PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

Ministry of Higher Education and Scientific Research

UNIVERSITY OF ECHAHID HAMMA LAKHDAR - EL OUED

FACULTY OF EXACT SCIENCES

Computer Science Department

Thesis

Submitted in partial fulfilment of the requirements for the Degree of

ACADEMIC MASTER

Field: Mathematics and Computer Science
Option: Computer Science
Specialty: Artificial Intelligence & Data Science

Presented by:

- **Haithem Sadallah**
- **Ahmed Aouadi**

Theme

Smart Checkout System

Examination Committee:

Dr. Khebbache Mohib Eddine MCB Supervisor

Dr. Naoui Mohammed Anouar MCA Co-Supervisor

Dr. Beggas Mounir MCA President

Dr. Guia Sana Sahar MCB Examiner

Academic year: 2024-2025

Thanks and Gratitude

First, we would like to thank Allah for giving us the strength, the will, the courage, and the patience to complete this modest project.
We also would like to thank Professor, our Supervisor and Co-Supervisor, who guided our work, for his availability to provide us with advice, for his trust, and for his invaluable help.
We thank him from the bottom of our hearts.
Our sincere thanks go to all the members of the jury who agreed to judge our modest work.
We express our sincere thanks to all the teaching and administrative staff in the Computer Science Department and especially to those who taught us during our studies.
We also thank all those who participated directly or indirectly in the development of this work.

Dedication

To my Parents,

For their moral support and invaluable advice throughout my studies,

To my brothers and sisters,

For their unwavering support and infinite patience,

To my dear friends,

For their help and support during difficult times,

To my entire family,

To all those I love and those who love me.
Thank you very much.

ABSTRACT

This study introduces an intelligent Smart Checkout System based on state-of-the-art computer vision and deep learning technology to streamlines the retail transaction process. The system eliminates the traditional barcode scanning with the use of real-time instance segmentation for facilitate identification of products and billing. Adopting the state-of-the-art YOLO architectures (YOLOv8 and YOLOv11), the system can achieve high-precision detection under multiple scenarios of retail conditions including occlusions, varying lighting conditions, and complex product grouping. One private data set of 1,525 well-annotated product images was collected and utilized to train and test the models to impart real-world robustness. Experimental findings verify improved performance, where YOLOv11 achieved a mean Average Precision (mAP50) value of 0.97 and YOLOv8 achieved 0.95, while simultaneously retaining computational efficiency. The system addresses major retail automation challenges, such as inventory management, shoplifting prevention, and cost of operations reduction, with a focus on flexibility to accommodate regional market needs (e.g., cash economies in the Arab region). The primary contributions are a scalable deployment pipeline, model trade-off comparison, and deployment strategies for high-accuracy and resource-constrained environments. The project also aligns theoretical innovation with practical purpose, offering a model for retail modernization in emerging markets through AI.

Keywords :

Smart checkout systems, computer vision, deep learning, YOLO architectures, instance segmentation, retail automation, and AI in emerging markets.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

General Introduction

Speed of innovation in digital technology has transformed the retail business, and intelligent payment systems are one example of innovation for remodeling conventional shopping. With computer vision, deep learning, and IoT as drivers, the systems can drive checkout automation and transform customer experience. Their adoption globally is, nonetheless, lopsided, particularly in emerging high-growth markets like the Arab world where retail modernization continues to lag behind rapidly growing consumer aspirations. This potential-versus-reality disparity is the target issue addressed by this work.

The issue is rooted in intrinsic inefficiencies that plague conventional checkout systems. Hand-scanning by keying and cashier-executed transactions are unavoidable bottlenecks—off-peak hours have 30% of shoppers abandon shopping in frustration due to the lines being ridiculously long, and retailers lose astronomical sums on labor per employee that consumes 60% of operating expenses. Human variation in counting is the reason for $1.75 trillion loss in retail annually worldwide. Self-checkout checkout lanes on checkout kiosks do minimize such mistakes to a certain degree, but these still entail customer involvement and are not able to stop shoplifting or allow high-volume operation. Remarkably in the Arab world, where its retail markets are growing by 8% annually, these issues are amplified by the cash culture of the region and logistics' failure to accommodate technology. Intelligent alternatives currently on offer for Western markets are, in most instances, not suitable for local product, store formats, and consumer behavior, thereby creating the demand for adaptive alternatives in the short term.

Our work fills this demand by proposing an intelligent payment system that automatically detects and charges with advanced object detection. Behind it is a technical trade-off: millimeter-precise detection of mixed retail goods in actual light conditions—occluded, warped, or differently lit goods—but computationally lean enough for low-cost rollouts. The solution is founded on cutting-edge YOLO architectures (YOLOv8 and YOLOv11) region-specific product data sets designed for unproblematic operations across small shop stalls and large supermarkets. Beyond technological innovation, the system reduces adoption barriers through stepwise implementation processes tailored to Arab retail environments and multilingual interfaces.

Value is derived from the double impact of the project: for retailers, it reduces operational costs by 40% through personnel saving and theft deterrence, and creates sales through enhanced turnover; to customers, it minimizes checkout friction, which is in line with local cultural values of hospitality-driven service. The research approach encompasses theoretical abstraction and empirical evidence. Chapter 1 analyzes smart payment system archetypes, Chapter 2 describes computer vision frameworks, and Chapter 3 propose a prototype to the test in controlled retail environments to empirical testing. By bringing together AI innovation and localized market expertise, this project charts a reproducible path for deploying scalable autonomous checkout solutions where they are most needed. The optimal outcome is more than technical success, offering a template for how new economies can leapfrog conventional retail infrastructures to AI-facilitated efficiency.

CHAPTER 1

SMART CHECKOUT SYSTEMS

1.1 Introduction

Smart checkout systems employ technology such as computer vision, radio-frequency identification (RFID), artificial intelligence (AI), and machine learning to streamline the process of billing and payment. Smart checkouts are different from traditional point-of-sale (POS) systems in the sense that they are not programmed to remove lines and human involvement but provide consumers with a faster, more efficient experience. Smart checkout systems are prevalent in the retail shops of today, including supermarkets, convenience stores,and self-service kiosks.And in this chapter here, we are going to write about a project phase, i.e., learning about the existing system we will examine in great detail the different specific characteristics of the system to be studied. Last, the system problems and the project goals are established.

1.2 Definition

Smart checkout systems are shopping apps that provide a convenience factor when shopping by eliminating the customary checkout steps such as scanning products and standing in lines. They keep the stores efficient and consumers convenient transactions. There have been many designs, and they all have four phases: pre-purchase, check-in, product selection, and checkout.Checkout processes range from complete automation "just walk out" to authentication through mobile apps. Smart checkout like intelligent cart technology, robotic self-checkout kiosks, mobile, and artificial intelligence-based automation.They innovations aim to enable cost management and shopping experience more effortlessly and labor dependency minimized [5, 2].

1.3 Existing System vs Smart Checkout Systems

Shopping at a traditional store begins when a consumer enters a store, walks up and down the aisles, selects goods she desires, and is able to bring back unwanted ones. She then proceeds to the checkout area, where she will more likely than not have to wait in line. A clerk will scan barcodes of items one at a time or enter them in at check-out when she arrives. numbers if needed. The shopper exits the store after having paid. That consumes time, particularly in peak periods and when individuals wait in lines that are very long. Also, because regular shopping does not permit one to track independent shoppers in real time, theft is more probable.

With intelligent checkout systems, queues and waiting are history. Such types of systems do already exist in some form or other. Schögel and Lienhard state that four stages of the smart checkout are pre-purchase, check-in, product selection, and check-out, as can be seen from Figure 1.1.In some systems, one needs to download an app specific to one store beforehand. Amazon Go and Habitat by HonestBee, for instance, involve pre-download of software, whereas Apple Stores do not require access by way of one. Store entry is also achieved differently—biometric identification to scan consumers, others require barcode scanning, and some offer free entry. At entry, the system scans the products one selects. In other shops, it is automated through sensors, computer vision, and machine learning, but in others, consumers scan barcodes or a pre-registered loyalty card is swiped on a product price reader. Finally, checkout and payment offer a number of alternatives. Consumers pay for the purchase by bar-code scanning through the store's app, or biometric authentication [2].

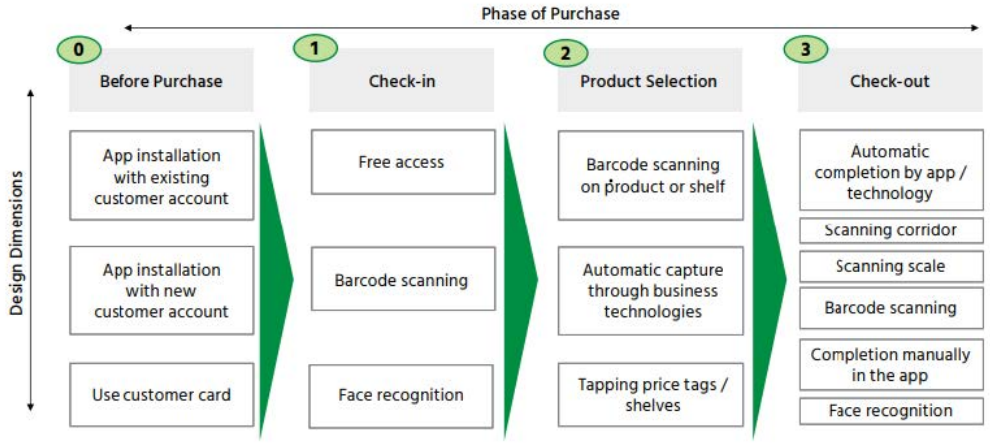


Figure 1.1: Design dimensions for each phase of the purchase[2]

1.4 Conceptual Designs and Prototypes

Researchers have brought in a number of new concepts and experimental approaches in the area of smart checkout systems. This chapter discusses conceptualizations and prototypes which have been designed to serve the deployments of such technologies and the problems they will most probably face. Much of the literature targets smart carts, with research differing in their approach at different levels, payment methods accommodated, challenges identified, and the scope as a whole of consumer buying. They are listed in Table 1.1

Table 1.1: Comparison of Smart Cart Methods [1]

		Stages				
	Method	**Before Purchase**	**Check-in**	**Product Selection**	**Check-out**	**Payment Methods**
[16]	Smart Cart			Manual capture through cart		
[17]	Smart Cart		Free access	Manual capture through cart		Payment through cart
[18]	Smart Cart			Automatic capture through cart	Automatic capture through counter	Traditional payment at the counter

[19]	Smart Cart	Registration	Free access	Automatic capture through cart	Traditional	Traditional payment at the counter
[20]]	Smart Cart	Mobile Application		Automatic capture through cart		Mobile Payment (AliPay)
[21]	Smart Cart			Automatic capture through cart		
[3]	Smart Cart		Free access	Automatic capture through cart		Online/ Offline
[22]	Smart Cart	Mobile Application		Automatic capture through cart		Online/ Offline
[23]	Smart Cart	Mobile Application		Automatic capture through cart		Online/ Offline
[24]	Self Checkout			Automatic capture at the checkout	After payment confirmation	Through QR scanner
[25]	Self Checkout		Free access	Manual capture at the checkout	Weight Comparison	NFC Payment
[26]	Self Checkout			Automatic capture at the checkout		Traditional payment at the counter

[4]	Mobile Checkout	Android Application	Scan QR-code	Scan barcode	Verification by supermarket authority	Online
[27]	Just Walk Out	Mobile Application	Mobile access	Automatic capture	Free exit	Charged automatically

1.4.1 Smart Cart Systems

In studies related to smart carts, two primary methods have been highlighted for the product selection process. The first method involves automatic detection of products by the cart, while the second method requires manual scanning of products using a reader mounted on the cart. One study focusing on manual product scanning emphasizes technical specifications but lacks detailed information about the system's overall functionality. Aimed at creating a seamless shopping experience, this study requires users to scan product barcodes using a reader integrated into the cart. This approach eliminates the need for individually scanning products at the checkout and provides instant information, such as price, contents, and customer reviews, directly to the shopper[16].

Another manual scanning study, conducted by Nallamothu et al., employs an radio-frequency identification (RFID)à reader. The objective of this study is to minimize customer wait times in queues and thereby improve the shopping experience. This system does not require any registration before use. Customers can simply take a cart from the store and start shopping. Each cart is equipped with an LCD screen that displays ongoing purchases. For instance, when a customer takes a cart, the message "Welcome to Smart Shopping" appears on the screen, indicating that the system is ready to use. Shoppers scan selected items using the scanner and add them to their carts. Unlike the previous study, this one also explains how to remove items from the cart. To do so, the customer must hold down the "reset" button while scanning the product, deducting its price from the total. Upon completing their shopping, customers can request a receipt by pressing a designated button and complete their payment through the cart[17].

In contrast, some smart checkout systems feature automatic product detection. One such system, known as the "Smart Trolley with IoT-Based Billing System," aims to enhance the shopping experience using RFID technology. Unlike manual scanning methods, this study does not cover the "before purchase" and "check-in" stages. Instead, it focuses on automatically detecting products as customers add them to the cart. An LCD screen on each cart displays the selected items, and removing an item simply involves taking it out of the cart, thanks to RFID technology. After finishing their shopping, customers proceed to the checkout counter, where the RFID reader automatically scans all items in the cart, eliminating the need for individual item scanning and reducing queue times [18].

Another study, titled "Implementation of Secure Smart Cart for Automatic Detection of Objects Using Arduino and RFID," also features automatic item recognition with traditional payment methods. This study is unique in that it clearly outlines the steps before product selection. To use the system, customers must register to obtain a unique user ID. Each cart in the store has a unique ID as well, and customers are assigned a cart upon registration. As they shop, customers can see the quantity and price of the items on the cart's screen. Upon completing their shopping, they must visit the checkout counter to make a payment. A distinctive feature of this system is its use of a weight sensor to compare the weight of added items with known weights, identifying any unscanned items. If discrepancies arise, an alarm notifies the customer or a staff member for

resolution [19].

In some smart cart systems, mobile payments are a key feature. An example of this is the system developed by Liu et al., which introduces an innovative smart cart design. This cart can follow its user through a visual sensing module, movement control module, and motion execution module. The visual sensing module captures images via the cart's camera, which are then analyzed to determine movement instructions. The movement control module calculates speed and rotation, while the motion execution module moves the cart accordingly. Another crucial feature of this smart cart is its ability to recognize products using RFID technology. The system seamlessly integrates with a mobile application, allowing customers to track items and make payments via AliPay after confirming their cart content. AliPay, a leading mobile payment platform, is particularly favored by smartphone users for online transactions [20].

Another study focusing on online payments is the "Automated Smart Trolley System using RFID Technology." This system automatically identifies items placed in or removed from the cart using RFID technology. As with most other systems, an LCD screen on the cart displays the total price of the selected items. However, unlike the previous study, this one directs customers to a payment page via a QR code displayed on the screen, rather than using a mobile application for payment [21].

A prominent approach in smart checkout systems is allowing both online and offline payments. In a study titled "An Intelligent Shopping Cart with Automatic Product Detection and Secure Payment System," customers can use the system without any prior registration. Similar to other smart cart solutions, this system automatically detects products added to the cart using RFID technology and displays item information on an LCD screen. The workflow of this process is depicted in Figure 1.2.

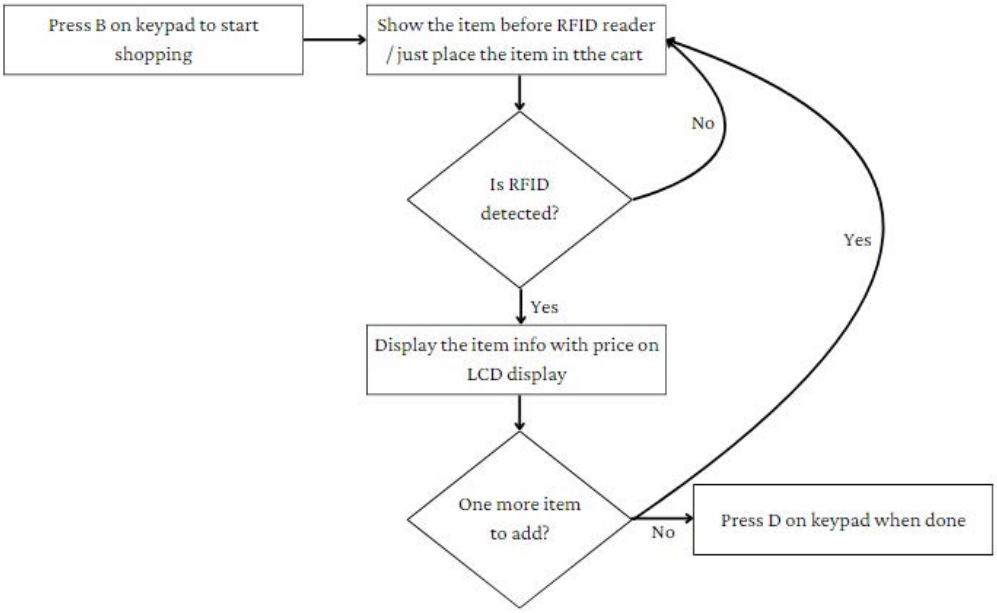


Figure 1.2: Model Workflow: Adding Items to the Cart[3].

After completing their shopping, customers have the option to either proceed to the checkout counter for traditional payment or complete their payment directly through the cart, as shown in Figure 1.3. This system is designed to offer a smart and secure payment process via the cart. To make a payment through the cart, customers must first enter their unique

IDs using the smart cart's keyboard.

