreover, the project will encourage research and development in the field of embedded vision, IoT-based retail automation, and real-time object recognition, paving the way for smarter and more convenient shopping solutions.

1.4. Significance of the Study

The Cam2Cart project demonstrates the growing potential of embedded vision and artificial intelligence in automating everyday processes. Its significance lies in the ability to perform intelligent product identification using low-cost, accessible hardware such as the ESP32-CAM and Raspberry Pi. By replacing traditional barcode-based systems with vision-based recognition, the project introduces a smarter, more flexible method for retail automation. This approach reduces dependency on physical tags and manual operations, improving both accuracy and user convenience.

Furthermore, the project highlights how embedded systems, IoT connectivity, and computer vision can be effectively integrated into a single compact solution. The Cam2Cart system is not only cost-effective but also energy-efficient, making it suitable for small businesses and research-based implementations. It provides a foundation for future advancements in smart retail technologies, automated billing, and inventory management systems. Beyond its technical outcomes, the project contributes to learning and experimentation in embedded AI and real-time image processing.

1.4.1. Practical Importance

The practical importance of the Cam2Cart system lies in its real-world applicability across multiple domains. In retail, it can assist customers and store managers by automatically identifying and displaying product details. In academic settings, it serves as a model for understanding embedded computer vision and IoT communication. Its scalable design allows for future expansion, such as adding cloud support or integrating with mobile apps. Overall, this project provides both educational and commercial value by showcasing how affordable intelligent systems can transform traditional retail operations.

1.5. Scope and Limitations

The Cam2Cart project focuses on developing a vision-based product search and recognition system that utilizes embedded platforms such as the ESP32-CAM and Raspberry Pi to automate the process of product identification. The system aims to enhance automation in retail and inventory environments by replacing traditional bar-

code and manual search methods with an intelligent, image-based recognition approach. It demonstrates how computer vision and IoT can be combined to improve efficiency and accuracy in real-world retail systems.

The core scope of the project involves the integration of hardware and software components to achieve a compact, low-cost, and reliable embedded solution. The ESP32-CAM module functions as the image-capturing unit that collects product images in real time. These images are then transmitted wirelessly to the Raspberry Pi, which processes them using computer vision and machine learning algorithms implemented through OpenCV and Pytesseract OCR . The recognized product details—such as name, price, and description—are retrieved from a local or cloud database and displayed on a connected interface. Through this process, Cam2Cart establishes a complete workflow from image acquisition to intelligent product recognition and display.

1.5.1. Applications of the System

The Cam2Cart system has a wide range of practical applications across different sectors. In retail automation, it can be deployed in smart stores to identify products instantly and provide customers with relevant details without scanning barcodes. In smart shopping carts, the system can recognize items as they are placed in the cart, automatically updating a virtual bill. It can also be used in inventory management to track and organize stock in warehouses, reducing manual workload. Beyond commercial applications, Cam2Cart can serve as an educational prototype for demonstrating embedded vision, IoT integration, and artificial intelligence concepts to students and researchers.

The flexibility and scalability of the system allow it to be adapted for future extensions such as voice-assisted search, mobile app integration, cloud-based analytics, and automated checkout mechanisms. These potential enhancements demonstrate the system's capability to evolve into a more sophisticated and commercially viable product. By leveraging open-source tools and affordable hardware, Cam2Cart promotes innovation in smart retail systems while keeping implementation costs minimal.

1.6. Report Organization

This report is organized into a sequence of chapters that describe the development of the Cam2Cart project in a clear and structured manner. The first chapter, Introduction, outlines the background, motivation, objectives, problem definition, and overall scope of the work. It also highlights the significance of the project and provides a brief overview of the report's contents. The second chapter, Literature Review, discusses existing research and technologies related to embedded vision systems, product recognition, and IoT-based automation, identifying limitations in current solutions that inspired the proposed system.

The third chapter, System Design and Methodology, explains the architecture and workflow of the proposed model. It describes the functions of the ESP32-CAM and Raspberry Pi, along with the use of computer vision and machine learning algorithms for image processing and product identification. The fourth chapter, Implementation and Results, presents the practical execution of the design, including hardware integration, software coding, and testing. It also highlights the experimental results and evaluates system performance based on accuracy and response time.

Finally, the fifth chapter, Conclusion and Future Scope, summarizes the project's outcomes and suggests possible improvements such as cloud connectivity, enhanced recognition models, and mobile application support. The overall organization ensures a smooth flow of ideas—from identifying the problem to presenting a functional and efficient solution for vision-based product recognition.

2. Literature Review

The literature review provides an overview of existing research, technologies, and developments related to computer vision, embedded systems, and product identification. It establishes the theoretical foundation for the Cam2Cart project and helps identify the gaps in current approaches that motivate the proposed vision-based product search system.

Over the years, product identification systems have evolved from traditional barcode and RFID technologies to more advanced image-based recognition methods. Barcode systems, while accurate, require physical scanning and depend on printed labels, making them less flexible in dynamic retail environments. Similarly, RFID-based methods provide automated identification but are costly and require specialized readers. As technology has advanced, the focus has shifted toward computer vision and artificial intelligence to recognize objects visually without manual intervention.

In recent years, embedded vision systems have gained popularity due to their compact design, affordability, and ability to perform local processing. Devices such as the Raspberry Pi and ESP32-CAM have made it possible to build intelligent systems capable of image capture, processing, and wireless communication. The Raspberry Pi supports frameworks like OpenCV and Pytesseract OCR , which enable real-time object detection and classification. Research studies have demonstrated that combining these tools allows low-power embedded platforms to perform tasks such as face recognition, object tracking, and product classification efficiently.

Several researchers have worked on developing smart shopping systems using embedded vision. For example, camera-based checkout systems and automated inventory systems have been implemented using neural network models that identify products placed before a camera.

In summary, the review of existing literature indicates a clear research opportunity for a low-cost, vision-based retail automation system that operates locally with high accuracy. The Cam2Cart system builds upon these existing works by integrating image capture through ESP32-CAM and processing via Raspberry Pi to deliver an efficient, intelligent, and accessible solution for modern retail environments.

2.1. Existing Technologies

Over time, various technologies have been developed to support product identification and retail automation. The most commonly adopted methods include barcode scanning, RFID-based identification, and computer vision-based recognition. Each of these systems plays a key role in improving product management and sales efficiency, but each also has limitations that create the need for more intelligent and flexible solutions.

2.1.1. Barcode and RFID Technologies

Barcode systems are among the oldest and most widely used product identification methods. They are simple, reliable, and cost-effective, making them the standard in most retail stores. However, barcode scanning requires direct visibility and manual alignment of the product with the scanner. This dependency on physical tags slows down the billing and inventory process, especially in large retail environments. Additionally, if the barcode is damaged, faded, or missing, the product cannot be identified, resulting in operational inefficiencies.

Radio Frequency Identification (RFID) technology emerged as an improved alternative to barcodes. It enables wireless and contactless identification through electromagnetic signals, allowing multiple items to be scanned simultaneously. RFID tags store unique product information, which can be read by compatible readers without the need for line-of-sight contact. Despite these advantages, RFID remains costly to implement, as tags and readers are more expensive than barcode systems. Furthermore, interference from metals or liquids can affect signal strength, and maintaining large-scale RFID networks is still a challenge for small retailers.

Beyond these traditional methods, computer vision-based identification has gained popularity due to advances in artificial intelligence and embedded computing. Using cameras and deep learning algorithms, vision-based systems can recognize objects by analyzing features such as shape, color, and texture. Unlike barcodes and RFID, these systems do not rely on physical labels, offering greater flexibility and automation potential. With affordable hardware like the ESP32-CAM and Raspberry Pi, and software frameworks such as OpenCV and Pytesseract OCR , vision-based identification has become practical even for small-scale applications.

In conclusion, while barcode and RFID technologies have served as the foundation for product identification, their limitations in flexibility, scalability, and cost highlight the growing need for vision-based recognition systems like Cam2Cart. This evolution represents a major step toward fully automated, intelligent retail environments.

