A PROJECT REPORT ON

Applying Image Processing to Determine Product Placement and Availability on the Shelf in Retail Stores

SUBMITTED TO THE SAVITRIBAI PHULE UNIVERSITY, PUNE
IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE

OF

BACHELOR OF ENGINEERING (Computer Engineering)

SUBMITTED BY

Harsh Dhawale	Exam No: B190054272
Pratik More	Exam No: B190054402
Sakshi Rathi	Exam No: B190054455
Sanket Jhavar	Exam No: B190054317

DEPARTMENT OF COMPUTER ENGINEERING
Pune Institute of Computer Technology
Dhankawadi, Pune - 411043

SAVITRIBAI PHULE PUNE UNIVERSITY
2023-2024

CERTIFICATE

This is to certify that the project report entitled

Applying Image Processing to Determine Product Placement and Availability on the Shelf in Retail Stores

Submitted by

Harsh Dhawale	Exam No: B190054272
Pratik More	Exam No: B190054402
Sakshi Rathi	Exam No: B190054455
Sanket Jhavar	Exam No: B190054317

is a bonafide work carried out by them under the supervision of **Prof. B.P.Mashram** and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University for the award of the Degree of **Bachelor of Engineering** (COMPUTER ENGINEERING).

Prof. B.P.Masram
Internal Guide
Dept. of Computer Engg.

Dr. G.V. Kale
Head
Dept. of Computer Engg.

Dr. S. T. Gandhe
Principal
Pune Institute of Computer Technology

Place:Pune
Date:

ACKNOWLEDGEMENT

It gives us great pleasure in presenting the project report on ***'Applying Image Processing to Determine Product Placement and Availability on the Shelf in Retail Stores'****.*

We would like to take this opportunity to thank ***Prof. B.P.Masram*** *giving us all the help and guidance we needed. We are really grateful to them for their kind support. Their valuable suggestions were very helpful.*

We are also grateful to ***Dr. G.V. Kale****, Head of Computer Engineering Department, PICT for her indispensable support, suggestions.*

In the end our special thanks to all faculty members for their whole hearted cooperation for completion of this report. We also thank our laboratory assistants for their valuable help in laboratory.

Harsh Dhawale
Pratik More
Sakshi Rathi
Sanket Jhavar
(B.E. Computer Engg.)

ABSTRACT

The absence of product availability and the improper arrangement of items on the shelves in retail stores can lead to reduced sales for the retailer. personnel of Fast-Moving Consumer Goods (FMCG) businesses, whose products are stocked on these shelves, as well as personnel of the retailer undertake visual checks to detect products that are lost or out of stock are expensive and tends to commit mistakes. A technique for automating the manual inspection process is presented in this study. The research also shows that it is possible to identify empty spaces on the shelves in addition to recognizing and counting products that face front by using image processing algorithms. Moreover, this approach can be applied to images captured from a video stream, such as those from security cameras, to count the number of visible units of a particular product on a shelf and determine whether they are correctly oriented, as required. The image processing method proposed in this paper primarily facilitates the proper positioning of products on the front row of the shelves. While this might appear to be a limitation in terms of inventory management,it holds substantial significance for product manufacturers who lease shelf space and allocate specific shelf positions, compelling retailers to place particular products in designated locations. The innovative aspect of this paper lies in the extension of feature extraction in image processing to highlight and rectify incorrect placements and arrangements of items on the shelves. Importantly, the implemented solution does not entail significant additional infrastructure costs.

Contents

List of Figures

CHAPTER 1

INTRODUCTION

1.1 Overview

As the retail industry continues to undergo rapid transformations driven by technological advancements and changing consumer expectations, the significance of optimizing product availability and shelf organization becomes even more pronounced. The intricacies of this optimization extend beyond the mere visual appeal of a well-organized shelf; they delve into the realms of operational efficiency, customer satisfaction, and ultimately, store profitability.

At the heart of this complex retail ecosystem lies the meticulous dance of arranging products on store shelves. This choreography, while often underappreciated, is a critical factor that shapes the overall retail experience. The way products are presented influences not only a customer's perception of a store but also holds the power to sway their purchasing decisions. In this intricate tapestry of challenges, one particularly formidable hurdle emerges—the detection of void spaces.