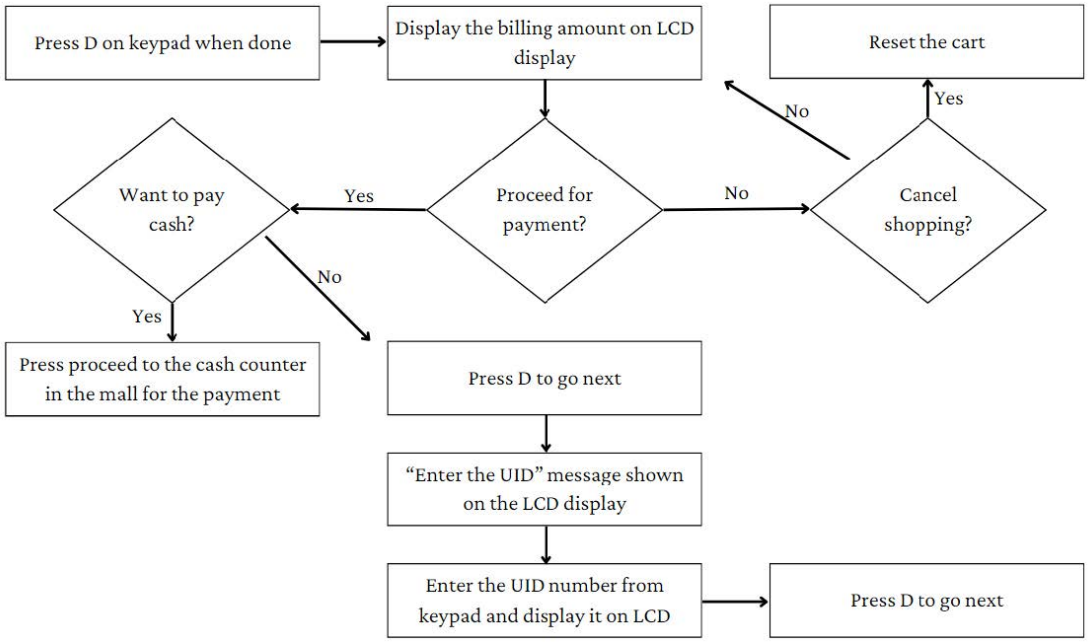


Figure 1.3: Model Workflow: Entering the UID for Authentication [3]

Next,customers need to verify their identity using their fingerprints. The biometric authentication process is illustrated in Figure1.4.

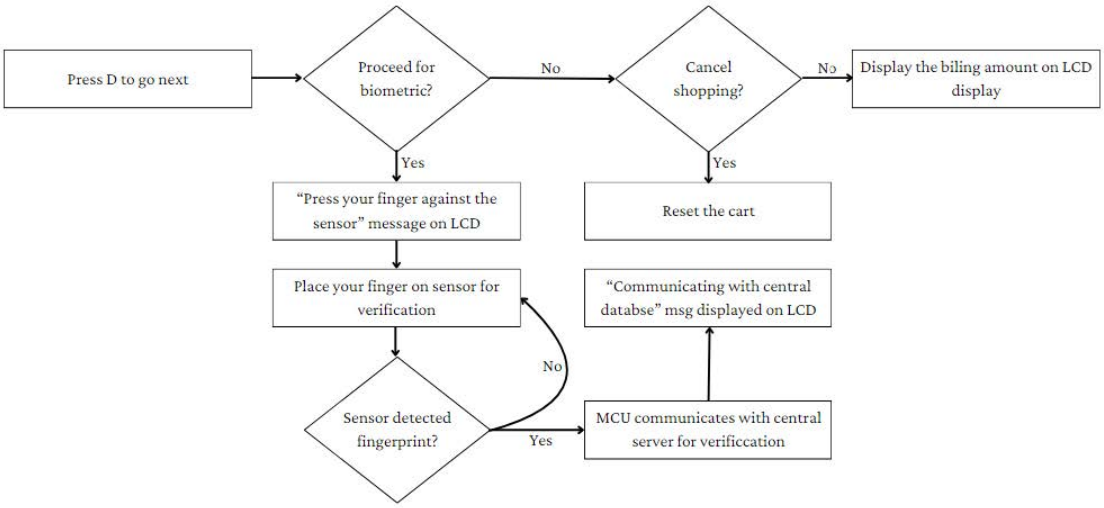


Figure 1.4: User Biometric Authentication[3]

Afterward, customers can complete their payments using either UPI (Unified Payments Interface) or OTP (One-Time Password). To make a payment via UPI, customers need to scan the QR code shown on the cart screen using their UPI payment app on their smartphones and then finalize the payment through the app. The process is illustrated in Figure1.5.

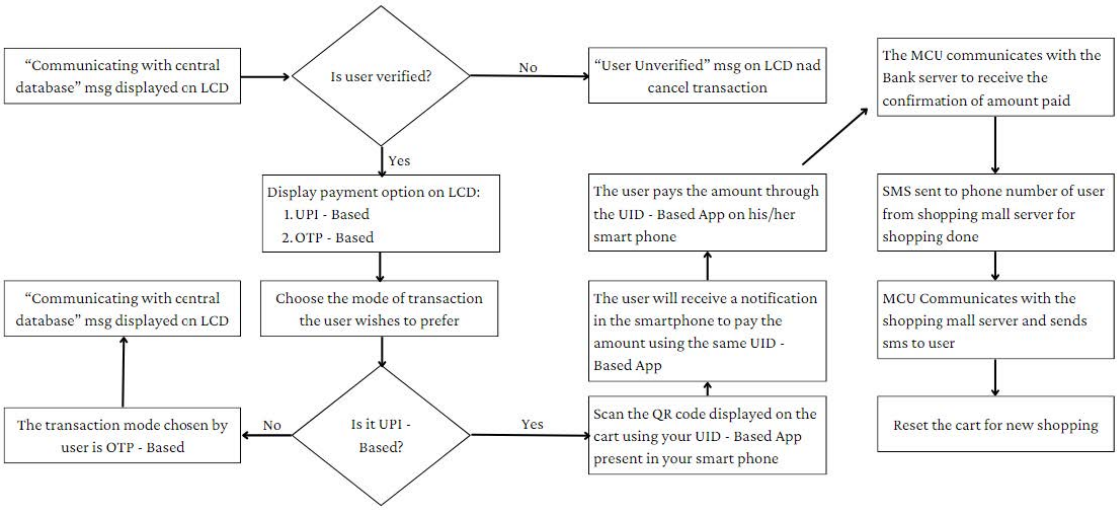


Figure 1.5: Payment via UPI[3]

For customers opting to make payments using OTP, all bank accounts linked to the customer's unique ID are displayed on the LCD screen. The customer selects one of these accounts, after which an OTP is sent to their phone. To confirm the transaction, the customer must enter the OTP using the keyboard. The complete process is illustrated in Figure1.6.

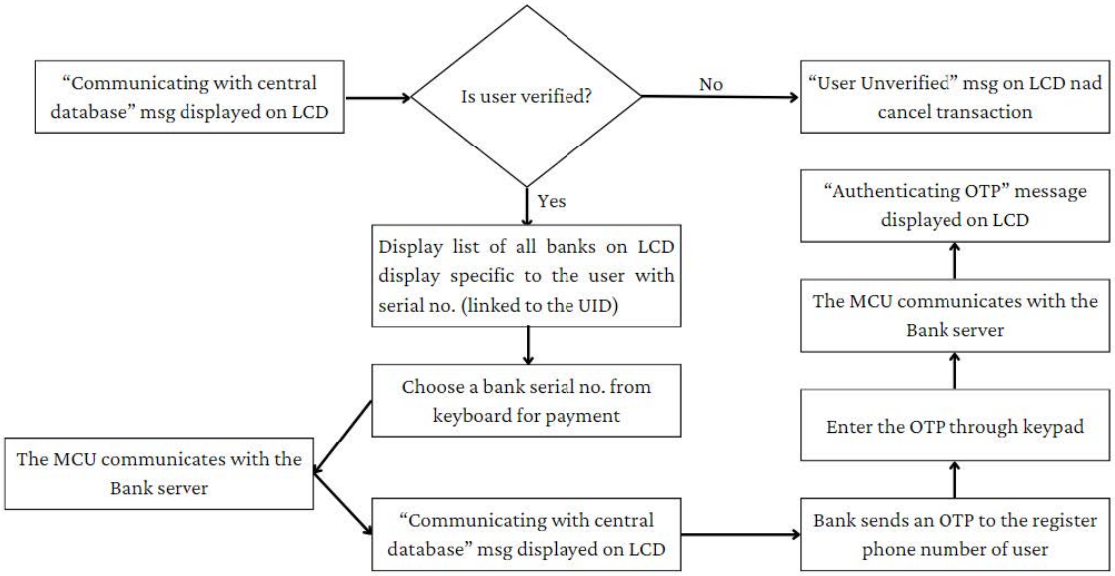


Figure 1.6: Workflow of the Model – Payment via OTP (One-Time Password)[3]

Once the OTP is entered correctly, the payment process is completed, and the customer receives a confirmation of the purchase via SMS, as shown in Figure1.7 [3].

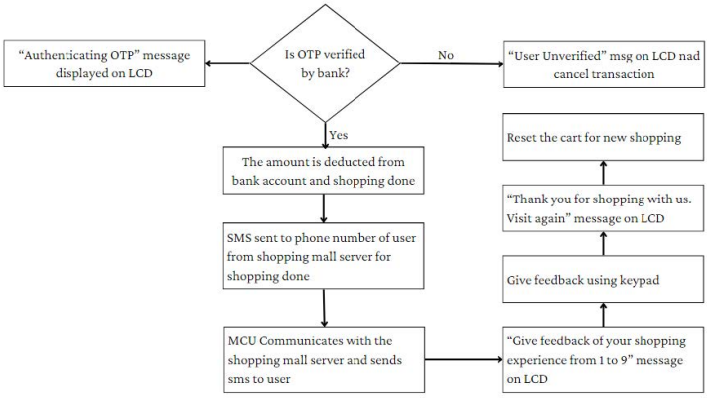


Figure 1.7: Workflow of the Model – Transaction and OTP Verification[3].

One innovative system design that supports both online and offline payment methods is the "Smart Shopping Trolley with Automated Billing". In this approach, the smart checkout system consists of a smart cart and a mobile application. Utilizing RFID technology, customers can effortlessly add items to their trolleys or remove them if they change their minds. The contents and prices of the cart can be monitored on both the LCD screen and the mobile application. What sets this system apart from others is the integration of an IR sensor, which can detect products placed in the cart that have not been scanned by RFID[22].

Another study, titled "Automatic Product Detection and Smart Billing for Shopping Using Li-Fi", also explores smart checkout systems. This approach, however, does not address the issue of unscanned items but still employs RFID technology for product recognition. Customers can view the contents and prices of their carts through a mobile application. Those opting for cash payments need to proceed to the cash register, while card payment users can complete transactions directly through the application[23].

1.4.2 Self-Checkout Systems

Another high-profile example who is seeking to offer an intelligent shopping and checkout experience is RFGo. While doing so, customers have a regular shopping experience until the point of payment. They go in, select what they wish to purchase, and bring it to the payment desk—either basket or cart-hand-carried in, or hand-carried hand in hand in their own bag. After they are on the counter to pay, the consumers wait for any kind of payment and move to the checkout area once finished. On the checkout area, passive RFID tags on all goods are identified passively automatically without taking items from baskets or bags. This is done after some 2 seconds after bill amount and a payment QR code become visible on the billing screen. The immediate payment has been made and the shopper leaves the checkout area, IR sensors detect the departure and allow the next customer to start a new transaction [24].

Another study, titled "Self-Service Checkout System for Groceries," features a self-service terminal where customers can scan the products and pay independently. The customer scans the products after buying the desired products and drives the products to the terminal. This study proposes a new payment system to overcome the shortcomings of existing self-service checkout systems. The solution is pre-loaded payment cards such as loyalty cards for all stores. This is an solution that doesn't involve plugging into banking or payment outside systems since connection will only need a local database. Payment from customers is achieved with their RFID card at point of sale terminal. For an additional level of security, the all scanned

goods will be archived to a part of which has embedded weight sensors so as to validate against known items weight as well as actual measurement weight. If the discrepancy is not present, the customer may be allowed to proceed from the payment area [25].

Unlike the afore-stated systems, another piece of research works also addresses unpacked and bar code-less produce such as vegetables and fruits. The system concerns the detection of such objects in order to properly work and implements a pseudo-labeling algorithm. The system gets trained to classify objects from top-view images relying on various sets from natural imagery to synthetically generated ones in reliance on object-based augmentation. The system thereby achieves at least 95% accuracy in classification of fruits [26].

1.4.3 Mobile Checkout Systems

Some innovative system designs in the literature enable customers to complete the entire shopping journey—including pre-purchase, check-in, product selection, checkout, and payment—through a mobile application. One such example is the "Smart Self-Checkout System for Supermarkets". To use this system, customers must first download and register on the mobile application. The app offers both in-store shopping and omnichannel options. For in-store use, customers scan a QR code at the store entrance to gain entry. They then scan the barcodes of the products they wish to purchase, with the app displaying the scanned items and their details. If customers decide not to buy a product, they can remove it directly through the app. Once all desired items are scanned, payments can be made directly through the mobile application. After completing the payment, customers present the payment confirmation to the attendant at the exit before leaving the store[4].The flowchart illustrating the offline shopping process using the mobile application is shown in Figure 1.8.

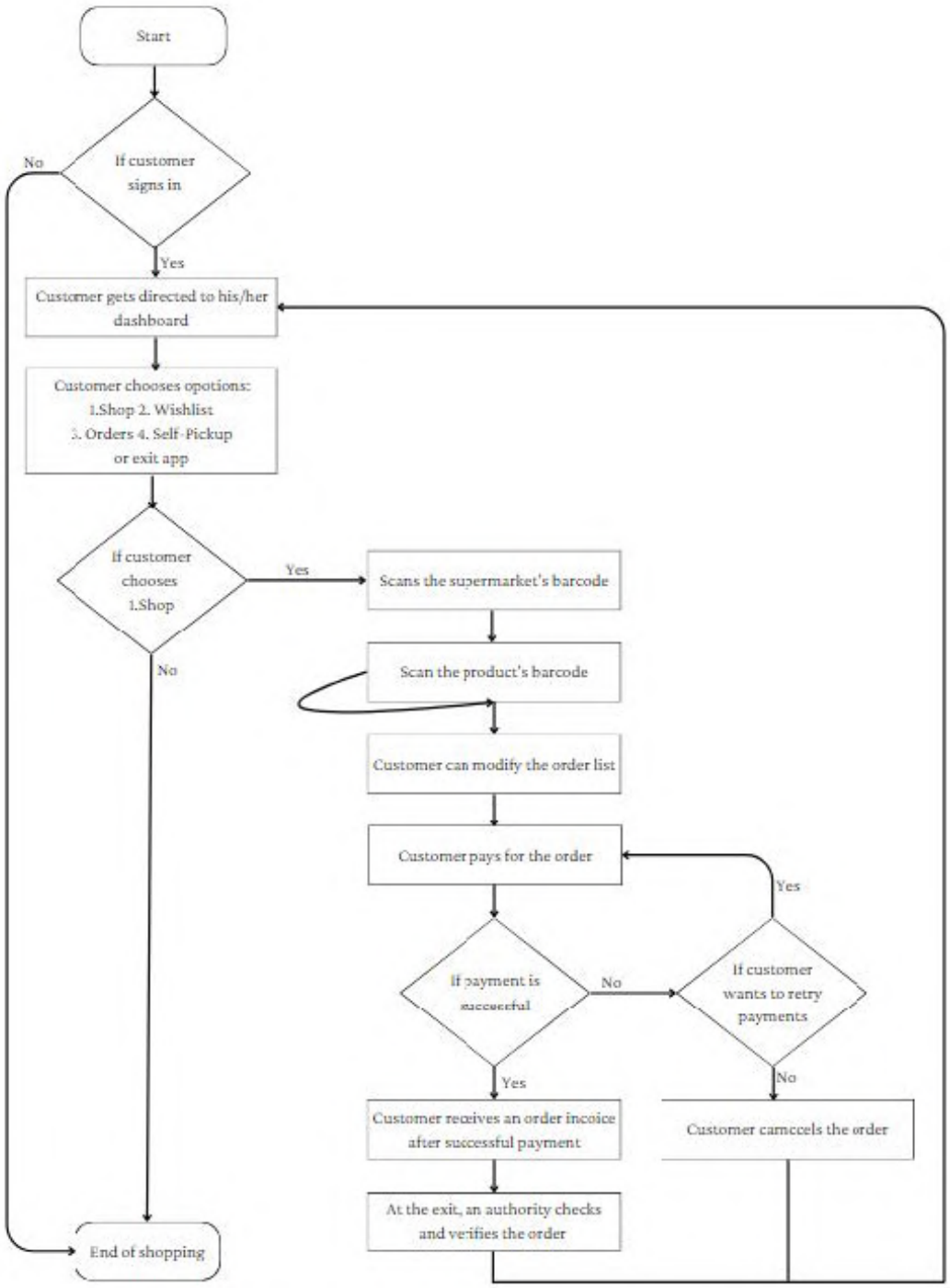


Figure 1.8: Workflow for Mobile Checkout[4]

1.4.4 Just Walk Out Systems

Amazon's "Just Walk Out" technology represents a groundbreaking innovation in the retail shopping experience by integrating advanced technologies such as artificial intelligence, machine learning, image recognition, and sensors. This system allows customers to take the items they want and leave the store without the need to stop at a checkout or payment point. To utilize this technology, customers must be identified upon entering the store. Once inside, the system tracks customers' movements

using biometric data and physical characteristics detected through sensors and cameras. As customers pick up products from the shelves, the technology automatically adds these items to their virtual cart. When the shopping is complete, customers can simply walk out, and the payment is automatically processed using their pre-registered payment method[27].

1.5 Advantages of Smart Checkout Systems

This technology offers numerous benefits for both customers and store owners. Customers, in particular, gain key advantages such as quicker service and more efficient purchase management. These benefits enhance customer satisfaction, which in turn fosters loyalty. For instance, shoppers who value faster service may be more likely to return to stores that implement smart checkout systems. Additionally, offering real-time recommendations based on customers' current baskets or past shopping habits can boost store revenue. Moreover, store owners also enjoy distinct advantages from this technology.

1.5.1 Benefits for Customers

Shopping at stores, especially supermarkets and grocery stores, is a routine activity for customers. These locations tend to be particularly crowded during peak hours and special occasions, often resulting in long queues and extended waiting times. Studies indicate that excessive waiting time is a major factor driving customers away from physical stores[4]. Smart checkout systems aim to address this issue by providing a faster checkout experience, significantly reducing or even eliminating waiting times—one of their key advantages.

Another benefit is that when customers can view their basket contents and total cost in real time on a screen or device, they can make more budget-conscious decisions and avoid unnecessary purchases [19]. Instantly knowing the total amount they will pay allows for a more efficient shopping experience.

Moreover, in systems where shopping progress is displayed on a screen, customers can receive real-time product recommendations or personalized discounts and promotions based on their current basket or previous purchases[18]. This enhances the overall shopping experience, making it more convenient and engaging.