2.2. Comparison Between Existing Methods and Proposed System

Existing product identification methods such as barcode scanning, RFID tagging, and cloud-based computer vision have played vital roles in automating retail operations. However, each technology carries its own set of limitations that restrict efficiency, scalability, and affordability. The Cam2Cart system proposes a more flexible and intelligent approach that integrates embedded vision, IoT communication, and real-time image processing to overcome these issues.

Traditional barcode systems require printed tags and manual scanning, which introduces delays and dependency on physical handling. Though accurate, these systems fail when barcodes are damaged or obscured. RFID-based systems eliminate the need for direct scanning but require costly tags and readers, making them impractical for small or medium-scale retail environments. Cloud-based computer vision approaches can deliver high accuracy through powerful servers, yet they depend heavily on constant internet connectivity and entail privacy concerns due to the transmission of image data to external servers.

In contrast, the proposed Cam2Cart system operates on affordable embedded hardware—the ESP32-CAM for image capture and the Raspberry Pi for local processing. It leverages OpenCV and Pytesseract OCR frameworks to perform object detection and classification directly on-device, reducing latency and eliminating the need for external computation. This makes the system faster, more secure, and accessible for small businesses or educational institutions. Moreover, its modular design allows easy scalability, integration with mobile applications, and future enhancements such as voice assistance or cloud synchronization when required.

2.2.1. Advantages of the Proposed System

The Cam2Cart system introduces several advantages over existing methods. It provides real-time product recognition without the need for barcodes or RFID tags, thus

minimizing manual effort and operational costs. The system's edge-based processing ensures data privacy and reduces dependency on network infrastructure. Its low-cost components make it suitable for widespread adoption, while the open-source nature of its software encourages further development and customization. Additionally, the combination of embedded computing and AI enables adaptability to various retail settings—from small local stores to automated inventory management systems.

In summary, by combining accuracy, cost-efficiency, and independence from external infrastructure, the proposed Cam2Cart framework stands as a practical and scalable alternative to conventional product identification systems, paving the way for intelligent and autonomous retail automation.

2.3. Research Gap

Although several technologies and systems have been proposed for automated product identification, many of them face challenges that limit their practical usability, affordability, and adaptability. Existing methods such as barcode and RFID-based systems, while effective in controlled environments, rely on physical tags and scanning equipment, which restrict automation in dynamic retail settings. Meanwhile, cloud-based computer vision approaches, though highly accurate, depend on powerful remote servers and stable internet connectivity, which introduces latency, security concerns, and additional costs. These limitations highlight the need for a cost-effective, standalone, and vision-based embedded solution that can operate efficiently at the edge without requiring continuous cloud dependency.

Many studies have demonstrated the potential of computer vision in object recognition using deep learning models. However, these implementations often require high-end GPUs or cloud infrastructure, making them unsuitable for low-cost embedded devices. There is limited research that focuses on implementing vision-based recognition systems on compact platforms such as the ESP32-CAM and Raspberry Pi. Additionally, most existing research targets specific use cases like face detection or traffic monitoring rather than retail automation. As a result, there exists a research gap in designing a lightweight yet accurate product identification system optimized for low-power, affordable embedded platforms.

2.3.1. Identified Research Opportunities

The identified research opportunities lie in developing a real-time, embedded vision system that combines efficiency, affordability, and adaptability. The Cam2Cart project addresses this gap by integrating the ESP32-CAM for real-time image capture and the Raspberry Pi for edge-based processing. Using frameworks like OpenCV and Pytesseract OCR , the system performs on-device image recognition and product matching without external cloud resources. This approach ensures faster processing, enhanced privacy, and reduced operational cost. The research opportunity extends to exploring model optimization techniques for embedded devices, enabling high performance even with limited computational power.

In summary, the major research gap lies in the absence of a low-cost, locally processed vision-based retail system that bridges the divide between traditional scanning methods and advanced AI-driven automation. The Cam2Cart project is designed specifically to fill this gap and demonstrate how embedded intelligence can revolutionize modern retail systems.

2.4. Related Work Summary

A review of existing literature reveals significant progress in the field of automated product identification and embedded vision systems. Earlier technologies such as barcode and RFID-based systems provided foundational automation for inventory and retail management. These systems demonstrated high accuracy and ease of use but were limited by the need for physical tags, line-of-sight scanning, and relatively high infrastructure costs. As a result, researchers began exploring computer vision and machine learning approaches that rely on image recognition rather than physical identifiers.

Several studies have shown the potential of camera-based object recognition for smart retail and automation. Research involving deep learning models such as Convolutional Neural Networks (CNNs), YOLO (You Only Look Once), and MobileNet has achieved impressive results in object classification and detection. However, most of these implementations rely on powerful servers or cloud platforms to perform computation, which increases latency and requires constant network access. Some works using Raspberry Pi for real-time recognition have proven feasible but often

struggle with processing limitations when running complex models. Similarly, the ESP32-CAM, while efficient for image capture, requires external support for intensive processing tasks.

The related works collectively highlight the importance of integrating embedded systems with optimized vision algorithms to balance performance and cost. However, the gap remains in developing a self-contained, low-cost, and efficient vision-based product identification system that can perform inference locally on embedded hardware without cloud dependency. This is where the Cam2Cart project introduces innovation — combining the strengths of ESP32-CAM and Raspberry Pi with lightweight frameworks like OpenCV and Pytesseract OCR to create a practical, real-time embedded solution for product search and retail automation.

2.4.1. Key Observations from Related Studies

From the review of existing works, it is evident that most prior systems focus either on accuracy or cost reduction, but very few achieve both simultaneously. The Cam2Cart system bridges this gap by ensuring reliable image-based recognition using affordable and accessible hardware. It also emphasizes real-time operation, scalability, and adaptability — characteristics that are crucial for the future of intelligent retail systems.

2.5. Key Findings from Literature

A comprehensive analysis of existing literature reveals that the integration of computer vision and embedded systems has opened new possibilities for intelligent automation. Traditional identification systems such as barcode and RFID have served as reliable tools for retail operations but lack the scalability and automation required in modern contexts. Vision-based technologies, supported by deep learning and image processing, have demonstrated the ability to recognize products and objects with high accuracy and flexibility. These methods eliminate the need for manual scanning or tagging, making them more suitable for dynamic retail environments where visual features can uniquely identify products.

Despite the remarkable progress in this field, there is still a need for optimization when deploying vision-based algorithms on low-power embedded devices. Many studies rely on cloud-based solutions or high-end processors to perform real-time

image classification, which increases cost and dependency on internet connectivity. Research also shows that while Raspberry Pi and ESP32-CAM are effective for small-scale implementations, there is limited exploration of how these devices can work collaboratively to achieve robust, real-time recognition performance. Thus, existing works suggest that combining embedded computing power with efficient lightweight neural networks could lead to the development of scalable and cost-efficient retail automation systems.

3. System Analysis

System analysis is a crucial stage in the development of any engineering or technological solution, as it provides a comprehensive understanding of the problem domain, user requirements, and the technical feasibility of the proposed system. In the context of Cam2Cart: Vision-Based Product Search Using ESP32-CAM and Raspberry Pi, system analysis focuses on evaluating how embedded vision and machine learning techniques can be combined to automate product identification in retail environments. This process helps identify system limitations, determine hardware and software requirements, and define functional objectives to ensure optimal system performance.

The primary objective of system analysis is to bridge the gap between conceptual design and practical implementation. It enables the identification of various components, such as image capture, data transmission, and product recognition, and assesses how these elements interact to achieve the overall system goals. By analyzing existing systems and understanding user expectations, the Cam2Cart project aims to design a solution that improves operational efficiency, minimizes manual dependency, and provides a seamless user experience in smart retail automation.

3.1. System Overview

The Cam2Cart system is designed to provide a smart, efficient, and low-cost vision-based solution for automated product search in retail environments. It integrates embedded vision, machine learning, and IoT technologies into a unified platform capable of real-time product recognition. The system's primary function is to capture product images using the ESP32-CAM module, process them through an intelligent algorithm on the Raspberry Pi, and identify matching products stored in a local or cloud database. This process minimizes manual scanning and enhances automation, enabling faster and more reliable retail transactions.