Void spaces, those seemingly inconspicuous gaps between products on store shelves, may appear minor, but their impact can reverberate throughout the entire retail landscape. Overlooking these voids can lead to suboptimal shelf utilization, disrupting the delicate balance of inventory management systems and potentially eroding the overall shopping experience. As consumers increasingly demand a seamless and convenient shopping journey, addressing this challenge becomes imperative for retailers looking to stay ahead in a competitive market.

The detection of void spaces requires a concerted effort to forge innovative solutions that seamlessly integrate the dynamic world of retail with cutting-edge technologies. One such solution involves the implementation of advanced computer vision systems and artificial intelligence (AI) algorithms. These technologies can analyze the arrangement of products on shelves in real-time, identifying void spaces with precision and efficiency. By leveraging image recognition and machine learning capabilities, retailers can gain valuable insights into shelf organization, ensuring that products are not only visually appealing but also strategically positioned to maximize sales.

Moreover, the integration of Internet of Things (IoT) devices can enhance the accuracy of void detection. Smart shelves equipped with sensors can provide real-time data on product placement, allowing retailers to proactively address void spaces and prevent potential disruptions in the shopping journey. This level of connectivity also facilitates data-driven decision-making, empowering retailers to optimize shelf layouts based on consumer behavior and product popularity.

To tackle the challenge of void space detection comprehensively, collaboration between retailers and technology providers is essential. Retailers need to invest in training their staff to understand and utilize these technological solutions effectively. Additionally, technology providers must continue refining their algorithms and systems to adapt to the evolving demands of the retail landscape.

1.2 Motivation

For small retail shops, keeping track of stock and ensuring products are in the right place is often a struggle. Mistakes can lead to customer frustration and lost sales. By using machine learning technology to monitor shelves in real time and make sure products are where they should be, these shops can work more efficiently and avoid running out of popular items. This helps them serve customers better and saves time and money on manual stock checks.

When customers can easily find what they need, they're more likely to enjoy their shopping experience and come back for more. By using machine learning to maintain an organized and well-stocked store, small retailers can build trust with their customers and stay ahead of the competition. This means happier customers and a more successful business.

Using machine learning also helps these shops save money. By automating tasks like inventory management, they can reduce the risk of errors and save time that can be better spent on serving customers. This leads to better use of resources and a healthier bottom line, making the business more stable and sustainable in the long run.

In a constantly changing retail world, small shops need to stay flexible and adapt to what customers want. By using machine learning to make sure products are where they're supposed to be, these shops can stay ahead of the curve and continue to meet the needs of their customers. This helps them grow and thrive in a competitive retail market.

1.3 Problem Definition and Objectives

Title: Automated Void Space Detection and Planogram Compliance Verification in Retail Environments.

Problem Statement: In the context of small retail operations, the accurate management of product positioning and the timely detection of void spaces on store shelves are critical for ensuring efficient inventory management and a seamless shopping experience for customers. However, the manual monitoring of shelf layouts and the identification of misplaced products and empty spaces often prove to be labor-intensive, error-prone, and time-consuming, leading to operational inefficiencies and potential revenue losses.

Objective: The primary objective of this project is to develop a machine learning-based system that can accurately detect void spaces and verify the correct positioning of products within the designated shelves in real-time. The system aims to streamline the process of planogram compliance and enhance the overall efficiency of stock management in small retail shops, ultimately improving the customer shopping experience and fostering long-term business growth.

1.4 Project Scope & Limitations

Scope:
The project will involve the collection and preprocessing of a diverse dataset of shelf images and videos from various retail environments. The dataset will be annotated to label products, void spaces, and any anomalies using a robust annotation tool. The yolov9 model will be employed as the core machine learning algorithm for object detection, tailored specifically for void space identification and product positioning verification within the context of planogram compliance.

The system will be designed to provide real-time notifications to store staff or management upon the detection of any discrepancies in product placement or the presence of void spaces on the shelves. Additionally, the system will incorporate a follow-up mechanism to ensure that any identified void spaces are promptly addressed and rectified within a specified time frame, minimizing the risk of stockouts and customer dissatisfaction.