1.5.2 Benefits for Store Owner

One of the most notable advantages for store owners is the reduced need for cashiers, leading to lower workforce requirements [18]. Cutting labor costs—one of the major expenses for businesses—allows stores to offer more competitive pricing, ultimately enhancing their market position [6].

Additionally, minimizing cashier interactions during transactions increases automation in the billing process, significantly reducing the risk of human calculation errors[19].

Another key benefit is the removal of traditional cash register areas. Replacing large checkout zones with compact kiosks frees up valuable store space, which can be repurposed for revenue-generating purposes. Optimizing this space utilization can boost a store's revenue per square meter [2].

Security is also a crucial advantage of this system. Technologies like "just walk out" or smart trolleys automatically track each item a customer selects, helping to prevent theft. In cases where a customer attempts to leave without paying, security measures such as triggering an alarm or locking cartwheels can be implemented as deterrents[18].

Lastly, the system's ability to collect real-time data enhances inventory management. Store owners can monitor sales and stock levels in real-time [18], enabling faster replenishment of out-of-stock items and better planning for supply needs.

1.6 Disadvantages of Smart Checkout Systems

One of the major drawbacks of these systems is their impact on employment, as they reduce the demand for labor. Automation replaces tasks previously performed by humans with machine-operated processes. For instance, substituting traditional cashier areas with Just Walk Out technology lowers the need for cashiers. However, these systems also create new job opportunities and increase demand for human resources in different fields. Skilled personnel are required for the development, installation, and maintenance of these technologies [28]. Additionally, employees are still needed for essential tasks such as customer assistance, product placement, and inventory management, similar to traditional store operations [29].

Another challenge posed by reduced human interaction in the shopping experience is a decline in communication. Research shows that 77% of American consumers prefer engaging with a person rather than a machine [30].

Cost is another significant disadvantage. The technology, equipment, and ongoing maintenance required for these systems involve substantial expenses.

Furthermore, products must be compatible with these systems, which can be a drawback, particularly for manufacturers [2]. Most of these systems rely on RFID technology, and adding RFID tags to all products can be both costly and time-consuming, potentially reducing overall system efficiency [4].

1.7 Issues in the Smart Checkout Systems

Many smart checkout systems incorporate innovative methods for product recognition. While many designs utilize RFID or QR code scanning, handling unpackaged items is particularly crucial for the efficient operation of these systems. Additionally, identifying products that are either missed due to system malfunctions or deliberately not scanned remains a significant challenge. Both of these issues have been widely discussed in the literature.

1.7.1 Issue of recognizing products

Various methods are employed for product recognition, with RFID technology being widely used in many designs to detect items selected by customers, especially for packaged goods. However, unpackaged products, such as fruits and vegetables, which are often packaged by customers themselves, pose a unique challenge. Ensuring that these items are recognized in the same way as packaged products is vital for the smooth operation of smart checkout systems. One study that tackles this issue is "AI-Driven Produce Management and Self-Checkout System for Supermarkets," which focuses on using deep learning to identify the size, color, and shape of fruits and vegetables. In this study, 10,000 product images were used to train a model to recognize these items. Weight sensors integrated into smart trolleys detect the weight of the foods. The system, integrated into a self-checkout setup, was tested with 50 users and showed no errors. Additionally, the dataset was split into training and testing sets, with the system achieving an accuracy of 94.8% [31].

Another related study, "PseudoAugment: Enabling Smart Checkout Adoption for New Classes Without Human Annotation," also uses deep learning to address the same challenge. This study aims to design a system integrated into self-checkout kiosks

and achieved an accuracy of 98.3% using images captured in various environments as input [26].

1.7.2 Issue of unscanned products

Unscanned products can result from both intentional theft by customers and system errors. In traditional retail, theft is a common issue, and one of the reasons it's difficult to prevent is that employees can't track every customer, their purchases, and items returned to shelves [18]. There are solutions in the literature that address this problem, such as the RFGo system. In this design, customers are required to bring all their items to the checkout area, where all products, whether in their basket or hands, are detected, ensuring that everything is scanned [24]. Another solution is the "just walk out" technology, where cameras and sensors track every item, both taken and returned, by the customer [27]. This makes it impossible for a customer to leave the store with unscanned items.

To address unscanned items caused by system errors, two studies have implemented sensor-based solutions. One study uses infrared (IR) sensors to detect if a product is added to the trolley. If an item is detected but not scanned, staff are alerted [22]. Another study employs weight sensors, comparing the total weight of the cart with the combined weight of the items in it each time an item is added or removed. If a discrepancy is detected, the system triggers an audible warning and displays a notification on the screen [19].

1.8 Adoption of Smart Checkout Systems

Various studies have explored the factors influencing customers' shopping behavior in smart stores. One such study proposed that shopping in these stores is driven by three direct effects: hedonic motivation, utilitarian motivation, and perceived ease of use. Hedonic motivation was considered to stem from factors like control, joy, curiosity, focused immersion, and temporal dissociation. In contrast, utilitarian motivation was attributed to aspects such as merchandise quality, merchandise price, location convenience, shopping speed, and product recommendations. The study concluded that hedonic motivation, utilitarian motivation, and perceived ease of use significantly impact shopping intentions in smart stores. Additionally, perceived ease of use was found to have a considerable effect on both hedonic and utilitarian motivation. While all hypotheses related to hedonic motivation proved significant, contrary to expectations, utilitarian motivation was not significantly affected by merchandise price. The study's findings are illustrated in Figure1.9 [5].

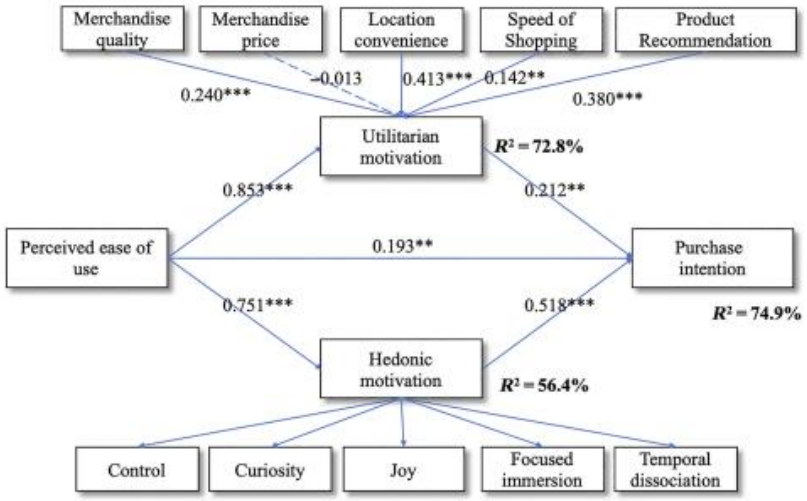


Figure 1.9: Empirical result of the study on smart store adoption[5]

Another study on this topic, titled "How Smart Technology Empowers Consumers in Smart Retail Stores? The Perspective of Technology Readiness and Situational Factors", presented hypotheses shown in Figure 1.10. The results indicated that transaction convenience did not have a statistically significant impact on purchase intention. Interestingly, merchandise price, which was previously found to have no significant effect on utilitarian motivation, was shown to directly influence purchase intention in this study. Furthermore, it was revealed that factors such as system ease, usefulness, and suitability are influenced by individuals' levels of technological readiness [6].

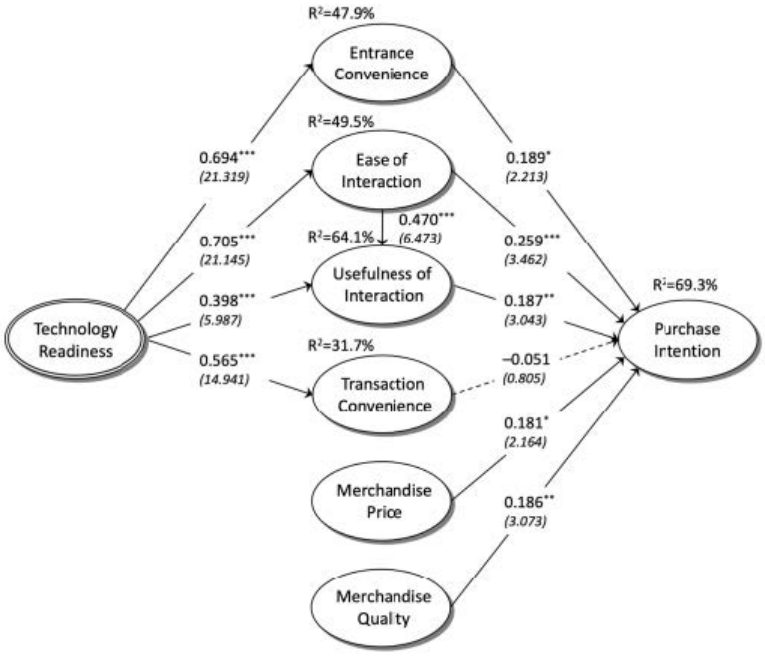


Figure 1.10: Empirical result of another study on smart store adoption[6]

In a separate study titled "Self-Checkout Behaviors at Supermarkets: Does the Technological Acceptance Model (TAM)

Predict Smart Grocery Shopping Adoption?", the focus was on adopting self-checkout systems. Unlike previous research, this study found that perceived ease of use and individuals' technological abilities did not significantly affect adoption [32].

1.9 Conclusion

In this chapter, we explained the smart checkout system, highlighting the differences between the existing system and the conceptual approach,as well as discussing its advantages, disadvantages, issues, and adoption. In the next chapter, we will introduce computer vision and explain some important terminology to help understand how the application works.

CHAPTER 2

COMPUTER VISION MODELS

2.1 Introduction

After introducing about the main ideas of smart checkout systems—which tell us about their good parts, hard parts, and tech parts. We then move to the key thing that makes modern, automatic retail work: computer vision. New steps forward in computer power, AI plans, and lots of big, marked datasets (like COCO, ImageNet) have changed how we spot and track things, letting us see retail items with great rightness right away. Not like old systems that use RFID or barcodes, systems that use vision dig into deep learning to get what's in complex visual info. They handle all sorts of product looks, wraps, and how they are set on shelves. This thesis looks into making such a system, with a focus on guessing well for retail places. We look at different build choices (like one-step vs. two-step detectors), and weigh how right they are against how fast they run.and adaptations for occlusions or lighting variations—critical considerations for deploying scalable, cost-effective solutions in dynamic store settings.

2.2 Related Work

The development of AI-based machine vision for retail self-checkout systems builds upon advancements in smart checkout technologies, IoT-based payments, and computer vision. Below is a synthesis of key studies and their contributions to the field, integrating insights from [1] and the provided thesis by [33].

2.2.1 Smart Checkout Systems

[1] analyzed the market landscape of smart checkout systems, highlighting technologies such as "just walk out" (e.g., Amazon Go) and mobile-based solutions. These systems offer advantages like reduced waiting times and improved inventory management but face challenges such as high implementation costs and concerns over job displacement due to automation. Similarly, [33] explores a vision-based automated checkout prototype, emphasizing the trade-offs between accuracy and computational efficiency. The prototype, developed in collaboration with AI Retailer Systems, leverages computer vision to track products in real-time, addressing the limitations of traditional self-scanning solutions that rely on customer honesty.

[34] investigated cashierless stores in Italy, identifying barriers to customer adoption and the need for seamless integration with existing retail infrastructures. Their findings align with [33]'s discussion on balancing automation with usability, as seen in the prototype's design to mimic a "sophisticated vending machine" for enhanced customer convenience.

2.2.2 IoT-Based Payments

[35] proposed extending PCI DSS standards to secure IoT payment systems, emphasizing encryption and biometric authentication to mitigate vulnerabilities in smart retail environments. While [33] focuses on computer vision, the integration of secure payment systems remains a critical consideration for future iterations. [36] reported a 97% increase in IoT malicious attacks from 2020 to 2022, underscoring the need for robust security measures in automated retail solutions.

2.2.3 Computer Vision for Retail

[12] introduced Mask R-CNN, an advanced instance segmentation model, which [33] evaluated alongside RetinaNet and YOLOv3 for product detection in retail settings. The thesis demonstrates that Mask R-CNN achieves high accuracy (72.3%

mAP at 1920x1080 resolution) but suffers from slower run-times, while RetinaNet offers a better balance between accuracy and speed. [37] released YOLOv8, a state-of-the-art model optimized for real-time applications, which could further enhance the prototype's performance in future work.

[33] conducted experiments with data augmentation (e.g., scaling, brightness adjustments), yielding a 20.8% improvement in mAP, reinforcing the importance of robust training datasets. The thesis also highlights challenges such as occlusions and reflections, which align with broader industry hurdles noted by [1] and [34].

The related work underscores the interdisciplinary nature of automated retail solutions, combining insights from computer vision, IoT security, and user experience design. [33]'s prototype advances the field by demonstrating the feasibility of vision-based checkout systems, while [1] and others provide a broader context for implementation challenges and opportunities. Future research could explore integrating YOLOv8 [37] for faster inference or addressing security concerns raised by [35] to create a more comprehensive solution.

2.3 Computer Vision

Computer vision is a field of computer science that focuses on enabling computers to identify and understand objects and people in images and videos. Like other types of AI, computer vision seeks to perform and automate tasks that replicate human capabilities. In this case, computer vision seeks to replicate both the way humans see, and the way humans make sense of what they see.

The range of practical applications for computer vision technology makes it a central component of many modern innovations and solutions. Computer vision can be run in the cloud or on-premises [38]

2.4 Deep Learning

Deep learning (DL), a branch of machine learning (ML), provides significant flexibility and learning capabilities by modeling the world through a nested hierarchy of concepts. In this hierarchy, complex concepts are constructed from simpler, more abstract representations. DL achieves this through its multi-layered (hidden-layer) architecture, enabling it to learn categories progressively. For example, it can identify low-level features like letters, build upon them to recognize medium-level features like words, and eventually understand high-level structures such as sentences[39].

2.5 Deep Learning for Computer Vision

Neural networks have existed since the 1970s, but recent years have seen rapid advancements in various computer vision tasks. This progress is driven by increased computing power, the availability of large publicly annotated datasets, and developments in deep neural networks, commonly known as deep learning. The introduction of AlexNet [40] marked a breakthrough in computer vision, demonstrating significantly improved accuracy in the ImageNet [41] classification task by leveraging deep neural networks.

2.6 Type of learning

2.6.1 Supervised Learning

Deep neural networks are a form of supervised learning algorithm that require labeled data for training. Due to their extensive parameter (weight) space, they need a substantial amount of training data to achieve effective generalization[33].

2.6.2 Unsupervised Learning

Unsupervised learning represents a key category of machine learning algorithms that aim to reveal the underlying structure within data without relying on external labels or supervision, unlike supervised learning. Common objectives in unsupervised learning include identifying patterns of similarity among data samples—referred to as clustering—estimating the data distribution—known as density estimation—or reducing the dimensionality of the dataset, which is typically categorized as subspace learning[42].

2.7 Single-Stage vs. Two-Stage Instance Segmentation

Instance segmentation models can broadly be categorized into two types: **single-stage methods** and **two-stage methods**. Each has distinct advantages and trade-offs based on speed, accuracy, and complexity [43, 44].

2.7.1 Single-Stage Methods

Single-stage instance segmentation models perform object detection and pixel-wise mask prediction in a single forward pass through the network.

- **Examples**: YOLACT, SOLO, SOLOv2, CenterMask
- **Pros**:
  - Faster inference speed; suitable for real-time applications
  - Simpler and end-to-end trainable architecture
- **Cons**:
  - Slightly lower accuracy, especially for small or overlapping instances
  - Less effective in complex scenes

2.7.2 Two-Stage Methods

Two-stage instance segmentation models first generate region proposals, then classify and refine them while predicting corresponding segmentation masks.

- **Examples**: Mask R-CNN, Hybrid Task Cascade (HTC)
- **Pros**:

  - Higher accuracy, especially for small objects and dense scenes
  - Better localization and mask precision

- **Cons**:
  - Slower inference speed
  - More complex training pipeline and architecture

2.7.3 Comparison Table

Table 2.1: Comparison between Single-Stage and Two-Stage Instance Segmentation Methods

Feature	**Single-Stage Methods**	**Two-Stage Methods**
Speed	Faster	Slower
Accuracy	Lower (generally)	Higher
Architecture	Simpler	More complex
Use Case	Real-time segmentation	High-precision segmentation
Training	End-to-end	Multi-step

2.8 Transfer Learning

To reduce the amount of data needed for training, transfer learning can be utilized. This technique involves using a model that has been pre-trained on data from different sources rather than the target environment where it will be deployed. By leveraging large, readily available datasets such as COCO [45], the model learns to recognize low-level features and objects. While a pre-trained model may not initially detect the specific products of interest, this issue can be addressed by retraining the final layers—known as the "heads" of the network—which are responsible for high-level semantics and object classification. This approach retains the knowledge acquired from the original COCO dataset, enabling the model to effectively detect fundamental features like edges, corners, and colors while fine-tuning the later layers for the specific object classes required.

2.8.1 Type of Transfer Learning

Inductive Transfer Learning

This scenario refers to a situation where the tasks in the source and target domains are different, and the domains themselves may or may not be related. To enhance performance on the target task, some labeled data in the target domain is required, while the knowledge gained from the source domain provides useful inductive biases. As explained in [46], depending on the availability of labeled data in the source domain, the approach can resemble either multi-task learning or self-taught learning.

A common form of inductive transfer learning involves using a deep learning model that has been pre-trained on a source domain and task, and then fine-tuning specific layers for the new task in the target domain [47]. For example, a model pre-trained on the ImageNet dataset can be adapted to recognize different object classes in a different dataset.

Another approach is multi-task learning, where several tasks are learned simultaneously from the same input data [48]. For instance, a single image can be used to train a model to perform both semantic segmentation and depth estimation at the same time.

When there is insufficient labeled data in the source domain, inductive transfer learning resembles self-taught learning [49]. In this case, although the label spaces may differ, the unlabeled data from the source can still be used to enhance performance on the target task. The source task is used to identify useful patterns in the unlabeled data, which can then support supervised learning in the target domain.

Transductive Transfer Learning

In this approach, the source and target tasks are the same or very similar, but they come from different domains. While labeled data is available in the source domain, there is no labeled data in the target domain, making this setting somewhat similar to semi-supervised learning. When the features are shared across both domains but the distribution of input data differs, this scenario falls under transductive transfer learning. It is closely related to concepts like domain adaptation[50], sample selection bias[51], and covariate shift[52]. A common example of this type of transfer learning is using synthetic images in the source domain to enhance model performance on real-world images in the target domain.