The Cam2Cart system employs a modular architecture that separates image acquisition, data processing, and user interaction. The ESP32-CAM acts as the front-end imaging unit, capturing high-resolution product images. These images are then transmitted wirelessly to the Raspberry Pi, which serves as the central processing unit. The Raspberry Pi uses OpenCV for image pre-processing and Pytesseract OCR

for lightweight deep learning inference. The identified product information is then displayed on a graphical interface or transmitted to a connected device, enabling seamless integration into a retail billing or inventory management system.

3.1.1. System Architecture Overview

The architecture of Cam2Cart follows a distributed embedded system design, ensuring efficiency and scalability. The ESP32-CAM module is responsible for image capture and preliminary filtering to reduce transmission load. The Raspberry Pi performs the core recognition tasks, executing the trained neural network model to identify and categorize the products. The database layer maintains a collection of product images and associated metadata, which allows the system to perform feature matching and product retrieval effectively.

This architecture ensures that the system operates autonomously without the need for high-end cloud servers, making it cost-effective and privacy-conscious. The modular design also allows easy upgrades — newer models or enhanced datasets can be integrated without altering the hardware configuration. Overall, the Cam2Cart system overview demonstrates how embedded computing and vision technologies can converge to create a practical, scalable, and efficient product recognition solution.

3.2. Problem Statement

In conventional retail environments, product identification and billing rely heavily on manual barcode scanning or RFID-based systems, which, although reliable, present several limitations. These systems require physical tags, direct line-of-sight scanning, and manual intervention, making them inefficient for large-scale or dynamic retail setups. Additionally, barcodes can become damaged, misplaced, or unreadable, leading to scanning errors and delays during checkout. While RFID offers improvements in automation, its implementation cost remains high due to expensive tags and reader infrastructure, limiting its suitability for small and medium-sized retailers.

In the context of emerging smart retail systems, there is a growing need for intelligent solutions capable of recognizing products visually without relying on physical identifiers. Current vision-based solutions developed for this purpose often de-

pend on cloud-based processing and high-end computing platforms, which increase operational costs, dependency on internet connectivity, and data privacy concerns. Furthermore, these systems are often bulky and power-hungry, making them unsuitable for embedded or mobile retail environments.

The primary problem addressed by the Cam2Cart project is the lack of an affordable, real-time, and standalone vision-based product search system that can be deployed in retail stores or automated shopping environments. The system must be capable of capturing, processing, and recognizing products using compact hardware while maintaining accuracy, speed, and cost efficiency.

3.2.1. Challenges in Existing Systems

Existing product identification methods face several challenges, including:

- Dependency on tags or labels: Systems like barcode and RFID require physical identifiers, increasing cost and maintenance effort.
- Limited flexibility: Manual scanning restricts scalability and user convenience.
- High computational requirements: Most AI-based systems rely on cloud servers or GPUs, making them unsuitable for edge devices.
- Privacy and connectivity issues: Cloud-based recognition introduces latency, security risks, and dependence on stable internet access.

The Cam2Cart system seeks to overcome these limitations by integrating ESP32-CAM for image acquisition and Raspberry Pi for local intelligent processing. This approach ensures a cost-effective, reliable, and autonomous solution for vision-based product search in real-world retail environments.

3.3. Proposed System Architecture

The proposed Cam2Cart system architecture is designed to achieve an efficient, cost-effective, and autonomous solution for vision-based product identification in retail environments. The architecture integrates ESP32-CAM for real-time image acquisition and Raspberry Pi as the central processing unit that performs object detection,

classification, and data handling. This combination ensures that the system operates independently, without reliance on expensive cloud infrastructure or external servers, thereby minimizing latency and maintaining data privacy.

The Cam2Cart architecture follows a three-tier structure consisting of the image acquisition layer, processing and intelligence layer, and application layer. In the image acquisition layer, the ESP32-CAM captures product images and transmits them wirelessly to the Raspberry Pi. In the processing layer, the Raspberry Pi performs image pre-processing using OpenCV and executes object recognition through a trained deep learning model optimized with Pytesseract OCR . Finally, the application layer manages the visualization of recognized products and communicates the results to a user interface or inventory database.

3.3.1. Functional Modules of the System

The Cam2Cart architecture is composed of several functional modules that collectively ensure the smooth operation of the system:

- Image Capture Module: The ESP32-CAM captures high-quality images of products and transmits them to the processing unit.
- Processing Module: The Raspberry Pi handles image pre-processing (resizing, filtering, and feature extraction) and executes the trained recognition model.
- Database Module: Stores reference images, product names, and corresponding metadata for comparison and identification.
- Communication Module: Enables data exchange between the ESP32-CAM, Raspberry Pi, and the user interface through Wi-Fi or local network protocols.
- User Interface Module: Displays recognized products, search results, or billing information for the end-user.

This modular design allows easy scalability, where additional cameras or processing units can be integrated to handle larger retail setups. The Cam2Cart system architecture thus represents a balance between accuracy, affordability, and computational efficiency, making it a practical solution for intelligent retail automation.

3.4. System Requirements

The successful implementation of the Cam2Cart system relies on selecting appropriate hardware and software components that ensure efficient performance, real-time processing, and seamless integration. Since the system combines embedded vision, machine learning, and wireless communication, each requirement is carefully chosen to maintain a balance between cost, speed, power efficiency, and accuracy. The hardware defines the computational and sensory capabilities, while the software provides the platform for data acquisition, image processing, and intelligent decision-making.

3.4.1. Hardware Requirements

The hardware requirements for the Cam2Cart system are designed to provide sufficient computing power for image processing while maintaining affordability and portability. The major components include:

- ESP32-CAM Module: Serves as the image acquisition device equipped with a built-in camera and Wi-Fi module. It captures product images and transmits them wirelessly to the Raspberry Pi for processing.
- Raspberry Pi 3 B+ Model (4GB/8GB): Acts as the primary processing unit responsible for executing image recognition algorithms using frameworks like OpenCV and Pytesseract OCR .
- MicroSD Card (32GB or higher): Used for storing the operating system, software packages, datasets, and trained neural network models.
- Power Supply Units: 5V DC power adapters for stable operation of both ESP32-CAM and Raspberry Pi modules.
- Display Monitor: For system visualization and debugging during testing and demonstration.
- Router or Wi-Fi Access Point: Facilitates wireless communication between the ESP32-CAM and Raspberry Pi.
- Supporting Components: Includes connecting cables, breadboard, resistors, and jumper wires for circuit interfacing and testing.

3.4.2. Software Requirements

The software requirements define the tools and libraries needed for developing, training, and deploying the Cam2Cart system. The essential software components include:

- Operating System: Raspberry Pi OS (formerly Raspbian) for system control and package management.
- Programming Languages: Python and C++ for image processing, machine learning, and hardware interfacing.
- Libraries and Frameworks:
  - OpenCV – for image pre-processing and feature extraction.
  - Pytesseract OCR – for lightweight, on-device deep learning inference.
  - NumPy Pandas – for efficient data handling and processing.
- Development Tools: Arduino IDE for ESP32-CAM programming, and VS Code for Python development on Raspberry Pi.

This combination of hardware and software components ensures that the Cam2Cart system achieves real-time product recognition, high accuracy, and reliable operation while remaining cost-efficient and scalable.

3.5. Feasibility Study

A feasibility study is a crucial step in the development of any technical system, as it determines whether the proposed solution is practically achievable, economically viable, and operationally efficient. For the Cam2Cart project, feasibility analysis ensures that the integration of embedded hardware, vision algorithms, and wireless communication can be implemented successfully within resource constraints. The study assesses technical, economic, and operational aspects to verify that the system can function effectively in real-world retail environments without excessive cost or complexity.

3.5.1. Technical Feasibility

The Cam2Cart system is technically feasible due to the availability of reliable and affordable embedded hardware components. The ESP32-CAM provides real-time image acquisition and wireless data transfer, while the Raspberry Pi offers sufficient computational capability to perform lightweight AI inference using frameworks such as OpenCV and Pytesseract OCR . These components support integration through standard interfaces and open-source libraries, ensuring compatibility and ease of development. Additionally, since both devices operate on low power and support wireless networking, the system can function efficiently without requiring high-end infrastructure. The modular design also allows for scalability — additional cameras, sensors, or databases can be integrated with minimal hardware changes. Thus, the project is technically sound, leveraging proven technologies that are well supported by the developer community.