Limitations:

Accuracy:
Model Performance: The accuracy of the system heavily relies on the performance of the yolov9 model. Factors like the quality and diversity of the training dataset, lighting conditions in the retail environment, and occlusions of products can all impact the model's ability to correctly detect void spaces and product positions. Labeling Errors: Inaccuracies in the annotation of the training data (e.g., mislabeled products or void spaces) can be propagated into the model's predictions, limiting its overall effectiveness. Real-Time Applicability:
Computational Requirements: Running complex machine learning models like yolov9 in real-time can require significant computational resources. Depending on the hardware used, there might be limitations on the processing speed and the number of cameras that can be simultaneously monitored. Environmental Constraints:
Lighting Variations: Fluctuations in lighting conditions throughout the day or due to shadows can affect the model's ability to accurately identify objects.
Camera Angles: The system's performance might be dependent on the placement and angles of the cameras. Improper camera placement could lead to blind spots or occlusions that hinder object detection. System Management:
False Positives/Negatives: The system might generate false positive (detecting void spaces where there are none) or false negative (missing actual void spaces) alerts. These errors can lead to unnecessary work for store staff or a false sense of security.
Alert Fatigue: Constant notifications about minor discrepancies or frequent false alarms might lead to alert fatigue, causing staff to ignore important notifications.

1.5 Methodologies of Problem solving

- A. Obtain Real-Time Security Camera Images: In this step, security cameras are positioned strategically throughout the retail space to continuously take pictures of the shelves. Real-time information about the availability and placement of products is provided by these photos. The photos that are taken are used as the raw data for later phases of processing and analysis. Clear, high-resolution photos are necessary for precise product and void space identification and detection.
- B. Use YOLOv9 to Find Void Spaces: YOLOv9, or "You Only Look Once version 9," is an object detection technique that relies on deep

learning. It is used to identify empty spots on the shelves—areas where products are missing—in the photos that have been taken. Because YOLOv9 can detect objects in real time, it is a good tool for swiftly evaluating vast amounts of photos. The algorithm assists retailers in identifying regions that need to be restocked or attended to in order to ensure optimal product availability by precisely identifying vacant spots.

- C. Give the input reference and the target picture. In this stage, the system receives reference photographs of properly packed shelves as input. These reference photos act as standards by which to measure how the shelves are currently doing. The target image, which shows the shelves as they are right now as obtained by the security camera, is given to the system. The system detects differences in product availability and positioning by comparing the target image with the reference images.

- D. Preprocess the images: In order to improve quality and streamline processing, incoming images are preprocessed before feature extraction and analysis. By doing this, the grayscale photos are converted, which lowers the computational cost of the actions that follow. Furthermore, photos can be edited as necessary to highlight particular focal points, like individual shelves or product displays. By removing redundant data, cropping increases the effectiveness of feature extraction and comparison algorithms.

- E. Feature extraction and matching of descriptors: Finding important locations or features, such corners, edges, or blobs, in the reference and target photos is the process of feature extraction. The visual and geometric qualities of these features are then described by computing descriptors, which are numerical representations of these aspects.

- F. The descriptors of the reference and target images are compared using descriptor matching techniques, including the closest neighbor approach. Through the establishment of feature correspondences, the system is able to discern similarities and differences between the images.

- G. Syncing : In order to evaluate the similarity between the reference and target images, the system synchronizes their collected attributes in this step. To find the level of connection between the images, this entails comparing the retrieved features and descriptors.

There are two approaches explained for synchronization: directly comparing descriptors or creating a connection between characteristic locations in the reference image and descriptors in the target image. Accurate comparison and alignment of the images for further analysis are made easier by both techniques.

- H. Finding the Positions of Products: The system determines the product placements in the target image by utilizing the synchronized features between the reference and target images. To locate products, this entails comparing features and descriptors from the reference image with those in the target image. The system can evaluate the spatial distribution of products on the shelves and discover any deviations from the ideal arrangement shown in the reference photographs by precisely detecting the positions of the products.

- I. Putting Found Products on Display: When products are found in the target image, the system surrounds them with polygons or bounding boxes to graphically represent their locations. This gives the specified products and their places on the shelves a clear visual depiction. Retailers can take use of the exhibited products' actionable insights, which emphasize product regions and point out empty places that need to be filled or rearranged.