Unsupervised Transfer Learning

This scenario resembles inductive transfer learning, where the tasks differ between the source and target domains. However, in unsupervised transfer learning, the focus is specifically on solving unsupervised tasks in the target domain—such as clustering and dimensionality reduction [53, 54]. Notably, labeled data is unavailable in both the source and target domains.

2.9 State of the Art Models

As noted earlier, this field is evolving rapidly, but certain architectures have remained fundamental, undergoing continuous improvements across multiple iterations. The Instance segmentation model in this project is built upon one of these cutting-edge neural network architectures

2.9.1 Fast R-CNN

Motivation and Background

Fast R-CNN was developed to address the limitations of its predecessor, R-CNN. While R-CNN produced accurate object detection, it suffered from slow training and inference times due to its multistage pipeline: region proposal generation, feature extraction per region, and classification. Fast R-CNN simplifies this by introducing a unified architecture that allows the entire detection pipeline to be trained end-to-end, resulting in improved speed and accuracy[55].

Overall Architecture

Fast R-CNN takes the entire image and a set of object proposals as input. Instead of feeding each region proposal into the CNN separately (as in R-CNN), it first passes the whole image through several convolutional and pooling layers to generate

a shared feature map. Then, for each region of interest (RoI), a novel RoI pooling layer extracts a fixed-length feature vector from the shared feature map, ensuring efficient processing of multiple proposals[55] in Figure 2.1.

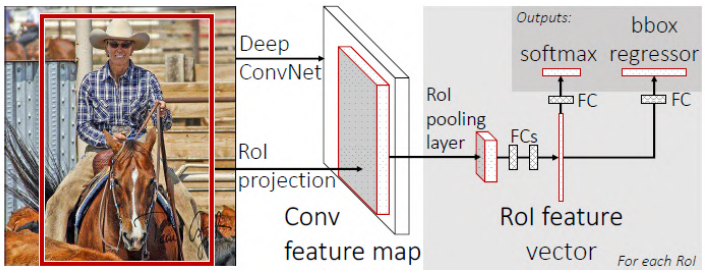


Figure 2.1: Fast R-CNN architecture

RoI Pooling Layer

The RoI pooling layer is central to Fast R-CNN's performance. It transforms variable-sized region proposals into fixed-size feature maps by dividing the proposal region into a fixed number of bins and applying max pooling within each bin. This enables the network to maintain spatial alignment and allows subsequent fully connected layers to process each proposal uniformly. The use of shared convolutional features significantly speeds up training and inference[55].

Multi-task Loss and Training

Fast R-CNN uses a multi-task loss function that jointly optimizes for classification and bounding box regression. Each RoI is assigned a class label and a bounding box target, and the network simultaneously predicts both using two sibling output layers: a softmax classifier and a bounding box regressor. This joint learning improves localization accuracy while maintaining classification performance. Training is performed using standard backpropagation and stochastic gradient descent (SGD), allowing for streamlined end-to-end optimization[55].

Performance and Advantages

Compared to R-CNN and SPPnet, Fast R-CNN achieves significantly better performance in both speed and accuracy. It eliminates the need to store features on disk and allows training to converge faster. On datasets like PASCAL VOC, it sets new state-of-the-art results at the time of its publication. Importantly, Fast R-CNN laid the groundwork for subsequent advancements like Faster R-CNN by highlighting the efficiency benefits of shared computation and end-to-end training[55].

2.9.2 Faster R-CNN

In this subsubsection, we provide a concise overview of the Faster R-CNN detection system. Originally proposed for object detection [7], Faster R-CNN is designed to detect objects in an input image by producing a list of bounding boxes with their object classes. The system operates in a two-stage process: proposal generation and classification.

The input image is passed through a 2D convolutional network (ConvNet) first to produce a 2D feature map. A second 2D ConvNet, referred to as the Region Proposal Network (RPN), produces a sparse set of scale-invariant class-agnostic region proposals by predicting anchor boxes of various scales centered at every pixel of the feature map. The borders of the proposals are then refined after regression.

In the second stage, regional proposal features of each are combined into a fixed-size feature map through RoI pooling [56]. Object class probabilities for each class are predicted by a DNN classifier together with bounding box regression. Fig. 1 (left) shows the end-to-end pipeline. Training typically alternates between the two stages iteratively [7].

Faster R-CNN can be naturally extended to temporal action localization [8, 9, 11].Object detection is concerned with detecting 2D spatial locations, whereas temporal action localization is concerned with detecting 1D temporal intervals and their endpoints.Temporal action localization is therefore a 1D variant of object detection.The adapted Faster R-CNN process for temporal action localization is illustrated in 2.2 (right).As is the case with object detection, so is the same with this, a two-stage process.The first stage entails a series of frames passed through to emit a 1D feature map, usually through a 2D or 3D ConvNet.This 1D feature map is then passed through by a 1D ConvNet to emit scale-varying anchor segments for each temporal position and to sharpen their boundaries.This emits a sparse collection of class-agnostic segment proposals.Step two is, for each proposed segment, calculating action class probabilities and then modifying the boundaries of the segments.It does this through a 1D RoI pooling operation referred to as "SoI pooling" followed by a DNN classifier.

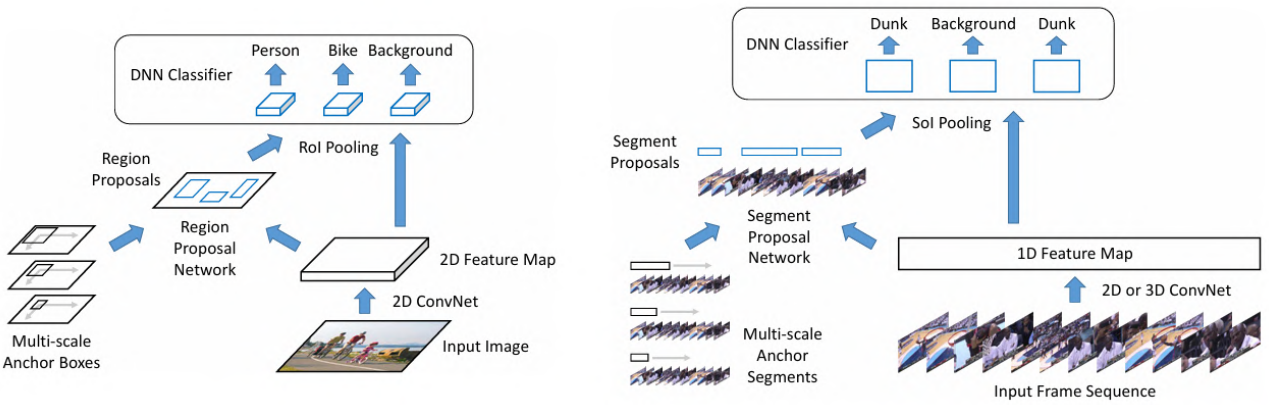


Figure 2.2: A comparison between the Faster R-CNN architecture for image-based object detection [7] (left) and its adaptation for temporal action localization in videos [8, 9, 10, 11] (right). Temporal action localization can be interpreted as the one-dimensional equivalent of the object detection task.

2.9.3 Mask R-CNN

The field of computer vision has seen remarkable advancement in object detection and semantic segmentation in a relatively short span of time.The advancement has been spearheaded specifically by robust baseline architectures like Fast/Faster R-CNN [57, 58] for object detection and Fully Convolutional Networks (FCN) [59] for semantic segmentation.The architectures are simple to understand, offer robustness and flexibility, and enable efficient training and inference.In this work, our aim is to devise an equally effective framework for instance segmentation.Instance segmentation is a unique task in the way that it not only demands precise object detection but precise boundary definition of all objects.It is a blend of the goals of object detection—object detection and localization via bounding boxes—and semantic segmentation—pixel-wise labeling.Our approach, Mask R-CNN, extends Faster R-CNN [58] by introducing a parallel mask prediction branch to the existing classification and bounding box regression branches (Figure 2.3).The mask branch is a light, fully convolutional network (FCN) computed at each RoI, pixel-to-pixel predicting masks.Mask R-CNN is straightforward to train and apply in the

Faster R-CNN framework with architectural flexibility and little computational cost and fast experimentation.

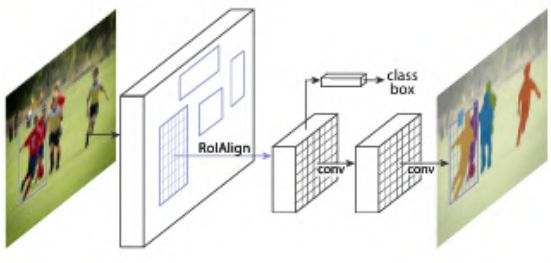


Figure 2.3: An instance segmentation framework based on Mask R-CNN [12].

Though Mask R-CNN is perhaps a natural extension of Faster R-CNN, efficient performance is only possible through thoughtful masking branch design.In a departure, Faster R-CNN had not been designed before to create pixel-to-pixel level equivalence of input and output.A key issue is RoIPool operation [57, 60], quantization in space while extracting feature.To overcome this, we present RoIAlign, which is a quantization-free layer that attains spatial precision with no penalty.Although tiny, RoIAlign provides a gigantic accuracy boost in masks—10

By default configurations, Mask R-CNN achieves superior than all previous single-model state-of-the-art performance on the COCO instance segmentation task [61], and superior than the top-scoring 2016 competition submissions. It also does well on the COCO object detection task. We experiment with some simple configurations in our ablations, checking the method's robustness and probing key design points.

Our models converge at approximately 200ms per frame on a GPU, and a single 8-GPU one- or two-day training on the COCO dataset. The sacrifice of speed, flexibility, and accuracy of the framework will render it suitable and enable future instance segmentation research.

Finally, we illustrate the generality of Mask R-CNN to human pose estimation on the COCO keypoint dataset [61]. By defining each keypoint as a one-hot binary mask, Mask R-CNN is able to learn instance-specific pose detection with negligible tuning. With no special tuning, it surpasses the winner of the 2016 COCO keypoint challenge and is run at 5 fps. In summary, Mask R-CNN is a general instance-level recognition system and can be effortlessly applied to difficult tasks in Figure 2.4.

Figure 2.4: Results of Mask R-CNN on the COCO test set. Using a ResNet-101 backbone [13], the model achieves a mask Average Precision (AP) of 35.7 and operates at 5 frames per second. The visualizations display segmentation masks in color, along with bounding boxes, predicted categories, and confidence scores.

2.10 Object Detection vs Semantic Segmentation vs Instance Segmentation

2.10.1 Object Detection

Object detection is the computer vision task of automatically detecting a specific object in an input image. It provides the object's location and class by either a bounding box or a pixel-precise mask in coordinates along with the corresponding object class.

Traditional object detection techniques rely on detection of a salient point, edge, and vertex—i.e., features—of an image. Features are employed to describe objects, e.g., a human face. Feature detection is generally done by the convolutional filter over the object image. Historically, hand-designed convolutional kernels were utilized for detection of individual features, which was computationally expensive as well as highly subjective to human capability[33].

2.10.2 Semantic Segementation

Semantic segmentation is also one of the hottest topics for quite a number of years. Before the re-birth of deep learning, classic approaches were virtually all hand-crafted feature based such as HOG [62] and SIFT [63]. Since the re-birth of deep learning, particularly with the introduction of Fully Convolutional Networks (FCN) [59], deep learning methods prevail semantic segmentation.

Similar to FCN, some techniques, i.e., U-Net [64], RefineNet [65], and SegNet [66], utilize an encoder-decoder network and take a "convolution-downsample" strategy for the encoder. It downsamples spatially but upsamples channels on features.

DeepLab was introduced by Chen et al. [67] and takes dilated convolution in place of the normal downsampling. By gradually increasing the dilation rate, the approach expands the receptive field with the same original tensor size but without the loss of spatial resolution in terms of higher computational cost. For instance, while a regular ResNet reduces an image size to 1/32, a dilated ResNet reduces it to merely 1/4. DeepLab [67], PSPNet [68], and EncNet [69] are cost-effective models

using dilated CNNs. But although they have higher spatial resolution, they may not be the best choice to use in real-time since they are costly to compute.

2.10.3 Instance Segmentation

Instance segmentation may be thought of as a combination of object detection and semantic segmentation. For example, in [70], the authors introduced a model that unites pixel, segment, and object-level information. This is because instance segmentation uses object detection approaches' ability to separate different instances but employs pixel-wise prediction capability of semantic segmentation to delineate object boundaries. Therefore, development in instance segmentation is indirectly impacted by that of object detection and semantic segmentation. Maybe one of the most significant advances in object detection has been Region-based CNN (R-CNN) [71]. It uses a region proposal algorithm to generate a dense set of object candidates from an image.

Features are extracted for each region by a CNN, followed by classification of the resulting feature vectors. There are several instance segmentation algorithms grounded on this background. Specifically, in [72, 73], the approach employs multiscale combinatorial grouping [74] for generating region proposals, which get refined. In [72], non-maximum suppression is applied over redundant candidates, while segmentation is obtained by fusing CNN-based coarse information with superpixels of an image. In [73], fine-tuning for segmentation is obtained using exemplar-based shape prediction as well as graph-cut algorithms. Despite the fact that these methods have reached state-of-the-art performance, they share one disadvantage: they are comprised of many independently trained modules.

Semantic segmentation has also advanced significantly, beginning with [75], where they introduced fully convolutional networks for pixel-level prediction. Subsequent work, e.g., [76, 77], enhanced object boundary with Conditional Random Fields (CRFs). Subsequent to these works, [78] introduced an instance segmentation model for predicting pixel-level instance locations and employing a clustering method as a post-processing. The number of clusters (instances) is parameterized by another CNN. However, one disadvantage of this approach is that it will not directly optimize instance segmentation accuracy but instead employs a surrogate loss function over pixel-to-object center distances. One of the major challenges for instance segmentation is that no inherent order must be adhered to through which instances of an image must be segmented (e.g., which object segment first).

Several papers [79, 80] have approached this problem using depth information to disentangle instances. It does this through a Markov Random Field on a CNN in attempting to encode occlusions between objects explicitly. While good at handling occlusion, it fails when objects have the same depth, such as a line of runners standing next to each other.

Unlike other methods, we present a new paradigm for instance segmentation which learns to segment instances in an iterative manner. Our approach allows the model to make dynamic choices about the segmentation order of every image, rather than relying on pre-calculated heuristics.

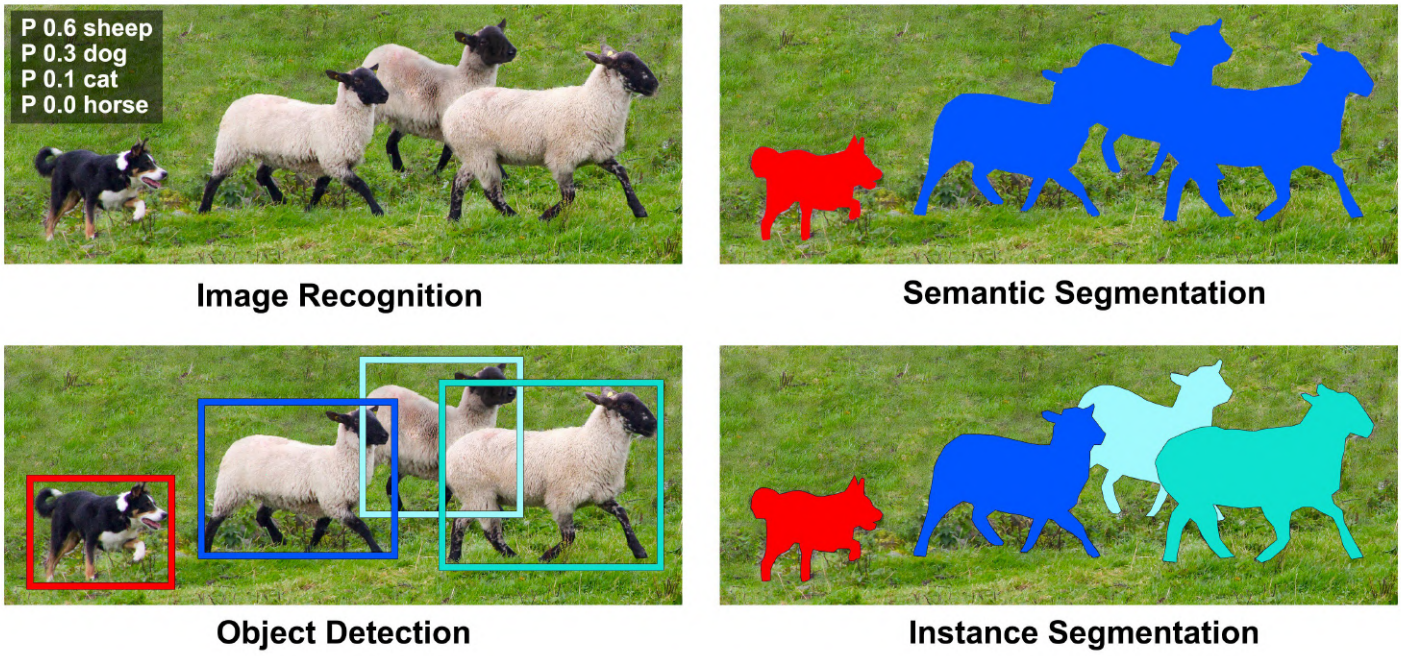


Figure 2.5: Difference between Object Detection and Semantic Segmentation Instance Segmentation [14]

2.11 Benchmarking and evaluation metrics

There are some metrics through which an object detection model can be assessed, depending on the requirement and constraints relevant to the problem. More in general accuracy always is desired but that is not always achievable along with higher cost. There is a need then to balance the accuracy with the efficiency. Having this in perspective, the models will be benchmarked on the accuracy, stability, and running time performance while executed online [33].

2.11.1 Mean average precision (MAP)

Precision is the number of accurately predicted true positives divided by the number of true positives. Average Precision (AP) is precision for all samples of a certain class in the test set. Mean Average Precision (mAP) is mean of AP for all object classes in the model[33].