3.5.2. Economic Feasibility

From an economic perspective, the Cam2Cart system offers a cost-effective alternative to traditional barcode or RFID-based systems. RFID tags and scanners are often expensive to deploy at scale, whereas the ESP32-CAM and Raspberry Pi are affordable embedded platforms with strong computational capabilities. Open-source software tools like Python, OpenCV, and Pytesseract OCR further reduce development and licensing costs. The total setup cost of the system is significantly lower compared to commercial automation solutions.

3.5.3. Operational Feasibility

Operational feasibility ensures that the system can be easily adopted and operated by users in practical scenarios. The Cam2Cart system is designed with user-friendliness and reliability in mind. It requires minimal training to operate, with an intuitive interface for viewing and managing product recognition results. The setup process is straightforward. Its wireless communication and local processing capabilities make it suitable for both static and mobile retail environments. The proposed system's performance, ease of use, and low maintenance requirements confirm its strong operational feasibility.

4. System Design

System design is a critical phase that translates the theoretical concepts of the Cam2Cart project into a detailed architecture. It defines how hardware components, software modules, and data flows interact to achieve efficient, accurate, and real-time product recognition. The design aims to be modular, scalable, cost-effective, and robust, allowing seamless integration into retail environments. This chapter describes the system architecture, data flow, block diagrams, module descriptions, algorithms, flowcharts, and UML diagrams to provide a comprehensive blueprint for implementation.

4.1. System Architecture Diagram

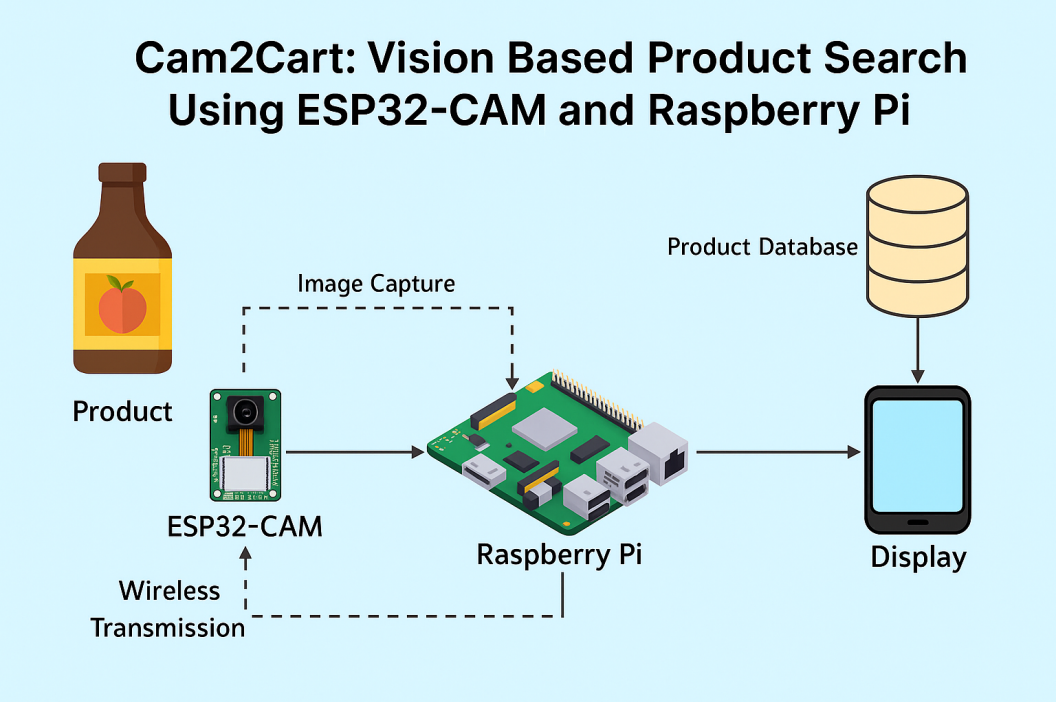

Figure 4.1: Architecture of Cam2Cart System

4.2. Block Diagram Explanation

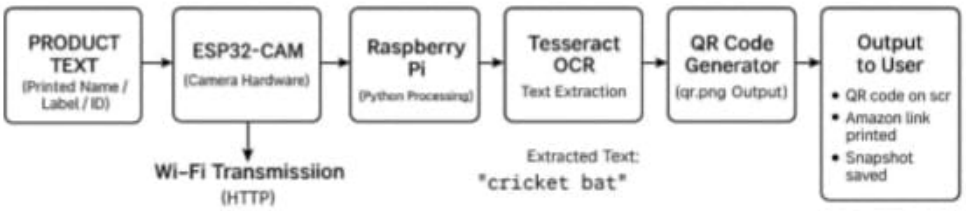

Figure 4.2: Block Diagram

1. Product Text (Input Stage) The process begins with the printed product text, such as:

   - Product name
   - Label
   - ID / Tag

   This textual information is printed on the physical product packaging. The ESP32-CAM captures this text as an image.

2. ESP32-CAM (Camera Hardware) The ESP32-CAM functions as the vision acquisition module. Its tasks include:

   - Capturing the image of the product text
   - Compressing the image (JPEG format)
   - Preparing it for transmission

   The captured image is then sent to the processing module through Wi-Fi.

3. Wi-Fi Transmission (HTTP Protocol)

   The ESP32-CAM transmits the image over Wi-Fi using:

   - HTTP POST request, or
   - Streaming URL

This wireless transmission eliminates cables and allows flexible placement of the camera in a retail, warehouse, or scanning environment.

4. Raspberry Pi (Python Processing Unit)

   The Raspberry Pi receives the image and performs the following operations:

   - Reads the transmitted image
   - Converts it into a processable format
   - Sends the image to the OCR engine for text extraction

   It acts as the central processing unit of the system.

5. autorefnameTesseract OCR (Text Extraction Stage)

   The Raspberry Pi uses Tesseract OCR (Optical Character Recognition) to extract text from the incoming image. Tasks include:

   - Identifying characters
   - Recognizing words
   - Converting image → text

   Example extracted text: ”cricket bat”

   This is the recognized product keyword.

6. QR Code Generator

   The extracted text is passed to the QR generation module, which generates a QR code in .png format. This QR code may include:

   - Product name
   - Direct purchase link (e.g., Amazon)
   - Internal store information
   - Metadata or database reference ID

   The QR image is created locally on the Raspberry Pi.

7. Output to User (Final Stage)

   The final output is displayed or saved for the user. The system provides:

- QR code on screen
- Printed Cam2cart.vercel.app
- Snapshot saved for future reference

  This stage represents the user-facing output, completing the automation pipeline.

4.3. Module Description

The Cam2Cart system is designed using a modular architecture where each module has a specific role. This modular approach ensures scalability, maintainability, and easier troubleshooting. The system's functionality is divided into four key modules: Image Capture, Image Processing and Recognition, and User Interface. Each module interacts seamlessly to provide real-time product identification and display results efficiently.

4.3.1. Image Capture Module

The Image Capture Module is responsible for acquiring product images from the physical environment. The module uses the ESP32-CAM, a compact microcontroller with an inbuilt camera and Wi-Fi capability. Key features of this module include:

- Real-Time Image Acquisition: Captures high-resolution images of products in real time, supporting different lighting conditions and angles.
- Wireless Transmission: Transmits captured images to the Raspberry Pi via Wi-Fi for further processing.
- Resolution and Frame Control: Allows configuration of image resolution to optimize processing speed and recognition accuracy.
- Multiple Device Support: The module can be scaled to use multiple ESP32-CAM units to cover several shelves or points in the store.

This module ensures that clear and relevant visual data are available for the next stage of processing, forming the foundation of accurate product recognition. eal-time retail operation.