- J. Iterative Procedure: Because the entire procedure is iterative, the retail environment can be continuously observed and analyzed throughout time. The system repeats the identification, comparison, and visualization processes whenever new images are taken by the security cameras in order to deliver current information about the availability and location of products.

CHAPTER 2

LITERATURE SURVEY

The first research paper titled ”Out-of-stock justifications and consumers’ behavioral outcomes- explores the role of product type and sales level information in out-of-stock situations”. This paper was published in the Journal of Retailing and Consumer Services in 2021. It investigates how information about product type and sales level affects consumer behavior when they encounter out-of-stock situations.

The second research paper titled ”Design and application of a UHF RFID tag for monitoring pollution in high voltage towers” was published in Electric Power Systems Research in 2023. It describes the design and application of a UHF RFID tag that can be used to monitor pollution on high voltage power lines.

The third research paper titled ”C. Chu, J. Niu, W. Zheng, J. Su, G. Wen A time-efficient protocol for unknown tag identification in large-scale RFID systems IEEE Internet of Things Journal (2021)” proposes a time-efficient protocol for identifying unknown tags in large-scale RFID systems.

The fourth research paper titled ”Analysis SURF feature extraction and SVM classification for facial image recognition from various angles. Citation Aripin Rambe et al 2020 IOP Conf.” investigates the use of SURF features and SVM classification for facial image recognition from various angles.

Year	Auhtor	Highlights of the Research
2021	Ezhil Kumar, Madhumitha, Sharma, Dheeraj P.	Out-of-stock justifications and consumers' behavioral outcomes–exploring the role of product type and sales level information in out-of-stock situations 2021, Journal of Retailing and Consumer Services
2023	Lidja N.T.Alves, Edson G. da Costa, Alexandre J.R.Serres	Design and application of a UHF RFID tag for monitoring pollution in high voltage towers 2023, Electric Power Systems Research.
2021	C. Chu, J. Niu, W. Zheng, J. Su, G. Wen A	C. Chu, J. Niu, W. Zheng, J. Su, G. Wen A time-efficient protocol for unknown tag identification in large-scale RFID systems IEEE Internet of Things Journal (2021),
2020	Aripin Rambe	Analysis SURF feature extraction and SVM classification for facial image recognition from various angles.Citation Aripin Rambe et al 2020 IOP Conf.

Table 2.1: Literature survey on Void-space Detection

CHAPTER 3

SOFTWARE REQUIREMENTS SPECIFICATION

3.1 Assumptions and Dependencies

The project's assumptions revolve around the availability of a diverse and well-annotated dataset for training the machine learning model, ensuring the system's adaptability to different shelf layouts and product categories. Additionally, the project assumes the availability of appropriate hardware resources capable of supporting real-time image processing and monitoring tasks.

Dependencies include the seamless integration of the developed system with the existing inventory management software used by the retail shop, as well as the availability of reliable internet connectivity for real-time data transmission and notifications. By acknowledging these assumptions and dependencies, the project aims to foster a comprehensive understanding of the underlying prerequisites and external factors influencing the successful implementation of the automated void space detection and planogram compliance verification system.

3.2 Functional Requirements

To ensure the successful implementation of the automated void space detection and planogram compliance verification system, the following functional requirements have been identified:

3.2.1 System Feature 1(Real-time Void Space Detection)

The system will incorporate cutting-edge computer vision techniques to enable real-time monitoring of the store's shelves. By analyzing high-resolution images captured at regular intervals, the system will utilize advanced image processing algorithms, such as convolutional neural networks (CNNs), to accurately identify and delineate any unoccupied spaces or gaps on the shelves. Leveraging these sophisticated algorithms, the system will meticulously distinguish between void spaces and areas occupied by products, ensuring precise and reliable void space detection with minimal false positives or false negatives.

Additionally, the system will be equipped with a robust notification mechanism, instantly alerting the store staff or management via a user-friendly interface or mobile application as soon as any void spaces are detected. These immediate notifications will enable swift corrective actions, allowing the store personnel to promptly replenish the empty shelves, minimize potential stock-outs, and maintain an organized and visually appealing store layout.