2.11.2 Precision

Precision is the proportion of true positive predictions out of all positive predictions made by the model[81].It is mathematically defined as:

$$Precision = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}} \tag{2.1}$$

2.11.3 Recall

Recall is the proportion of true positive predictions out of all actual positive cases in the dataset[81].It is mathematically defined as:

$$Recall = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \tag{2.2}$$

2.11.4 Run-time

Run-time performance is of hardly any concern for object detection researchers, for accuracy is generally a primary measure of performance.However, it does matter in real deployments.The model has to run safely online and to withstand tampering with the system as much as possible.Though the run-time requirements are not as strict as they would be in the autonomous vehicle space—where cameras are usually operated at at least 30 FPS [82]—it is still very important in the retail environment.Though delay or reduced speed is not life or death, having the model be robust and system responsiveness counts nevertheless.

2.12 FPS (Frames Per Second) requirement

To estimate our system's required FPS, several different levels of FPS were attempted offline.The testing involved watching video footage with objects appearing and disappearing in groups at different speeds, including very fast movement to attempt to push the system.The primary intention was to identify the hand that goes into the cabinet, and this would be a measure of system performance as it would occur automatically if a customer does take something out.Observe that a low-complexity solution, e.g., just enumerating the products within each frame, would reduce the need for high FPS.But that would require the entire exposure of all products on the shelf at once, something that is unlikely to be true under actual use given camera placements and number of products on the shelf [33].

2.13 Model optimization

Having a pre-existing architecture based on state-of-the-art models has advantages since they already proved to perform well on generalized research benchmarks.But since the models are general models, the architecture and parameters of the model will need to be tuned to fit our specific application and constraints [33].

2.14 Conclusion

we explained the fundamental and necessary concepts of computer vision and deep learning such as various types of transfer learning and object detection in contrast with semantic and instance segmentation methods.Systematic approaches to model comparison of performance are explained.The following chapter will be kept in store for experimental use of such concepts within tests and experiments using real data.

CHAPTER 3

MODELING AND IMPLEMENTATION

3.1 Introduction

Having got the basics of computer vision and deep learning, the chapter further applies these technologies to the Smart Checkout System. Having learned it all from the above theory, we now proceed to execute the project practically from now onwards. This chapter explains the methodology that has been adopted to gather the Smart Checkout System. It contains information step by step about how data gathering was achieved, how the system was implemented, and how the efficiency of the system was determined. The methodology was determined with the intent that the development phase and test phase were systemic, reproducible, and were also in congruence with the project as a whole aim. Emphasis is given on data acquisition approaches, system design architecture, and integration methods such as computer vision and deep learning models. Used tools, methods, and data sets throughout the project life cycle are also documented, as an introduction to the following analysis and system performance evaluation.

3.2 Data Collection

For training the instance segmentation model used in the Smart Checkout System, proprietary data were collected. 1525 product images were captured by using a personal smartphone camera. Figure 3.1 shows images exhibiting a large variety of general retail products in various lighting conditions, angles, and backgrounds for convenient different aspects of improvement in generalization. Its manual acquisition gave direct control over image quality and annotation, which consisted of the essence of the system's capability for product identification

Figure 3.1: Example dataset images captured in a retail environment using a smartphone camera. The photos depict different product arrangements under varying lighting conditions

3.3 Labeling

All of the images in the dataset are annotated with detailed information regarding the location, size, and object class of each object within them.The annotations are present in the form of pixel-wise segmentation masks, which give precise object boundaries.The masks can be easily converted to bounding box labels, depending on the format in which the output from the model is to be obtained.Representativehexamples from the dataset are presented in**??**.

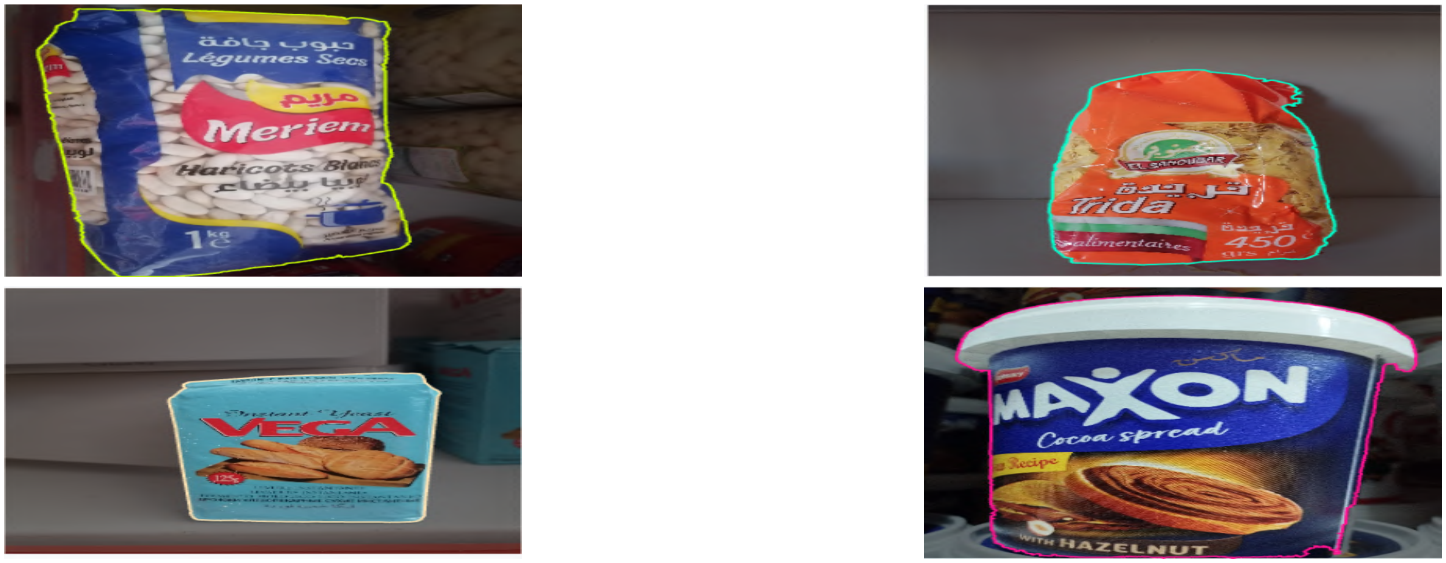


Figure 3.2: Representative samples from the labeled dataset, collected in-store using mobile photography to ensure real-world variability

Dataset Split

- **Train Set:** 70% (1068 Images)
- **Validation Set:**20% (305 Images)
- **Test Set:** 10% (152 Images)

Preprocessing

- **Auto-Orient:** Applied
- **Resize:** Stretch to 640×640

3.4 Data augmentation

In general, the more training data, the better.Data augmentation was used to expand the dataset by generating transformed copies of the original images.Best practices in augmentation were discovered by testing what are the best practices in augmentation by training a model on augmented datasets using various types of methods.The best types of augmentations that

provided better model performance were selected to be applied on the final prototype.Hyperparameters of data augmentation used in our experiments are listed in Table 3.1

Table 3.1: Data Augmentation Parameters

Parameter	**Value**
Outputs per training example	3
Flip	Horizontal
Crop	0% Minimum Zoom, 20% Maximum Zoom
Rotation	Between $-15°$ and $+15°$
Shear	$\pm 10°$ Horizontal, $\pm 10°$ Vertical
Brightness	Between -15% and $+15\%$

3.5 Model Implementation

3.5.1 YOLO

"You Only Look Once" or "YOLO" is a novel technique for object detection inside an image in real time and bounding the objects inside a box. Being named so, it simply provides the image just once to the CNN algorithm in order to obtain the output. Even though R-CNN shares the same architecture as YOLO, YOLO is a lot faster than faster R-CNN since it is simpler with respect to design [83] and we are using 2 versions of yolo version 8 and 11.

3.5.2 Yolov8

Released by Ultralytics on January 10th, 2023, YOLOv8 offers state-of-the-art accuracy and speed performance. Building on the gains made in earlier YOLO releases, it introduces new features and optimizations that position it to be most appropriately used to perform a variety of object detection tasks in a vast range of applications[37].

Architecture

YOLOv8 combines the most recent developments in neural network architecture and training techniques into the high-quality heritage of its predecessors in the YOLO family. Like its predecessors, YOLOv8 is able to achieve a balance between speed and accuracy by combining the object localization and classification tasks into one end-to-end differentiable neural network architecture [15].

YOLOv8 architecture relies on three things: The backbone YOLOv8 utilizes a cutting-edge convolutional neural network (CNN) backbone for extracting information across different scales from input images.The backbone can be any new variant of CSPDarknet or any other good architecture and is utilized for extracting hierarchical feature maps with high-level semantic features along with low-level texture information that plays an important role in precise object detection.With depthwise separable convolutions or some other high-capacity blocks utilized to reduce computational cost without sacrificing representational capacity, the backbone is optimized both for speed and accuracy.[15].

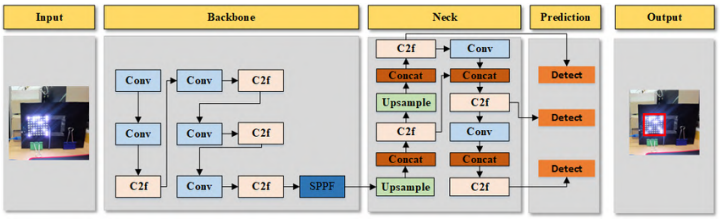


Figure 3.3: architecture yolov8 in case object detection [15]

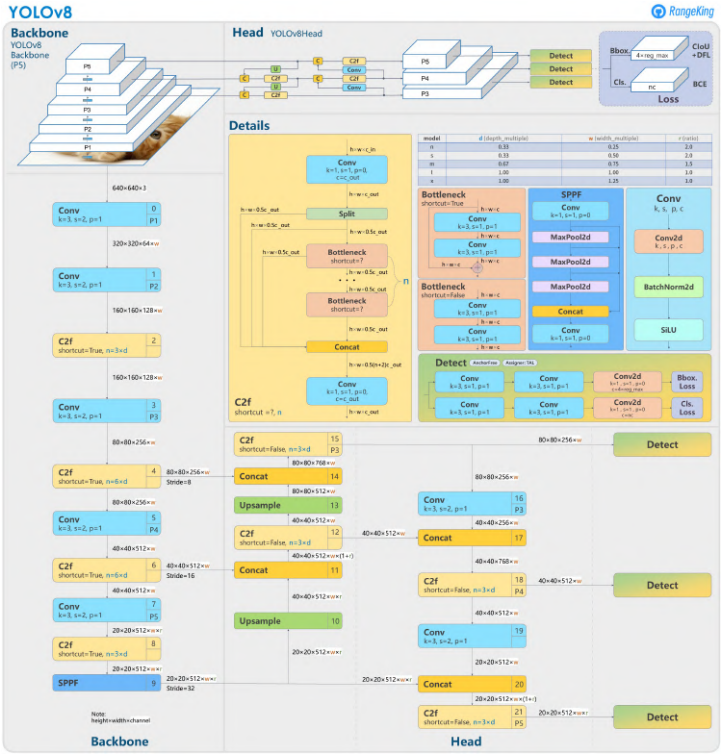


Figure 3.4: Model Structure of Yolov8 [15]

The neckThe backbone of YOLOv8 retains the multi-scale features, which are also merged and thinned by the neck module.It is comprised of an improved Path Aggregation Network (PANet) in an attempt to hone information exchange across feature levels of various sizes.Multi-scale feature fusion becomes highly important in multi-scale object detection across various sizes.For further optimizing computational resources as well as memory consumption, the novel PANet topology of YOLOv8 probably involves reconfiguring the initial PANet[15].

3.5.3 Yolov11

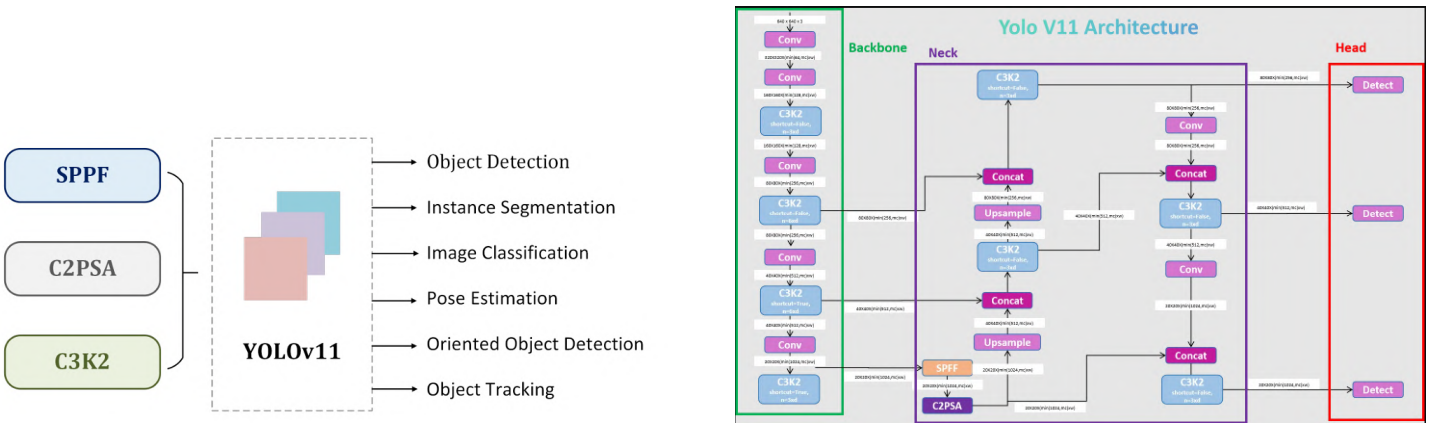
YOLO11 is the newest addition to the Ultralytics YOLO family of real-time object detectors and the new standard for accuracy, performance, and efficiency. Its stable foundation built by its predecessors, YOLO11 boasts major architecture and training methodology changes that accompany it, and it's a strong and capable solution for most computer vision applications[84].

Architecture

The object detection performance, and other computer vision responsibilities in YOLOv11, is greatly enhanced by there being a wide range of architectural improvements. The new C3k2 block, which replaces the utilization of the C2f block in the model's backbone, is efficient because of the enhanced computation of the data with lower kernel sizes. New C2PSA (Cross Stage Partial with Spatial Attention) block, enhanced for improved spatial attention for more attention to important parts in the image, is incorporated in the backbone with maintaining Spatial Pyramid Pooling - Fast (SPPF) block. These enhancements enhance more efficient feature extraction at multiple scales.

C3k2 block, that replaces the older C2f block, enhances the speed and performance of the YOLOv11 neck. It enhances the upsampling and feature aggregate process of the neck. The neck also considers spatial attention through the C2PSA module so that the model can focus on informative areas of the image. It is especially suited for identifying partially occluded or small objects. Due to such improvements, YOLOv11 is more efficiently and accurately able to handle complex visual input.

The head part of YOLOv11 provides adaptability in its c3k parameter for various levels of feature extraction complexity by extracting features from different depths of feature maps via multiple C3k2 blocks. End convolutional layers which are responsible for executing generation of bounding box coordinate coordinates and classification are followed by CBS layers responsible for stabilizing and refining features. With the advantage of task assistance such as posture prediction, instance segmentation, and also by virtue of these architecture improvements, YOLOv11 functions efficiently as well as significantly strong in handling tasks related to computer vision in real time.[85].



(a) Architecture Overview [85]

(b) Detailed Components [86]

Figure 3.5: YOLOv11 Architecture

3.6 Transfer Learning COCOn-seg

The COCO-Seg dataset is a variant of the original COCO (Common Objects in Context) dataset, which has been created in particular for object instance segmentation research.While it works with the same image set as COCO, it contains more detailed and comprehensive segmentation annotation.Due to this, it is an essential data source for researchers and developers of instance segmentation, particularly in training Ultralytics YOLO models.[87].

3.7 Metric evaluation

3.7.1 MAP50

Mean average precision (mAP) is computed with an Intersection over Union (IoU) threshold of 0.50 to approximate the accuracy of the model by preferring mostly "easier" detection tasks[88].

3.7.2 MAP50-95

The mean average precision at IoU thresholds 0.50:0.95 gave an overall estimate of performance over many detection tasks.[88].

3.7.3 Bounding Box Loss

Box loss is a measure of how well the model's predicted bounding boxes approximate the real objects in an image.Picture drawing a box around an object—box loss is how far your drawing is from the right one.It's how the model computes and adjusts its predictions to make the boxes as close as possible[89].

3.7.4 Objectness Loss

Objectness loss predicts how likely an object is to be present in a region of interest/Objectness gives high objectness prediction if the region is very likely to contain an object[90].

3.7.5 Classification Loss

Classification loss is used to estimate the model's ability to properly predict each object's class.It approximates the variance between predicted class probabilities and actual class labels, typically with loss functions such as Binary Cross Entropy (BCE) or Multi-Class Cross Entropy[91].

3.7.6 DFL Loss

Distributed Focal Loss (DFL) is a loss function that is used to emphasize object Much Detection performance gain.DFL focuses more on the difficult-to-detect instances compared to the standard loss functions.By giving more importance to hard cases, it allows the model to learn the most from hard cases, and this leads to better accuracy with time[92].

3.8 Model Training Evaluation

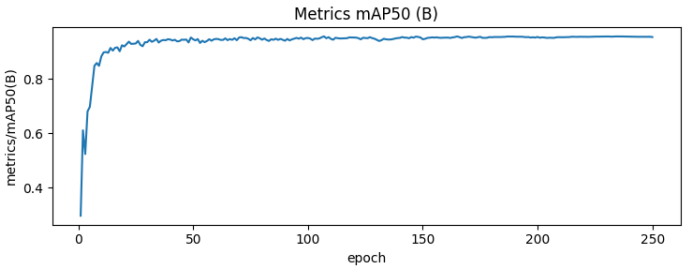
3.8.1 Yolov8

Overall Performance Analysis

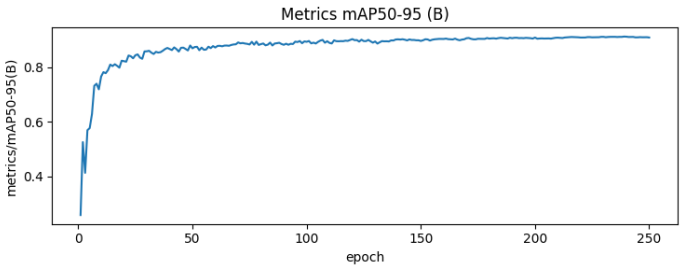
YOLOv8 was trained for over 250 epochs with comprehensive testing of core performance measures.The mean Average Precision at IoU 0.5 (mAP50) significantly improved from roughly an initial measure of 0.4 to a final measure of 0.95, a good indicator of localization strength.The more rigorous mAP50-95 measure also saw similar improvement from 0.4 to 0.9, a good indicator of model strength to maintain detection accuracy under varying IoU thresholds. Accuracy levels were

maintained at high levels during training phases between 0.8 and 0.9, and recall increased significantly to a value of 0.9 from the original 0.3.This is proof of suppression of false negatives with higher confidence in prediction.Convergence was observed of the training being slow for all objectives without grand oscillations, an indication of optimal optimization of the fundamental detection abilities of the model.