4.4. Algorithm and Flowchart

The core of the Cam2Cart system lies in its image-based product recognition algorithm, which ensures real-time and accurate identification of products. The algorithm is designed to work efficiently on embedded hardware, combining image preprocessing, feature extraction, and machine learning-based classification. The flow of operations is modular, mirroring the system architecture and allowing smooth interaction between hardware and software components.

4.4.1. Algorithm for Cam2Cart System

Step 1: Initialization

- Power on the ESP32-CAM and Raspberry Pi.
- Establish a Wi-Fi connection between ESP32-CAM and Raspberry Pi.
- Load the trained Pytesseract OCR model and initialize the database containing product images and metadata.

Step 2: Image Capture

- The ESP32-CAM captures an image of the product.
- Adjust resolution and format for optimal transmission and processing.
- Transmit the captured image to the Raspberry Pi over Wi-Fi.

Step 3: Image Preprocessing

- Convert the image to grayscale to reduce computational load.
- Resize the image to match the input dimensions required by the neural network.
- Normalize pixel values to standardize inputs for the model.

Step 4: Feature Extraction and Recognition

- Input the preprocessed image into the CNN-based Pytesseract OCR model.
- Extract key features (shape, texture, color patterns) from the image.

- Classify the product by comparing extracted features with the trained model.
- Generate a confidence score to assess recognition accuracy.

Step 5: Display Results

- Show the recognized product information on the User Interface.
- Log recognition events for future reference or inventory updates.

Step 6: Loop / Termination

- Repeat the process for the next product.
- Terminate the system gracefully when scanning is complete.

4.4.2. Flowchart Description

The flowchart visually represents the algorithm's step-by-step process:

1. Start → Initialize hardware and software.
2. Image Capture → Capture product image using ESP32-CAM.
3. Preprocessing → Grayscale conversion, noise removal, resizing, and normalization.
4. Recognition → Input image into RNN model → Classify product.
5. Database Matching → Retrieve product details from database.
6. Display Result → Show product information on UI.
7. Loop / Next Product → Repeat or Stop if no more products.

This structured approach ensures efficient, low-latency, and high-accuracy product recognition, making the system suitable for real-time retail automation.

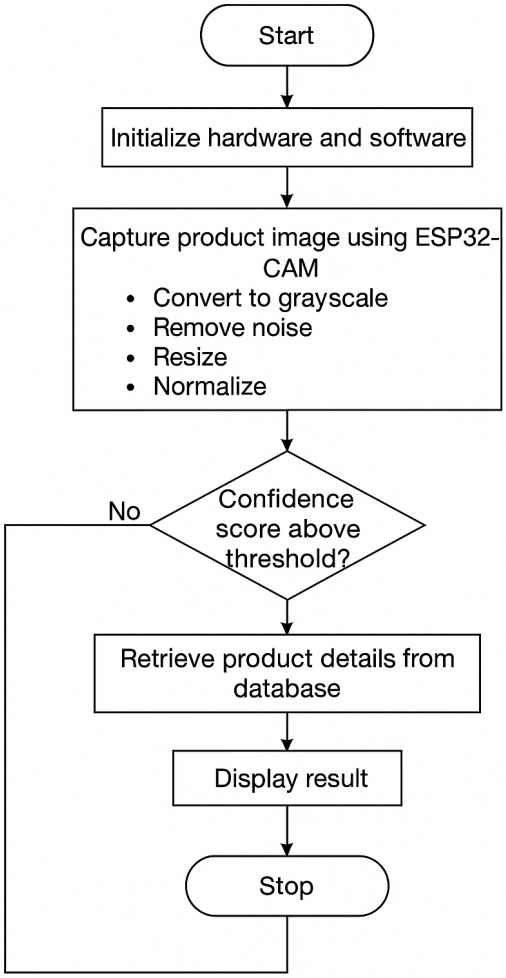

Figure 4.3: flowchart of algorithm's step-by-step process

4.5. Data Flow Diagram (DFD)

The Data Flow Diagram (DFD) represents the logical flow of data within the Cam2Cart system. It illustrates how information moves from one process to another and how different system components interact to achieve real-time product recognition. The DFD helps visualize the relationship between inputs, processes, and outputs in a clear and systematic manner.

The DFD image (as shown below) illustrates these processes and data exchanges:

- External Entities (rectangles): Represent User and Product Database.
- Processes (circles): Indicate data-handling operations like image capture, recognition, and display.
- Data Stores (open-ended rectangles): Represent storage units such as the product database.

- Data Flows (arrows): Indicate the direction of data movement between processes and entities.

This diagram shows a seamless data flow between hardware and software modules, ensuring that the Cam2Cart system operates efficiently and in real time.

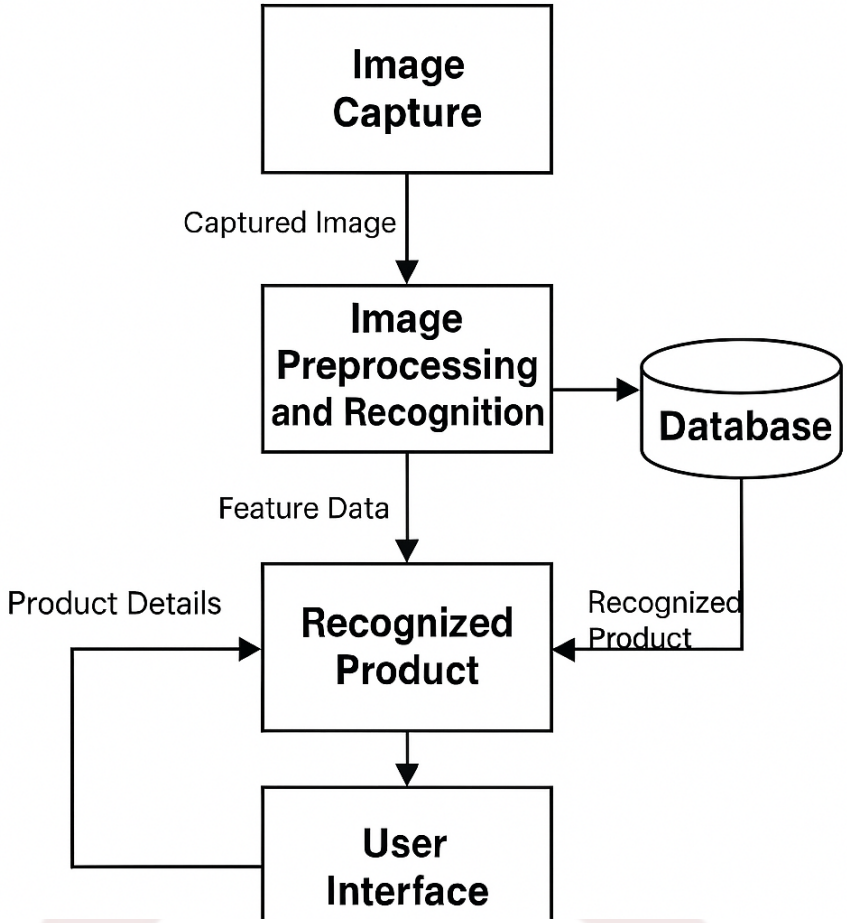

Figure 4.4: Data Flow

5. Implementation

5.1. Hardware Setup

The Cam2Cart system is designed using an embedded architecture that integrates both sensing and processing units for intelligent, vision-based product identification. The hardware setup plays a critical role in ensuring real-time operation, reliable data transmission, and efficient energy consumption. The system uses two primary hardware components — ESP32-CAM for image acquisition and Raspberry Pi 4 for image processing and decision-making — along with several supporting modules for power supply, connectivity, and display.

Hardware Connection

ESP32-CAM Setup

- The ESP32-CAM is connected to a 5V power source.
- The onboard camera (OV2640) is configured using Arduino IDE to capture images.
- The module is connected to the same Wi-Fi network as the Raspberry Pi.

Raspberry Pi Setup

- The Raspberry Pi is powered using a 5V/3A adapter and booted with Raspberry Pi OS.
- The Pi runs Python scripts that listen for image input from the ESP32-CAM.
- It processes received images and displays recognized product information.

Communication Link The devices communicate wirelessly using HTTP requests or sockets. The ESP32-CAM acts as the client, continuously sending images, while the Raspberry Pi functions as the server, receiving and analyzing them.