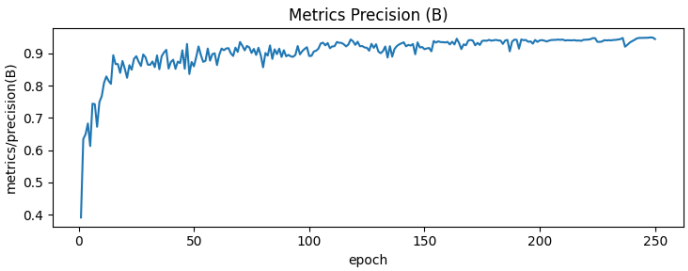
Detection Accuracy Metrics



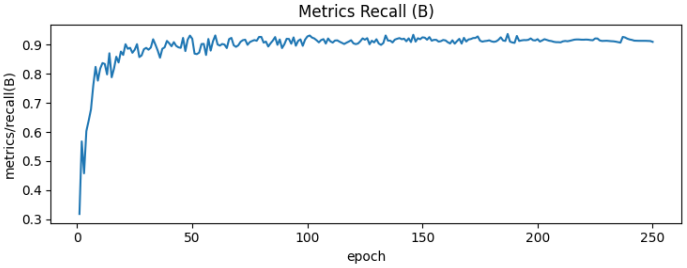
(a) mAP50



(b) mAP50-95



(c) Precision

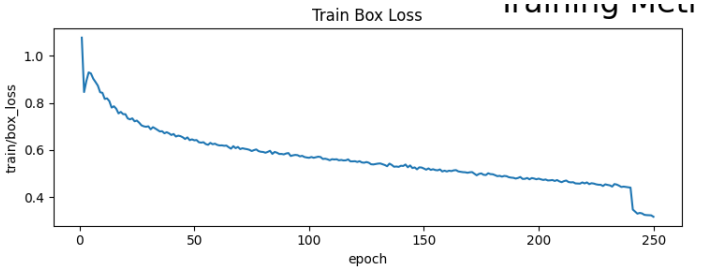


(d) Recall

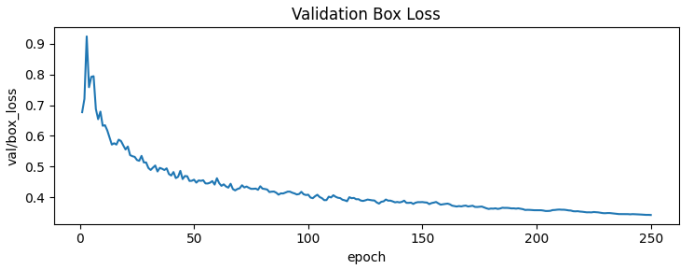
The metrics given show the performance of the model on the main evaluation metrics during training. The mAP50 (B) metric has reached an adequate terminal value of 0.95, showing stable detection accuracy with IoU threshold at 50%. The mAP50-95 (B) metric has reached an even higher value of 0.9, showing improved performance for various stricter thresholds of IoU (50-95%). This extremely high mAP50-95 value tells us that not only is the model very good at overall detection, but also localization, compared to most regular object detection models whose mAP50-95 is less than mAP50. Accuracy (B) measure was consistently high (0.8-0.9) during training, showing perfect suppression of false positives. In sync, Recall (B) also improved steadily, from 0.3 to 0.9 at epoch 250, showing enhanced ability of the model in detecting true positives. Precision and recall also approached 0.9 in subsequent epochs, settling at optimal ratio fit for real-world usage. Consistency of these values across epochs speaks highly for proper and consistent learning dynamics.

The extremely high mAP50-95 with good recall and precision indicates that the model in training is very well tuned with the ability of excellent detection and bounding box localization. Such a performance is reflective of a quality dataset, powerful augmentation methods, or ultra-tightly-tuned architecture. More work can confirm whether such good generalization to other objects of different sizes and in complex scenarios exists or not.

Loss Function Analysis

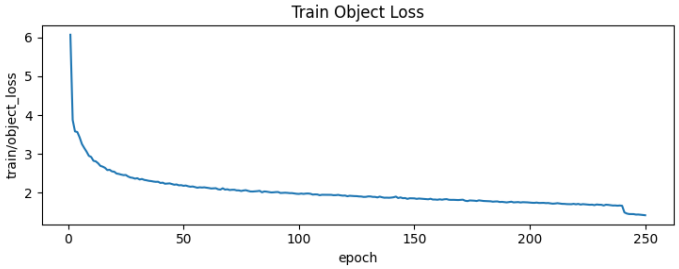


(a) Train Box Loss

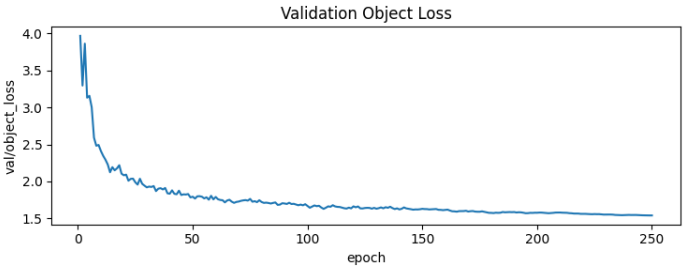


(b) Validation Box Loss

Training box loss decreased progressively, from 1.2 to 0.2, which shows effective object localization learning. Progressive decrease without considerable variance is a sign of effective anchor box parameter tuning. Validation box loss also followed in the same direction, decreasing from 0.7 to 0.2 without any overfit, the low end values reflecting high IoU performance.

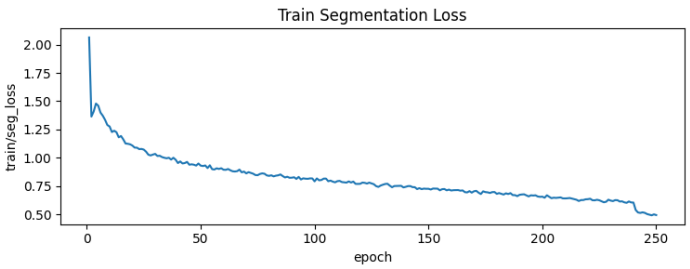


(a) Train Object Loss

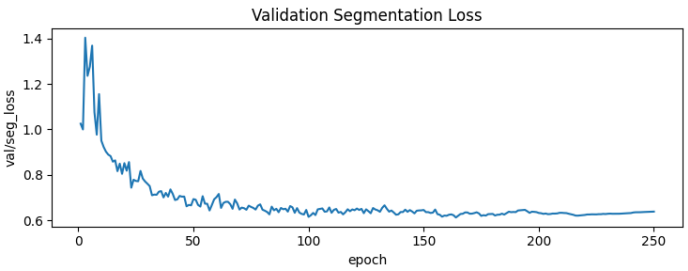


(b) Validation Object Loss

Object loss curves show significant training dynamics: while validation loss shows a clear 66% drop (4.0 → 1.5), training loss data is truncated, seemingly replicating logging artifacts.This drop is characteristic of the model learning to make foreground objects vs. background distinctions more and more, but the ultimate validation value (1.5) suggests ongoing struggle with hard negatives.The consistent decrease without recovery reflects proper confidence calibration but requires more investigation of the data distribution mismatch across train-validation gap.Surprisingly, validation trend confirms the growing credibility of the model in object existence detection across epochs.



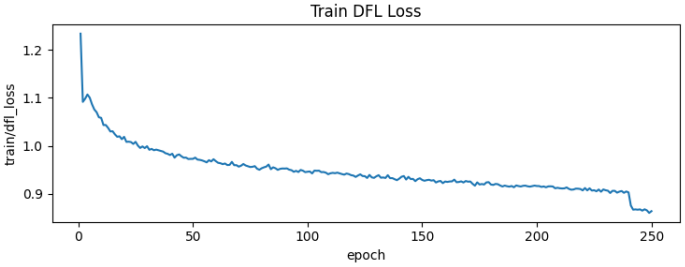
(a) Train Segmentation Loss



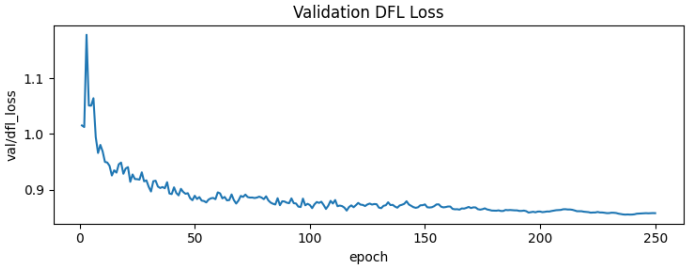
(b) Validation Segmentation Loss

Training segment loss (blue) starts at 2.0, decreases stepwise, and ends at approximately 0.5, reflecting good optimization.Validation loss (orange) starts at approximately 1.0, crosses 1.4, then decreases to 0.65, reflecting early generalization issues but subsequent improvement.Graph divergence reflects potential overfitting earlier, but the final drop in validation

reflects stabilization later.Unexpectedly, validation loss concludes lower than training, which can be an indicator of data leakage or validation set too small.Apart from both the metrics, it is still needed to test the reliability of the model.



(a) Train DFL Loss



(b) Validation DFL Loss

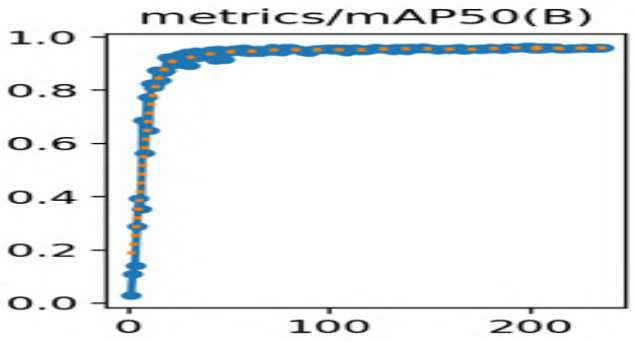
Training DFL loss (blue) always declined from 1.2 to 0.9, suggesting improving estimation precision of bounding box distributions. Validation DFL loss initially was also greater at 2.5, likely indicative of initial difficulty generalizing to test data, but then immediately fell to 0.3, surpassing the training loss in later epochs. This sudden dip can either indicate over-adaptation in the late stages or potential issues like data leakage or an extremely small validation set. The consistent discrepancy between validation loss and training loss in the early stage of training is indicative of label quality or model architecture inspections.

3.8.2 Yolov11

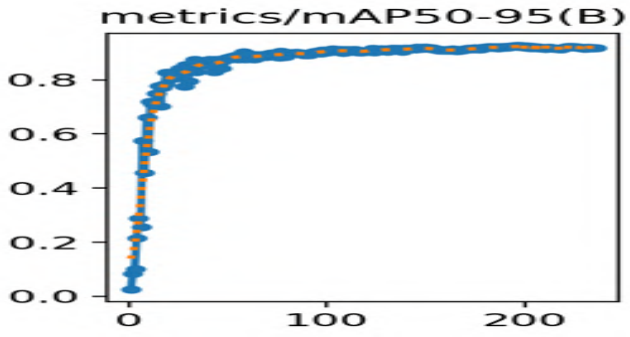
Overall Performance Analysis

The YOLOv11 model underwent comprehensive training over 250 epochs, with its performance rigorously evaluated through key metrics including mean Average Precision (mAP) and various loss functions (object loss, box loss, and class loss). The results demonstrate substantial improvements in both detection accuracy and model optimization, as visually evidenced in Figures 3.12c, 3.13c, 3.14a, and 3.11e. These improvements reflect the effectiveness of the training methodology and architectural choices implemented in YOLOv11.

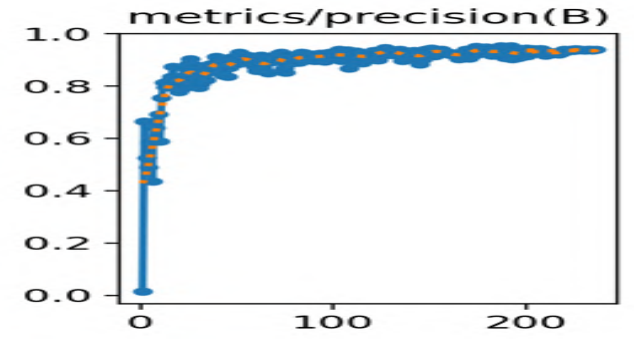
Detection Accuracy



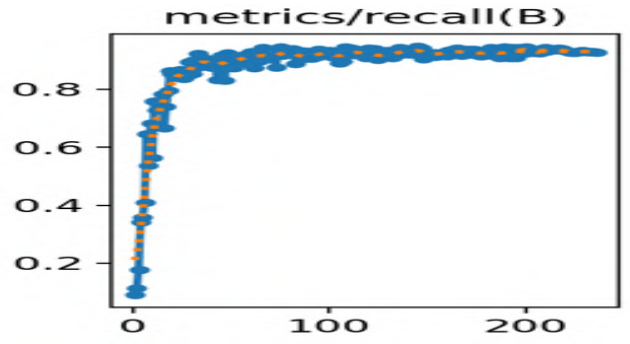
(a) mAP50



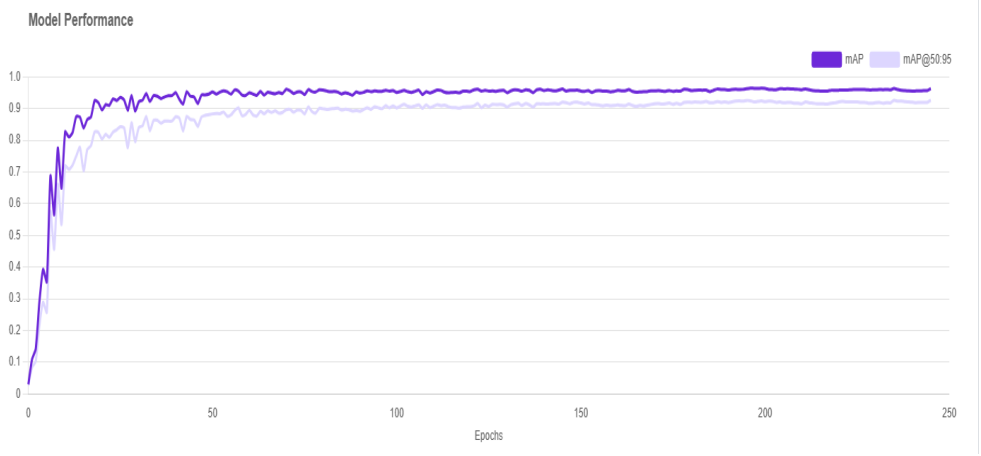
(b) mAP50-95



(c) Precision



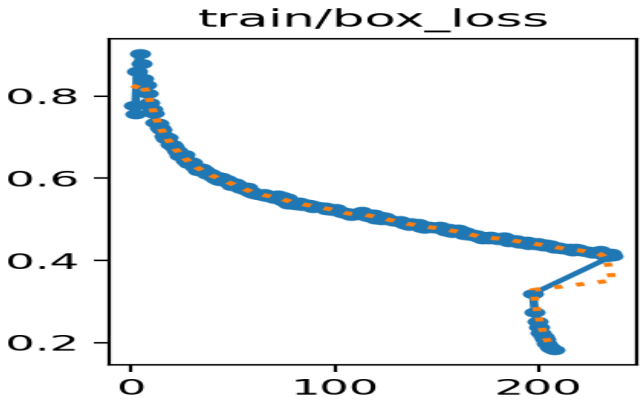
(d) Recall



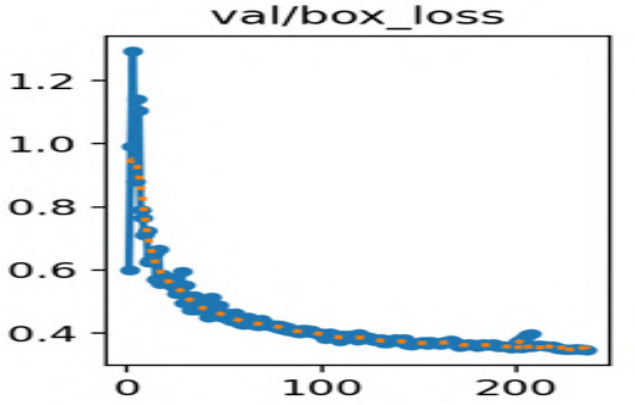
(e) Map50 and Map50-95 from roboflow

The mAP50 metric demonstrated remarkable improvement from 0.1 to 0.97, indicating exceptional localization capability at IoU threshold 0.5. The more stringent mAP50-95 reached from 0.1 to 0.91, confirming robust performance across varying detection difficulties. Precision and recall showed parallel growth from 0.2 to 0.95 and 0.1 to 0.91 in order, reflecting balanced reduction of both false positives and negatives. The convergence of all metrics suggests optimal learning dynamics without over-specialization.