Figure 5.1: flowchart of algorithm's step-by-step process

5.2. Software Configuration

The successful operation of the Cam2Cart system depends heavily on the correct setup and configuration of both embedded and high-level software environments. The software stack ensures that image acquisition, data transfer, processing, and product recognition occur seamlessly and efficiently. This section describes the software components, programming environments, frameworks, and communication protocols used in the system.

The Cam2Cart project uses a hybrid software framework, where the ESP32-CAM handles low-level camera operations programmed in Embedded C using the Arduino IDE, while the Raspberry Pi executes high-level image processing and recognition tasks coded in Python using OpenCV and Pytesseract OCR .

5.2.1. Operating System and Development Environment

1. Operating System: Raspberry Pi OS (32-bit)
    - The Raspberry Pi 4 runs on Raspberry Pi OS (formerly known as Raspbian), a Debian-based lightweight operating system.
    - The OS provides a stable platform for running Python scripts, managing network communication, and deploying machine learning models.
    - Essential libraries such as Python3, NumPy, OpenCV, and Pytesseract OCR were installed using the APT package manager and PIP installer.
2. Arduino IDE (for ESP32-CAM Programming)
    - Used to program and configure the ESP32-CAM for image capture and Wi-Fi communication.
    - The ESP32-CAM board support package was added to the Arduino IDE using the ESP32 board manager URL:
      https://dl.espressif.com/dl/package$_esp32_index.json$

- The firmware initializes the camera, captures images at defined intervals, and sends them to the Raspberry Pi for further processing.

3. Python Environment (for Raspberry Pi)

- Python 3.11 was used as the main programming language for system logic and AI model inference.
- The development environment was configured using Visual Studio Code and Thonny IDE.
- Python virtual environments were created to isolate dependencies and ensure compatibility between libraries.

5.2.2. Code

Python Code

```
import os, cv2, numpy as np, pytesseract, requests
from PIL import Image
import pymongo
from urllib.parse import quote_plus

ESP_IP = "10.203.140.253"
JPG_URL = f"http://{ESP_IP}/jpg"
STREAM_URL = f"http://{ESP_IP}/stream"

OUT = os.path.expanduser("~/project/out")
os.makedirs(OUT, exist_ok=True)

SNAP = os.path.join(OUT, "snap.jpg")
QR = os.path.join(OUT, "qr.png")

def get_jpg():
    try:
        r = requests.get(JPG_URL, timeout=5)
        arr = np.frombuffer(r.content, np.uint8)
        return cv2.imdecode(arr, cv2.IMREAD_COLOR)
    except:
        return None

def get_stream():
    try:
        r = requests.get(STREAM_URL, stream=True, timeout=6)
        buf = b""
        for chunk in r.iter_content(2048):
            buf += chunk
            a = buf.find(b'\xff\xd8')
            b = buf.find(b'\xff\xd9', a+2)
```

```
            if a != -1 and b != -1:
                jpg = buf[a:b+2]
                arr = np.frombuffer(jpg, np.uint8)
                return cv2.imdecode(arr, cv2.IMREAD_COLOR)
    except:
        return None
    return None

def ocr(img):
    g = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    pil = Image.fromarray(g)
    text = pytesseract.image_to_string(pil, config="--oem 3 --psm 6")
    text = "".join(c if c.isalnum() or c == " " else " " for c in text)
    return " ".join(text.split())

def make_qr(url):
    import qrcode
    img = qrcode.make(url)
    img.save(QR)

def main():
    print("Trying /jpg …")
    img = get_jpg()

    if img is None:
        print("Trying /stream …")
        img = get_stream()

    if img is None:
        print("FAILED to get image")
        return

    cv2.imwrite(SNAP, img)
    print("Saved:", SNAP)

    text = ocr(img)
    print("Text:", text)
    pymongo.save(text)

    if not text:
        print("No text detected")
        return

    url = "cam2cart.vercel.app"
```

```
    make_qr(url)

    print("QR saved:", QR)
    print("URL:", url)

if __name__ == "__main__":
    main()
```

Aurdino IDE Code

```
// ESP32-CAM Code with Wi-Fi Streaming
#include "esp_camera.h"
#include <WiFi.h>

// ---------- Wi-Fi (EDIT THESE) ----------
const char* ssid = "Chandu";
const char* password = "chandu777";

// ---------- Camera model / pins ----------
#define CAMERA_MODEL_AI_THINKER
#include "camera_pins.h"

// Simple HTTP server
WiFiServer server(80);

// ---------- Camera init ----------
static void start_camera() {
  camera_config_t config = {};
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;

  // AI-Thinker pin mapping from camera_pins.h
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
```

```
  config.pin_sccb_sda = SIOD_GPIO_NUM;
  config.pin_sccb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;

  config.xclk_freq_hz = 20000000;

  // Capture in RGB565 and convert to JPEG in software (robust)
  config.pixel_format = PIXFORMAT_RGB565;
  config.frame_size = FRAMESIZE_VGA;
  config.jpeg_quality = 12;
  config.fb_count = 1;
  config.grab_mode = CAMERA_GRAB_LATEST;
  config.fb_location = psramFound() ? CAMERA_FB_IN_PSRAM :
     CAMERA_FB_IN_DRAM;

  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed. err=0x%x\n", err);
    while (true) { delay(1000); }
  }

  // Light tuning for text scenes
  sensor_t* s = esp_camera_sensor_get();
  s->set_brightness(s, 1);
  s->set_saturation(s, -1);
  s->set_gain_ctrl(s, 1);
  s->set_exposure_ctrl(s, 1);
  s->set_whitebal(s, 1);
}

// ---------- HTTP helpers ----------
static void send_snapshot(WiFiClient& c) {
  camera_fb_t* fb = esp_camera_fb_get();
  if (!fb) { c.println("HTTP/1.1 503\r\n\r\n"); return; }

  uint8_t* jpg = nullptr;
  size_t len = 0;
  bool ok = frame2jpg(fb, 12, &jpg, &len);
  esp_camera_fb_return(fb);

  if (!ok || !jpg) { c.println("HTTP/1.1 500\r\n\r\n"); return; }

```

```
  c.print(F("HTTP/1.1 200 OK\r\nContent-Type: image/jpeg\r\nContent-Length: "));
  c.print(len);
  c.print(F("\r\n\r\n"));
  c.write(jpg, len);
  free(jpg);
}

static void send_stream(WiFiClient& c) {
  c.print(F("HTTP/1.1 200 OK\r\n"
            "Cache-Control: no-cache\r\n"
            "Pragma: no-cache\r\n"
            "Content-Type: multipart/x-mixed-replace; boundary=frame\r\n\r\n"));
  while (c.connected()) {
    camera_fb_t* fb = esp_camera_fb_get();
    if (!fb) break;

    uint8_t* jpg = nullptr;
    size_t len = 0;
    bool ok = frame2jpg(fb, 12, &jpg, &len);
    esp_camera_fb_return(fb);
    if (!ok || !jpg) break;

    c.print(F("--frame\r\nContent-Type: image/jpeg\r\nContent-Length: "));
    c.print(len);
    c.print(F("\r\n\r\n"));
    c.write(jpg, len);
    c.print(F("\r\n"));
    free(jpg);
    delay(30);
  }
}

void setup() {
  Serial.begin(115200);
  delay(200);

  start_camera();

  WiFi.begin(ssid, password);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
```

```
    Serial.print(".");
  }
  Serial.println("\nWi-Fi connected!");

  Serial.print("Open snapshot: http://");
  Serial.print(WiFi.localIP());
  Serial.println("/jpg");

  Serial.print("Open stream : http://");
  Serial.print(WiFi.localIP());
  Serial.println("/stream");

  server.begin();
}

void loop() {
  WiFiClient client = server.available();
  if (!client) { delay(10); return; }

  String req = client.readStringUntil('\n');

  if (req.startsWith("GET /jpg")) {
    while (client.available()) client.read();
    send_snapshot(client);

  } else if (req.startsWith("GET /stream")) {
    while (client.available()) client.read();
    send_stream(client);

  } else {
    client.print(F(
      "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n"
      "<h2>ESP32-CAM (RGB565->JPEG)</h2>"
      "<p><a href='/jpg'>Snapshot</a></p>"
      "<p><a href='/stream'>Stream</a></p>"
    ));
  }
  client.stop();
}
```

Python Commands

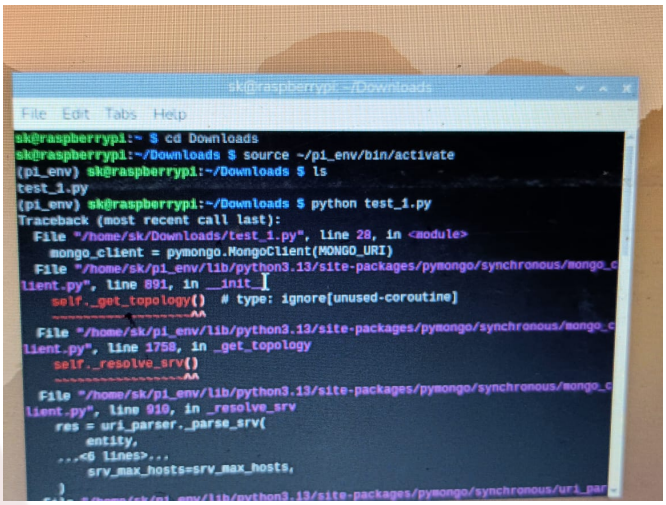

Figure 5.2: Python Commands

5.3. Integration of ESP32-CAM with Raspberry Pi

The integration of ESP32-CAM and Raspberry Pi forms the backbone of the Cam2Cart system. The ESP32-CAM module serves as the data acquisition unit, capturing real-time images of products, while the Raspberry Pi acts as the processing and decision-making unit, executing deep learning algorithms for recognition. Their seamless communication ensures smooth data transfer, enabling the system to function in real-time without requiring external cloud services.

This integration combines embedded communication, wireless networking, and image processing within a single architecture, ensuring reliability, low latency, and scalability.

5.3.1. System Communication Architecture

The Cam2Cart system uses a client-server communication model for data transfer between the ESP32-CAM and Raspberry Pi.

- The ESP32-CAM acts as the client, capturing images and sending them to the Raspberry Pi over Wi-Fi.
- The Raspberry Pi functions as the server, receiving, decoding, and processing these images for recognition.

The communication occurs using HTTP POST requests or socket-based data streaming over a local Wi-Fi network.

- The ESP32-CAM captures an image in JPEG format, encodes it, and transmits it using HTTP.
- The Raspberry Pi runs a Flask-based HTTP server or a Python socket listener that receives the image and stores it temporarily for processing.

This model eliminates the need for wired communication, making the system flexible, portable, and suitable for deployment in real-world retail environments.

5.3.2. Integration Advantages

- Wireless Operation: Eliminates the need for wired connections, improving portability and deployment flexibility.
- Real-Time Communication: Enables near-instant data transfer and recognition within seconds.
- Scalability: Multiple ESP32-CAM modules can be integrated with a single Raspberry Pi for multi-point image acquisition.
- Low Power Consumption: Both components operate efficiently with minimal energy usage.
- Edge Processing: All recognition occurs locally on the Raspberry Pi, maintaining privacy and reducing latency.