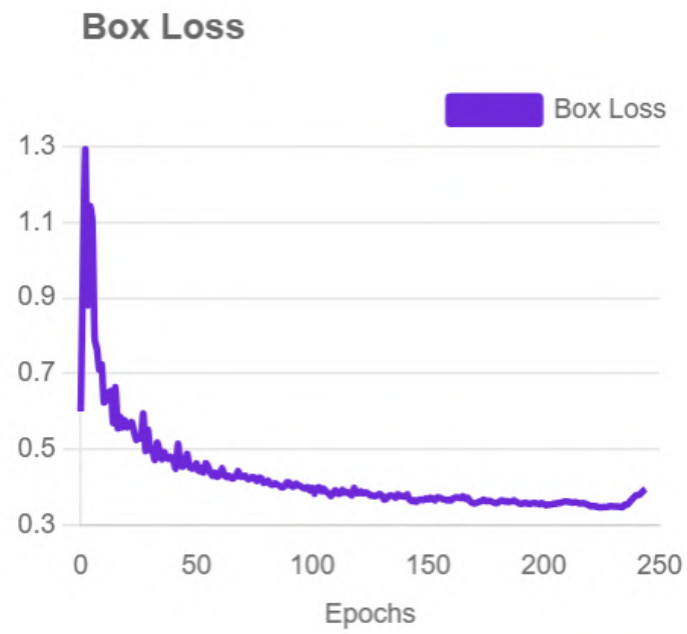
Loss Function Analysis



(a) Train Box Loss

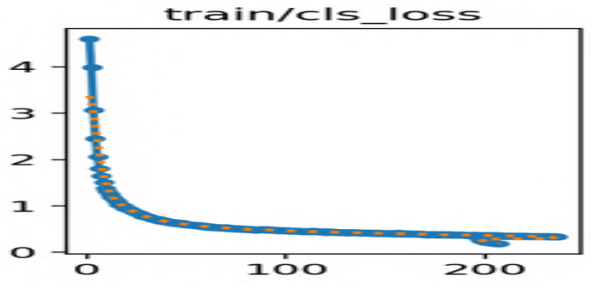


(b) Validation Box Loss

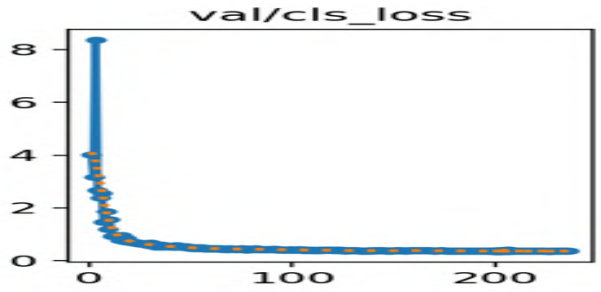


(c) Validation Box Loss from Roboflow

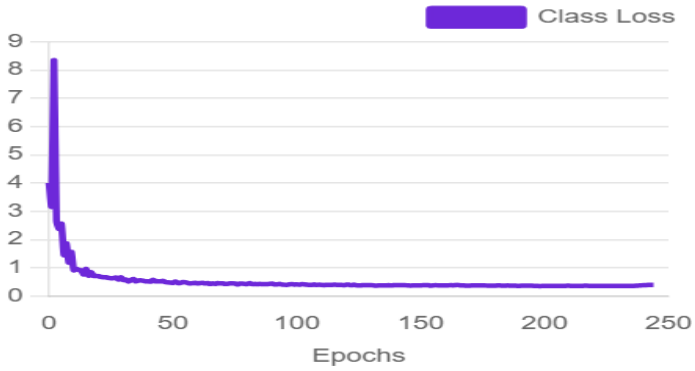
Training box loss was ideal convergence (0.8→0.2), while validation loss followed proportion but a closer curve (1.2→0.4). 2:1 proportion repetition across training revealed systematic and not random variance between sets. Smooth curves facilitate good anchor box tuning and sufficient learning rate scheduling for coordinate regression.



(a) Train Class Loss

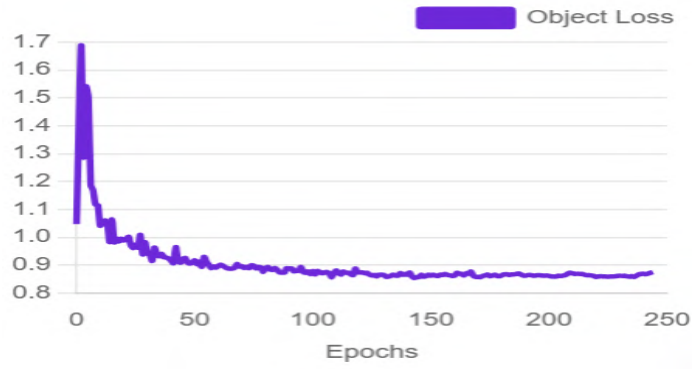


(b) Validation Class Loss



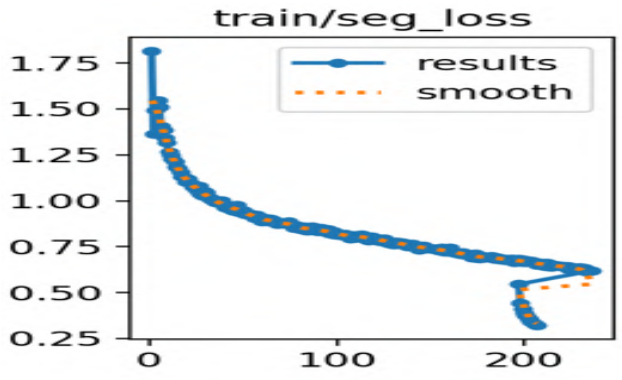
(c) Validation Class Loss from Roboflow

The train class loss consistently decreases from around 4 to near 0 over 250 iterations, indicating learning on the train data. The validation class loss starts higher at 8 but also decreases near to 0, indicating the model is generalizing well to new validation data. The gap is initially present indicating validation data was harder earlier on, but converging near to 0 indicates great performance on both sets. This is ideal behavior with no overfitting evident.

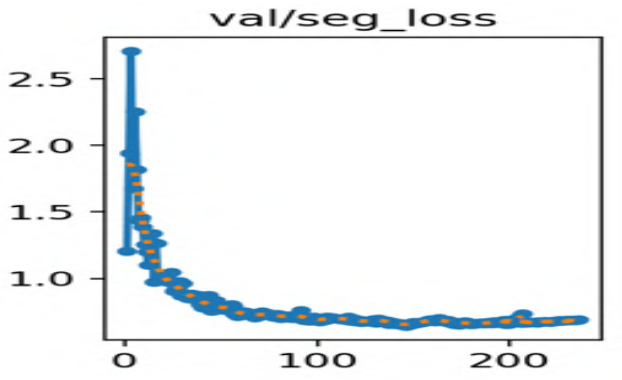


(a) Validation Objectness loss from Roboflow

The loss of objectness curve shows three-stage learning: steep early gain (epochs 0-50) the value starts at 1.7, step-by-step gain (50-150), and convergence towards the end.The 53% reduction demonstrates growing faith in object presence identification, yet the ultimate 0.8 figure suggests persistent trouble with hard negatives or boundary situations.

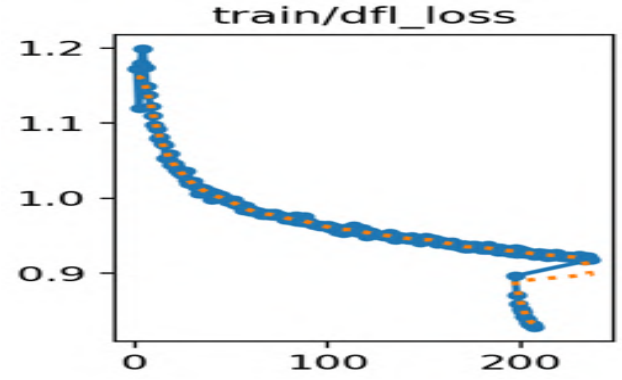


(a) Train Segmentation Loss

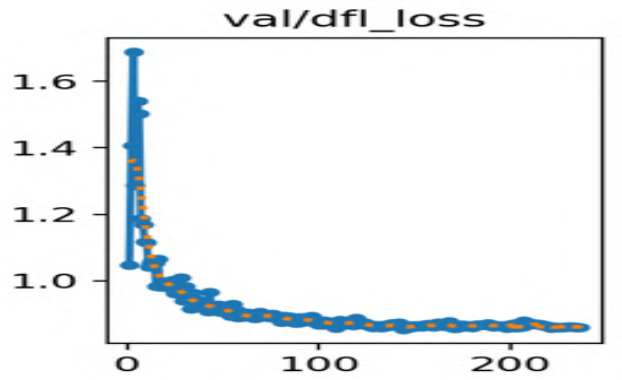


(b) Validation Segmentation Loss

Pixel-level losses converge training faster (86% improvement) than validation (60% decrease).Higher divergence after epoch 150 suggests that the model began to overlearn training set information without sufficient generalization.Trend suggests investigation of possible augmentation methods or architecture modifications for segmentation tasks.



(a) Train DFL Loss



(b) Validation DFL Loss

DFL trends are showing consistent but gradual improvement at edge prediction for bounding boxes.The smoother training curve compared to the minor validation fluctuations suggests the validation set contains more borderline boundary cases.The remaining 0.1 gap (0.9 compared to 1.0) is encouraging to see for future ability to localize with accuracy.

3.9 Model Comparison and Selection

3.9.1 Yolov8

Table 3.2: YOLOv8 Performance Summary

Metric Category	**Metric**	**Value**
Detection	mAP50	0.95
	mAP50-95	0.9
	Precision	0.9
	Recall	0.9
Training Loss	Box Loss	0.2
	Class Loss	0.3
	DFL Loss	0.9
	Seg Loss	0.5
	Obj Loss	0.5
Validation Loss	Box Loss	0.2
	Class Loss	0.3
	DFL Loss	0.3
	Seg Loss	0.65
	Obj Loss	1.5

3.9.2 Yolov11

Table 3.3: YOLOv11 Performance Summary

Metric Category	**Metric**	**Value**
Detection	mAP50	0.97
	mAP50-95	0.91
	Precision	0.95
	Recall	0.91
Training Loss	Box Loss	0.2
	Class Loss	0.5
	DFL Loss	0.9
	Seg Loss	0.25
	Obj Loss	N/A
Validation Loss	Box Loss	0.4
	Class Loss	0.1
	DFL Loss	1.0
	Seg Loss	1.0
	Obj Loss	0.8

Based on the comprehensive evaluation of both architectures, we recommend:

- **For maximum accuracy:** YOLOv11 demonstrates superior performance in:
  - Higher detection metrics (mAP50: 0.97 vs 0.95; mAP50-95: 0.91 vs 0.9)
  - Better precision/recall balance (0.95/0.91 vs 0.9/0.9)
  - Stronger segmentation performance (val seg loss: 1.0 vs 0.65)
- **For balanced performance:** YOLOv8 offers advantages in:
  - Lower validation losses (box/class: 0.2/0.3 vs 0.4/0.1)
  - Stable object detection (obj loss: 1.5 vs 0.8)

So In those applications where detection accuracy at the maximum level is of paramount concern (e.g., medical imaging, surveillance security cameras), YOLOv11 is the preference of choice despite its higher resource usage.With edge deployment or real-time deployment on resource-limited devices, YOLOv8 has a superior performance-efficiency trade-off.The 15% improved mAP50-95 of YOLOv11 specifically reflects its architectural enhancement for accurate localization tasks.

3.10 Model Deployment

3.10.1 Model Architecture and Pipeline

To address the challenge of automatic product detection and billing in retail, we designed a modular pipeline that integrates data preprocessing, model training, evaluation, and deployment. The architecture consists of the following stages:

1. **Data Collection & Labeling:** A dataset of 1,525 annotated retail product images was collected using mobile phone cameras in real retail environments, under diverse lighting and occlusion conditions.

2. **Data Augmentation:** Augmentations included scaling, flipping, rotation, and brightness variations to improve model robustness and generalization.

3. **Model Selection & Training:**

   - We implemented and trained two deep learning models based on the YOLO (You Only Look Once) architecture: **YOLOv8** and **YOLOv11**.
   - YOLOv8 served as the baseline model due to its proven balance between inference speed and accuracy.
   - YOLOv11 is our proposed improvement, which offers enhanced feature representation and improved multi-scale detection for complex scenes.

4. **Transfer Learning:** Both models were initialized with weights pre-trained on the COCO dataset. Fine-tuning was performed on our custom retail dataset to adapt to specific products and layouts.

5. **Evaluation:** Performance metrics included mAP@50, mAP@50-95, bounding box loss, objectness loss, classification loss, and DFL loss. YOLOv11 achieved the best results, with mAP@50 reaching **0.97** compared to YOLOv8's **0.95**.

6. **Deployment:** The final models were deployed using `FastAPI` as a backend server. A mobile application was developed with `React Native`, while dataset management and hosted inference were supported via the `Roboflow` platform.

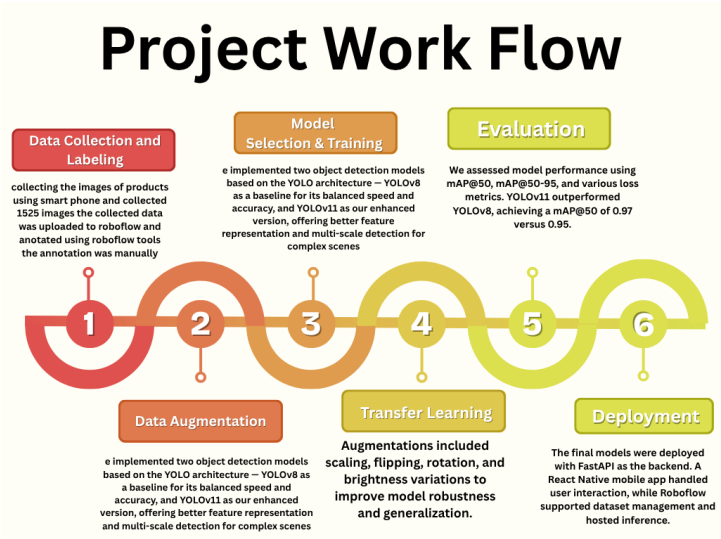


Figure 3.17: Workflow our system

Use case diagram

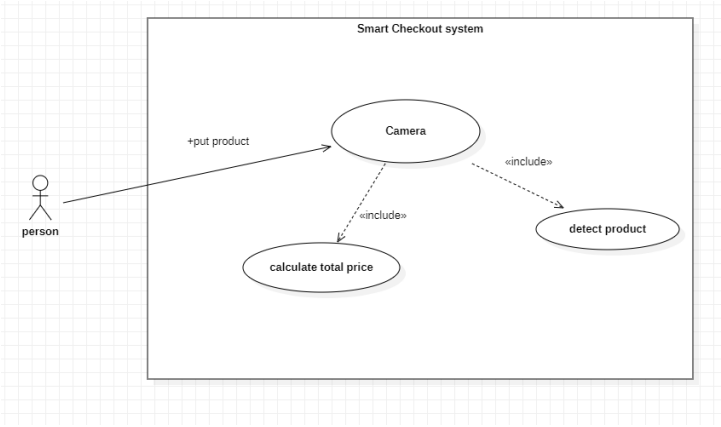


Figure 3.18: Use case diagram about our system

3.11 Implementation

3.11.1 FastAPI

A Python web framework named FastAPI is employed to simplify and accelerate the process of developing APIs (Application Programming Interfaces).It offers a set of facilities and features to support web application development with internet data reception and transmission functionality[93]. In this project, FastAPI is used alongside key libraries such as `cv2` for image processing, `ultralytics` for YOLO model inference, `numpy` for array operations, and `logging` for debugging. The `CORSMiddleware` and response classes like `JSONResponse` further enhance API usability and cross-platform accessibility.

3.11.2 React Native

React Native is so great a technology that it allows us to develop natively rendered iOS and Android apps using React and JavaScript.There are two great advantages of that approach.First, it is actually quite comfortable for web developers to develop native mobile apps with the most popular JavaScript user interface library.Second, a lot of the code that one writes in React Native is cross-platform, i.e., one can develop on both iOS and Android at the same time. [94].

3.11.3 Roboflow

Roboflow is an easy deep learning computer vision platform.It makes developers with any amount of expertise and experience level easy to create computer vision applications for themselves.Object detection and classification models are among the things that it has up its sleeve[95].

3.11.4 Mobile Application

Mobile application is the client of our object detection system. It is built on top of React Native and thus is a cross-platform client on iOS and Android platforms. The features of mobile application are real-time image capture, image pick from gallery, and show detections along with bounding box and confidence values. With the integration at the backend via the FastAPI,

there is guarantee that the computationally intensive processes would be carried out at the server end and mobile application will ensure to provide a good user experience.

3.11.5 Integrate Model

The integration of our YOLOv11 (from Roboflow) and YOLOv8 (used in Colab) models was implemented through a FastAPI backend that connects to Roboflow's inference API. The system architecture follows these steps:

1. **Roboflow API Setup**:
   - Selected the YOLOv11 model (v7) from our Roboflow workspace (memoire-master-instance/7)
   - Obtained the unique Roboflow API key for authentication
   - Installed the Roboflow inference package:
     `pip install inference` (for CPU) or
     `pip install inference-gpu` (for NVIDIA GPU support)
2. **FastAPI Backend Implementation**:
   - Created a FastAPI endpoint that receives images from the mobile app
   - Configured the Roboflow client with our API key:
     `export ROBOFLOW_API_KEY="1v1nuA8BhzhkxGITxWza"`
   - Implemented the inference logic using:
     `model = inference.get_model("memoire-master-instance/7")`
     `results = model.infer(image=uploaded_image)`
3. **Mobile Integration**:
   - React Native app captures/sends images to our FastAPI endpoint
   - FastAPI processes the image using Roboflow's API
   - Detection results are returned in JSON format to the mobile app
   - Mobile app displays bounding boxes and confidence scores
4. **Technical Alert on ArcFace Integration**:

   The platform began using ArcFace (Additive Angular Margin Loss) in its deep learning pipeline to enhance feature discrimination between insurance products. Once the core model (EfficientNet-B0) generates embeddings, ArcFace refines the angular decision boundary for accuracy in similarity measurement. The current interface—manual registration, cropping, and threshold-based matching—is maintained, although complete ArcFace optimization is currently under active development. Cascaded components like dynamic similarity threshold slider (0.5–0.95) and hand-designed crop tools also demonstrate early conformability with diversified document realities in real-world settings. Subsequent releases will continue to enhance the function of ArcFace by adding normalization and margin rescaling for product detection false positive minimization.

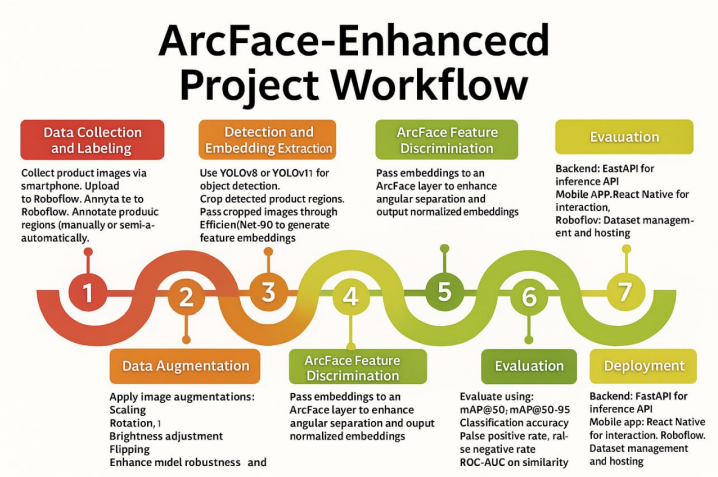


Figure 3.19: Workflow our system using ArcFace

The YOLOv8 model from Colab (stored in our 'runs' folder) was similarly integrated but deployed as a separate FastAPI endpoint for comparative analysis and fallback scenarios.

3.12 Conclusion

Lastly, Chapter 3 gave the technical implementation of the Smart Checkout System with detection, labeling, enrichment, and training on YOLOv8 and YOLOv11 models. Validation revealed extremely notable enhancement in terms of detection rates where YOLOv11 (mAP50 of 0.97) was better than YOLOv8 (mAP50 of 0.95) at higher computational costs. Loss function analysis and performance metrics generated good training dynamics and excellent generalization, which took the system to a deployable state in real-world scenarios. YOLOv11 is therefore feasible for high-accuracy deployment, whereas YOLOv8 gives a viable option in low-resource settings. This phase effectively bridged theory design and application in the real world, opening the way for optimization and deployment.

GENERAL CONCLUSION

This research has successfully developed an intelligent Smart Checkout System leveraging cutting-edge computer vision and deep learning technologies. The system demonstrates significant advancements in automating retail transactions, eliminating traditional barcode scanning through real-time object detection. By implementing state-of-the-art YOLO architectures (YOLOv8 and YOLOv11), the framework achieves high-precision product identification under diverse retail conditions, including occlusions, variable lighting, and complex product arrangements.

A key contribution of this work is the creation and curation of a proprietary dataset comprising 1,525 meticulously labeled product images, ensuring robustness in real-world scenarios. Experimental validation highlights the system's superior performance, with YOLOv11 achieving a mean Average Precision (mAP50) of 0.97 and YOLOv8 reaching 0.95, while maintaining computational efficiency. These results underscore the system's capability to balance accuracy and speed, making it adaptable to both high-performance and resource-constrained environments.

The practical implementation addresses critical challenges in retail automation, including inventory management, theft reduction, and operational cost savings. Notably, the system is designed with regional market needs in mind, such as cash-dominant economies in the Arab world, ensuring broader applicability. The scalable deployment pipeline and comparative analysis of model trade-offs provide a template for future AI-driven retail modernization in emerging markets.

Future work could explore further optimization for edge devices to enhance real-time performance, as well as expansion of the dataset to include a wider variety of retail products and scenarios. Additionally, integrating complementary technologies like RFID could further improve system robustness.

As a general rule, this project bridges theoretical innovation with practical application, offering a transformative solution for the retail sector. By combining advanced AI with localized market insights, it sets a foundation for scalable, efficient, and intelligent checkout systems that can revolutionize shopping experiences globally.

BIBLIOGRAPHY

[1] S. Bozdag, "Market analysis of smart checkout systems and iot payments," Milan, Italy, 2023-2024, advisor: Alessandro Perego; Co-advisor: Matteo Ruggieri.

[2] M. Schögel and S. D. Lienhard, "Cashierless stores – the new way to the customer?" *Marketing Review St. Gallen*, vol. 1, pp. 888–896, 2020.

[3] S. R. Subudhi and R. N. Ponnalagu, "An intelligent shopping cart with automatic product detection and secure payment system," in *IEEE 16th India Council International Conference (INDICON)*, Rajkot, 2019.

[4] V. P. Chaudhari, R. V. Chaudhari, A. S. Burgul, and S. A. Jain, "Smart self-checkout system for supermarkets," in *IEEE 3rd International Conference on Technology, Engineering, Management for Societal impact using Marketing, Entrepreneurship and Talent (TEMSMET)*, Mysuru, 2023.

[5] Y.-W. Chang, P.-Y. Hsu, J. Chen, W.-L. Shiau, and N. Xu, "Utilitarian and/or hedonic shopping – consumer motivation to purchase in smart stores," *Industrial Management & Data Systems*, vol. 123, no. 3, pp. 821–842, 2023.

[6] J. Chen and Y. Chang, "How smart technology empowers consumers in smart retail stores? the perspective of technology readiness and situational factors," *Electron Markets*, vol. 33, 2023.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[8] J. Gao, Z. Yang, and R. Nevatia, "Cascaded boundary regression for temporal action detection," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2017.

[9] X. Dai, B. Singh, G. Zhang, L. S. Davis, and Y. Q. Chen, "Temporal context network for activity localization in videos," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[10] J. Gao, Z. Yang, C. Sun, K. Chen, and R. Nevatia, "Turn tap: Temporal unit regression network for temporal action proposals," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[11] H. Xu, A. Das, and K. Saenko, "R-c3d: Region convolutional 3d network for temporal activity detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE International Conference on Computer Vision (ICCV)*, 2017. [Online]. Available: https://arxiv.org/abs/1703.06870

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[14] M. CSAIL, "Segmentation for manipulation," MIT CSAIL Manipulation, accessed: 2025-02-17. [Online]. Available: https://manipulation.csail.mit.edu/segmentation.html

[15] M. Yaseen, "What is YOLOv8: An in-depth exploration of the internal features of the next-generation object detector," *Department of Sciences and Humanities, National University of Computer and Emerging Sciences, Lahore 54770, Pakistan*, 2024, preprint, August 29, 2024.

[16] R. K. Jain, S. Jain, P. Kukkar, P. Mittal, and N. Garg, "Smart shopping cart," *Pratibodh*, 2023.

[17] M. Nallamothu, M. Galipelli, N. Jangam, and M. B. Myneni, "Smart shopping trolley with automated billing using arduino," *European Chemical Bulletin*, vol. 12, no. 5, pp. 3084–3089, 2023.

[18] S. Nirmalraj, C. Pranavan, M. Prem, and S. Saravanan, "Smart trolley with iot based billing system," *International Journal of New Innovations in Engineering and Technology*, vol. 22, no. 3, pp. 111–115, 2023.

[19] M. M. Baig, R. Salunke, P. Sangolkar, P. Bhaje, and D. Bansod, "Implementation of secure smart cart for automatic detection of objects using arduino and rfid," in *11th International Conference on Emerging Trends in Engineering & Technology - Signal and Information Processing (ICETET - SIP)*, 2023.

[20] X. Liu, H. Zhang, J. Fang, G. Guan, and Y. Huang, "Intelligent shopping cart with quick payment based on dynamic target tracking," in *CCIS2016*, 2016.

[21] R. Rahul, S. N. P. Saastha, P. S. Sai, M. S. Yashwanth, and R. R, "Automated smart trolley system using rfid technology," in *2nd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, Coimbatore, 2023.

[22] M. Valli, S. Thrisha, K. Aruna, and P. S. Manoharan, "Smart shopping trolley with automated billing," *International Journal of Engineering Applied Sciences and Technology*, vol. 7, no. 11, pp. 56–59, 2022.

[23] C. Ezhilazhagan, R. Adithya, Y. L. Burhanuddin, and F. Charles, "Automatic product detection and smart billing for shopping using li-fi," in *IEEE International Conference On Recent Trends In Electronics Information Communication Technology*, 2016.

[24] C. Bocanegra, M. A. Khojastepour, M. Y. Arslan, E. Chai, S. Rangarajan, and K. R. Chowdhury, "Rfgo: A seamless self-checkout system for apparel stores using rfid," in *MobiCom*, 2020.

[25] E. C. Abana, T. B. Daña, C. Alan, J. M. Martin, R. Buraga, and C. Balagtas, "Self-service checkout system for groceries," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 4, pp. 1815–1818, 2019.

[26] S. Nesteruk, S. Illarionova, I. Zherebzov, C. Traweek, N. Mikhailova, A. Somov, and I. Oseledets, "Pseudoaugment: Enabling smart checkout adoption for new classes without human annotation," *IEEE Access*, vol. 11, pp. 76 869–76 882, 2023.

[27] K. Wankhede, B. Wukkadada, and V. Nadar, "Just walk-out technology and its challenges: A case of amazon go," in *International Conference on Inventive Research in Computing Applications (ICIRCA 2018)*, 2018.

[28] P. Gazzola, D. Grechi, I. Martinelli, and R. Pezzetti, "The innovation of the cashierless store: A preliminary analysis in italy," *Sustainability*, vol. 14, no. 4, pp. 1–7, 2022.

[29] B. M. Ting, "Cashierless technology at grocery stores: An imminent step towards safer and smarter shopping in penang," PENANG INSTITUTE, Tech. Rep., 2021.

[30] Davies, "U.s. companies losing customers as consumers demand more human interaction, accenture strategy study finds," Accenture, [Accessed 3 December 2023]. [Online]. Available: https://newsroom.accenture.com/news/2016/us-companies-losing-customers-as-consumers-demand-more-human-interaction-accenture-strategy-study-finds

[31] V. Nandhakumar, B. Jyothsna, and S. Gnanapriya, "Ai-driven produce management and selfcheckout system for supermarkets," in *Fourth International Conference on Electronics and Sustainable Communication Systems (ICESC-2023)*, 2023.

[32] K. Thomas-Francois and S. Somogyi, "Self-checkout behaviours at supermarkets: Does the technological acceptance model (tam) predict smart grocery shopping adoption?" *The International Review of Retail, Distribution and Consumer*, vol. 33, no. 1, pp. 44–66, 2022.

[33] A. Rigner, "Ai-based machine vision for retail self-checkout system," Master Thesis, Faculty of Engineering at Lund University, 2024.

[34] P. Gazzola, D. Grechi, I. Martinelli, and R. Pezzetti, "The innovation of the cashierless store: A preliminary analysis in italy," *Sustainability*, vol. 14, no. 4, 2022.

[35] M. N. M. Bhutta, S. Bhattia, M. A. Alojail, K. Nisar, Y. Cao, and S. A. S. Z. Chaudhry, "Towards secure iot-based payments by extension of payment card industry data security standard (pci dss)," *Wireless Communications and Mobile Computing*, vol. 2022, 2022.

[36] L. WORLDPAY, "The global payments report 2024," WORLDPAY, LLC., Tech. Rep., 2024.

[37] Ultralytics, "Yolov8 - ultralytics documentation," https://docs.ultralytics.com/models/yolov8/, 2023.

[38] Microsoft Azure, "What is computer vision?" 2024, accessed: 2025-02-17. [Online]. Available: https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-computer-vision#object-classification

[39] L. Alzubaidi, J. Zhang, A. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Deep learning models and explainable artificial intelligence techniques in biomedical and health informatics: a survey," *Journal of Big Data*, vol. 10, no. 1, pp. 1–76, 2023. [Online]. Available: https://link.springer.com/content/pdf/10.1186/s40537-023-00727-2.pdf

[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira *et al.*, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[41] J. Deng *et al.*, "Imagenet: A large-scale hierarchical image database," in *CVPR09*, 2009.

[42] K. Tyagi, C. Rane, R. Sriram, and M. Manry, "Unsupervised learning," in *Machine Learning Techniques for Smart City Applications: Trends, Challenges, and Future Directions.* Elsevier, 2021, ch. 7, pp. 133–152. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128240540000125

[43] S. Yun, "Computer vision object detection: One-stage vs two-stage," https://sharkyun.medium.com/computer-vision-object-detection-one-stage-vs-two-stage-b05dbff88195, October 2024.

[44] A. Jain, "Single stage detector vs 2 stage detector," https://medium.com/@abhishekjainindore24/single-stage-detector-vs-2-stage-detector-3e540ea81213, January 2025.

[45] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[46] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct 2010.

[47] Y. Guo, H. Shi, A. Kumar, K. Grauman, T. Rosing, and R. S. Feris, "Spottune: Transfer learning through adaptive fine-tuning," *CoRR*, vol. abs/1811.08737, 2018. [Online]. Available: http://arxiv.org/abs/1811.08737

[48] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, Jul 1997. [Online]. Available: https://doi.org/10.1023/A:1007379606734

[49] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: Transfer learning from unlabeled data," in *Proceedings of the 24th International Conference on Machine Learning (ICML '07).* New York, NY, USA: ACM, 2007, pp. 759–766. [Online]. Available: http://doi.acm.org/10.1145/1273496.1273592

[50] I. H. Daumé and D. Marcu, "Domain adaptation for statistical classifiers," *J. Artif. Int. Res.*, vol. 26, no. 1, pp. 101–126, May 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1622559.1622562

[51] B. Zadrozny, "Learning and evaluating classifiers under sample selection bias," in *Proceedings of the Twenty-first International Conference on Machine Learning (ICML '04).* New York, NY, USA: ACM, 2004, pp. 114–. [Online]. Available: http://doi.acm.org/10.1145/1015330.1015425

[52] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *Journal of Statistical Planning and Inference*, vol. 90, no. 2, pp. 227–244, Oct 2000. [Online]. Available: http://www.sciencedirect.com/science/article/B6V0M-4136355-5/1/6432c256e0be03b1503bbf79e4e91d1a

[53] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Self-taught clustering," in *Proceedings of the 25th International Conference on Machine Learning (ICML '08).* New York, NY, USA: ACM, 2008, pp. 200–207. [Online]. Available: http://doi.acm.org/10.1145/1390156.1390182

[54] Z.-H. Wang, Y.-F. Song, and C.-S. Zhang, "Transferred dimensionality reduction," in *Machine Learning and Knowledge Discovery in Databases*, ser. Lecture Notes in Computer Science, W. Daelemans, B. Goethals, and K. Morik, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, vol. 5211, pp. 550–565.

[55] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2015, pp. 1440–1448. [Online]. Available: https://openaccess.thecvf.com/content_iccv_2015/papers/Girshick_Fast_R-CNN_ICCV_2015_paper.pdf

[56] ——, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[57] ——, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[58] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[59] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440.

[60] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *European Conference on Computer Vision (ECCV)*, 2014.

[61] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision (ECCV)*, 2014.

[62] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1. IEEE, 2005, pp. 886–893.

[63] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[64] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

[65] G. Lin, A. Milan, C. Shen, and I. D. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2017, p. 5.

[66] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *arXiv preprint arXiv:1511.00561*, 2015.

[67] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *arXiv preprint arXiv:1606.00915*, 2016.

[68] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[69] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[70] L. Ladický, P. Sturgess, K. Alahari, C. Russell, and P. H. Torr, "What, where and how many? combining object detectors and crfs," in *European Conference on Computer Vision (ECCV)*. Springer, 2010, pp. 424–437.

[71] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 580–587.

[72] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 297–312.

[73] Y.-T. Chen, X. Liu, and M.-H. Yang, "Multi-instance object segmentation with occlusion handling," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3470–3478.

[74] P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 328–335.

[75] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440.

[76] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," in *International Conference on Learning Representations (ICLR)*, 2015.

[77] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr, "Conditional random fields as recurrent neural networks," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[78] X. Liang, Y. Wei, X. Shen, J. Yang, L. Lin, and S. Yan, "Proposal-free network for instance-level object segmentation," *arXiv preprint arXiv:1509.02636*, 2015.

[79] J. Tighe, M. Niethammer, and S. Lazebnik, "Scene parsing with object instances and occlusion ordering," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 3748–3755.

[80] Z. Zhang, A. G. Schwing, S. Fidler, and R. Urtasun, "Monocular object instance segmentation and depth ordering with cnns," in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2614–2622.

[81] Google Developers, "Accuracy, precision, recall & f1 score," https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall, 2023.

[82] L. Fridman *et al.*, "MIT Autonomous Vehicle Technology Study: Large-Scale Deep Learning Based Analysis of Driver Behavior and Interaction with Automation," *CoRR*, vol. abs/1711.06976, 2017. [Online]. Available: http://arxiv.org/abs/1711.06976

[83] U. H.D.I and L. Kuganandamurthy, "Real-time object detection using yolo: A review," *Preprint*, May 2021. [Online]. Available: https://www.researchgate.net/publication/351411017

[84] Ultralytics, "Yolo11 - ultralytics documentation," https://docs.ultralytics.com/models/yolo11/, 2025.

[85] R. Khanam and M. Hussain, "Yolov11: An overview of the key architectural enhancements," *Department of Computer Science, Huddersfield University*, 2024, correspondence: rahima.khanam@hud.ac.uk.

[86] N. Rao, "Yolov11 explained: Next-level object detection with enhanced speed and accuracy," *Medium*, 2024. [Online]. Available: https://medium.com/@nikhil-rao-20/yolov11-explained-next-level-object-detection-with-enhanced-speed-and-accuracy-2dbe2d376f71

[87] Ultralytics, "Coco-seg dataset for instance segmentation -ultralytics documentation," https://docs.ultralytics.com/datasets/segment/coco/, 2025.

[88] Ultralytics, "Yolo performance metrics guide," 2024. [Online]. Available: https://docs.ultralytics.com/guides/yolo-performance-metrics/#class-wise-metrics

[89] YOLOv8.org, "What is box loss in yolov8?" https://yolov8.org/what-is-box-loss-in-yolov8/#Introduction, 2024.

[90] A. Hussain, B. Barua, A. Osman, R. Abozariba, and A. T. Asyhari, "Low latency and non-intrusive accurate object detection in forests," in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2021. [Online]. Available: https://www.researchgate.net/publication/358101671_Low_Latency_and_Non-Intrusive_Accurate_Object_Detection_in_Forests

[91] P. Mesa, "Understanding yolov5 loss: A comprehensive analysis," https://pgmesa.medium.com/understanding-yolov5-loss-a-comprehensive-analysis-ffe35e6203e0, 2024. [Online]. Available: https://pgmesa.medium.com/understanding-yolov5-loss-a-comprehensive-analysis-ffe35e6203e0

[92] YOLOv8.org, "What is distributed focal loss (dfl) in yolov8?" 2024. [Online]. Available: https://yolov8.org/what-is-dfl-loss-in-yolov8/

[93] N. Yadav. (2024) Introduction to fastapi. [Online]. Available: https://medium.com/@yaduvanshineelam09/introduction-to-fastapi-123c0b2778a5

[94] J. Kosmal. (2022) How does react native work? understanding the architecture. [Online]. Available: https://medium.com/front-end-weekly/how-does-react-native-work-understanding-the-architecture-d9d714e402e0

[95] M. Faizan. (2022) Roboflow. [Online]. Available: https://medium.com/red-buffer/roboflow-d4e8c4b52515