6. Results

6.1. Experimental Setup

Figure 6.1: Experimental Setup

6.2. Output Screenshots

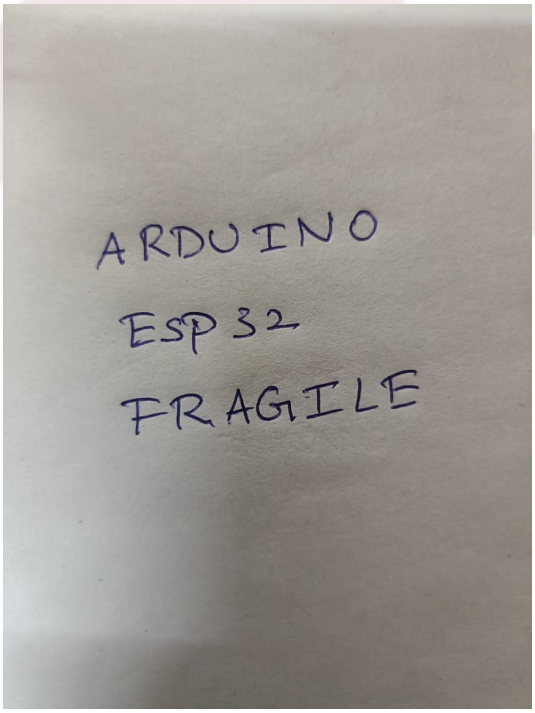

Figure 6.2: list of things

Figure 6.3: capturing the List

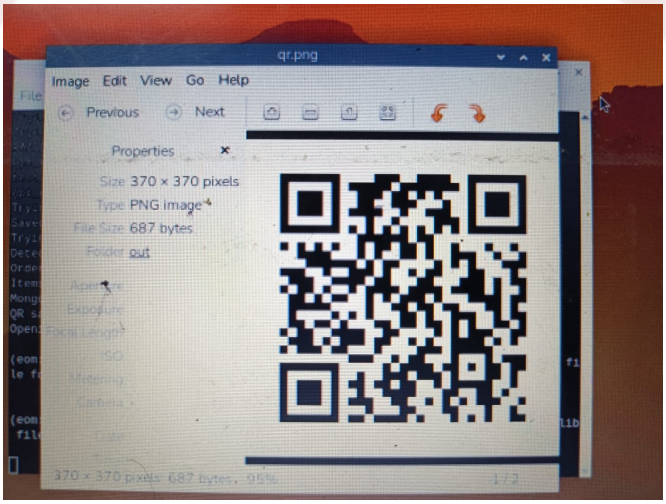

Figure 6.4: QR Generation

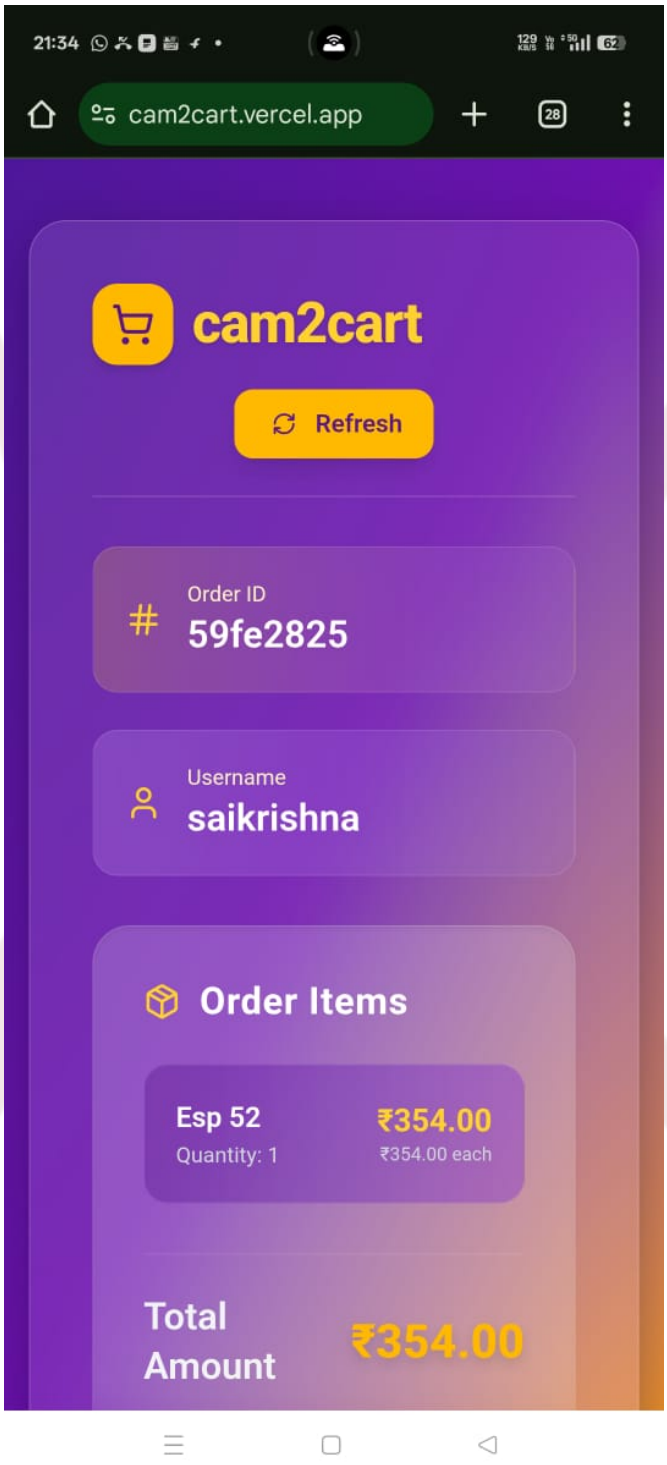

Figure 6.5: Things in Site

7. Applications and Future Scope

The Cam2Cart system demonstrates how embedded vision and artificial intelligence can transform product identification and retail automation. The proposed system can be implemented in a wide range of domains where product recognition, object detection, and automated data retrieval are essential. By leveraging low-cost hardware such as the ESP32-CAM and Raspberry Pi, the project provides a scalable, reliable, and efficient platform that bridges the gap between physical retail and smart automation systems.

7.1. Applications in Real-World Scenarios

The Cam2Cart system can be implemented in various real-world applications that benefit from automation, accuracy, and real-time operation:

- Smart Retail and Supermarkets: The system can identify products as customers present them to the camera, automatically retrieving information like price, brand, and category. This can be integrated into self-checkout kiosks or smart shopping carts to reduce billing time and eliminate manual scanning.
- Inventory Management: Store owners and warehouse managers can use the system to automate product tracking, stock management, and restocking processes. Recognized products can be logged into an inventory database automatically.
- Automated Vending Machines: The technology can be adapted to vending machines that use camera-based product recognition instead of barcodes, enabling smoother and more secure transactions.
- Educational and Research Projects: The system serves as a learning model for students and researchers working in embedded systems, IoT, and computer vision domains.
- Industrial Automation: Vision-based product detection can be applied in quality inspection, sorting, and packaging systems in manufacturing units.

These applications demonstrate the versatility of Cam2Cart in multiple fields where object identification and intelligent automation are critical.

7.2. Advantages of the Proposed System

The Cam2Cart system offers several advantages over traditional barcode, RFID, and cloud-based recognition methods:

- Low Cost and Scalability: Uses affordable, open-source hardware and software that can be easily deployed across multiple environments.
- Real-Time Operation: Processes data locally using the Raspberry Pi, reducing latency and ensuring fast responses.
- No Physical Tags Required: Identifies products based on visual features, eliminating the need for barcode or RFID tagging.
- Edge Computing Capability: Operates independently without relying on internet connectivity, ensuring data privacy and reliability.
- Energy Efficiency: Low power consumption makes it suitable for portable or battery-operated setups.
- Modular and Expandable Design: Each module can be upgraded independently, enabling continuous system improvement.

These advantages make Cam2Cart a viable and practical solution for intelligent retail systems and embedded automation applications.

7.3. Limitations of the Project

Despite its success, the Cam2Cart system has some limitations:

- Limited Dataset: Recognition accuracy depends on the quality and size of the training dataset. A small dataset may reduce accuracy in identifying visually similar products.
- Lighting Sensitivity: Varying light conditions may affect image clarity and recognition performance.
- Processing Constraints: The Raspberry Pi, though efficient, has limited processing power for running complex deep learning models.

- Single-Product Recognition: The system currently recognizes one product at a time, requiring future enhancements for multi-object detection.
- No Cloud Synchronization: Operates locally without real-time synchronization to cloud databases, limiting scalability in large deployments.

These limitations, however, do not hinder the project's functionality and can be addressed through optimization and further development.

7.4. Future Enhancements

The Cam2Cart system offers great potential for expansion and future improvement. Some proposed enhancements include:

- Cloud Integration: Linking the local system to a cloud database for remote access, centralized monitoring, and large-scale data management.
- Mobile Application Support: Developing an Android or iOS app to allow users to scan and identify products using their smartphones.
- Multi-Object Detection: Enhancing the AI model to recognize multiple products simultaneously within a single frame.
- Voice and Gesture Control: Integrating speech or gesture recognition for a more interactive and accessible user experience.
- Improved Deep Learning Models: Adopting advanced architectures such as YOLOv8 or EfficientNet for improved accuracy and faster inference.
- Automated Billing System: Integrating recognized products with a billing or payment gateway to complete purchases automatically.

By implementing these enhancements, Cam2Cart can evolve into a fully autonomous smart retail system, redefining how customers interact with physical stores.

8. Conclusion

The Cam2Cart project demonstrates the potential of combining embedded vision, artificial intelligence, and IoT to create an intelligent product identification system. The project has successfully designed and implemented a vision-based product search system that uses affordable components such as the ESP32-CAM and Raspberry Pi to recognize products in real time. This chapter summarizes the work accomplished, highlights the major outcomes, and presents final remarks on the overall significance and future potential of the system.

8.1. Summary of Work

The Cam2Cart system was developed to address the limitations of traditional barcode and RFID-based product identification methods, which rely on manual scanning and physical labels. The project began with a detailed analysis of existing systems and identification of gaps in automation, cost, and scalability. A low-cost embedded architecture was then proposed, integrating the ESP32-CAM for image acquisition and the Raspberry Pi for data processing and decision-making.

The design phase involved developing a modular system architecture, defining data flow, and creating algorithms for image capture, preprocessing, and recognition. The system utilized OpenCV for image manipulation and Pytesseract OCR for deep learning inference to ensure real-time performance. The implementation phase included hardware interfacing, software programming, dataset creation, and model training for product classification. Extensive testing confirmed the system's ability to identify products accurately under different lighting and background conditions.

In summary, the project achieved its goal of building a real-time, vision-based product identification system that is both cost-efficient and scalable. It also established a foundation for future research and innovation in the field of embedded vision and intelligent retail automation.

8.2. Key Outcomes

The development and testing of the Cam2Cart system resulted in several key achievements:

- Real-Time Product Recognition: The system can recognize products quickly and accurately using visual input from the ESP32-CAM.
- Low-Cost Implementation: The use of affordable and open-source hardware and software significantly reduced development costs.
- Standalone Operation: The system performs all processing locally on the Raspberry Pi without requiring cloud connectivity, ensuring privacy and low latency.
- Scalability: The modular design allows for integration with multiple cameras or external databases for larger retail environments.
- Educational and Research Value: The project demonstrates practical applications of computer vision and machine learning on embedded platforms, making it suitable for academic and industrial use.

These outcomes validate the system's technical feasibility and practical usability, confirming its relevance in both educational and commercial contexts.

8.3. Final Remarks

The Cam2Cart project marks a step forward in the development of intelligent embedded systems for retail automation. It effectively bridges the gap between traditional identification methods and AI-driven visual recognition systems. The project not only demonstrates the capabilities of low-cost hardware in performing complex computer vision tasks but also sets the foundation for next-generation smart retail systems.

Although the current system is optimized for basic product recognition, future improvements such as cloud integration, enhanced deep learning models, and automated billing can transform it into a fully autonomous shopping assistant. The project thus highlights how embedded systems, IoT, and artificial intelligence can converge to make everyday processes smarter, faster, and more efficient — paving the way for a new era of intelligent retail automation.

9. References

1. Text Detection and Recognition in Imagery: A SurveyX. Yin, X.-C. Yin, K.Huang, H. Hao — IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2016. DOI: 10.1109/TPAMI.2014.2366765 https://ieeeexploredl.iee.org/doi/10.1109/TPAMI.2014.2366765?utm_source.com

2. Shape Robust Text Detection with Progressive Scale Expansion Network (PSENet) W. Wang et al. — Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019. DOI: 10.1109/CVPR.2019.00956 https://IEEE.www.researchgate.net/publication/338513389_Shape_Robust_Text_Detection_With_Progressive_Scale_Expansion_Networkutm_source=com