3.2.2 System Feature2 (Product Positioning Verification)

The system will employ a combination of computer vision algorithms and machine learning techniques to verify the accurate positioning of products on the store shelves in accordance with the predefined planogram guidelines. By leveraging state-of-the-art object detection models, such as yolov9 or SSD (Single Shot Multibox Detector), the system will conduct a comprehensive analysis of the shelf images, precisely identifying each product's location and orientation within its designated shelf space. Through this process, the system will compare the actual product placements against the predetermined planogram specifications, flagging any discrepancies or misalignments for the store staff's attention.

The user interface will provide intuitive visual cues and indicators, highlighting the specific products that require repositioning or adjustment to ensure adherence to the planogram guidelines. By facilitating seamless product reorganization and positioning verification, the system will contribute to the maintenance of consistent planogram compliance, enhancing the store's overall visual aesthetics and optimizing the customer shopping experience.

3.3 External Interface Requirements

In order to facilitate seamless interactions and integrations with various external components, the automated void space detection and planogram compliance verification system necessitates specific external interface requirements. These requirements encompass user interfaces, hardware interfaces, software interfaces, and communication interfaces to ensure the smooth operation and effective integration of the system within the existing retail environment.

3.3.1 User Interfaces

The system's user interface will be designed to offer a user-friendly and intuitive experience for store staff and management. It will feature a visually engaging dashboard displaying real-time shelf monitoring data, void space detection alerts, and product positioning verification indicators. The user interface will provide interactive functionalities for viewing detailed shelf analytics, accessing historical data, and managing notifications related to void spaces or product misplacements. The interface will be accessible via desktop ensuring flexibility and convenience for users to monitor and manage the store's inventory in real time.

3.3.2 Hardware Interfaces

The system will require compatible hardware components, including high-resolution cameras or image-capturing devices strategically positioned within the store's shelving units. These hardware interfaces will facilitate the continuous capture of clear and detailed images of the store shelves, enabling the system to conduct accurate void space detection and product positioning verification tasks. Additionally, the system will integrate with standard computing hardware to support the processing and analysis of the captured images, ensuring efficient and rapid data processing capabilities to deliver real-time monitoring and notifications.

3.3.3 Software Interfaces

The system will be designed to seamlessly integrate with the existing inventory management software and retail management systems used by the small retail shop. It will feature robust software interfaces that enable data exchange, synchronization, and compatibility with the store's inventory databases and product catalogs. The software interfaces will allow for the seamless transfer of data between the automated void space detection and planogram compliance verification system and the store's inventory management software, ensuring a synchronized and cohesive approach to stock management and inventory control.

3.3.4 Communication Interfaces

The system will rely on efficient communication interfaces to facilitate real-time data transmission and notification delivery between the system's components and the store staff or management. It will utilize secure communication protocols, such as HTTPS (Hypertext Transfer Protocol Secure) or MQTT (Message Queuing Telemetry Transport), to ensure the encrypted and reliable transfer of data across different devices and systems. The communication interfaces will support bi-directional data flow, enabling seamless interaction and information exchange between the system's monitoring module, user interface, and the store's personnel, thereby enabling timely interventions and corrective actions in response to any detected void spaces or product misplacements.

3.4 Nonfunctional Requirements

To ensure the optimal performance, safety, and security of the automated void space detection and planogram compliance verification system, the following nonfunctional requirements have been identified:

3.4.1 Performance Requirements

Real-time Processing: The system should process shelf images and data in real-time, ensuring prompt void space detection and product positioning verification with minimal latency.

Scalability: The system must be scalable to accommodate an increasing volume of shelf data and support the integration of additional retail outlets or chains without compromising performance.

Accuracy: The system's void space detection and product positioning verification algorithms must achieve a high level of accuracy, minimizing false positives and false negatives to ensure reliable and precise monitoring results.

3.4.2 Safety Requirements

Data Integrity: The system should ensure the integrity and accuracy of the captured shelf images and monitoring data, minimizing the risk of data corruption or loss during processing or transmission.

Equipment Safety: The hardware components integrated within the system, such as cameras or image-capturing devices, must comply with industry safety standards and regulations to prevent any potential hazards or risks to store personnel or customers.

3.4.3 Security Requirements

Data Encryption: The system should employ robust encryption techniques to secure the transmission and storage of sensitive data, preventing unauthorized access or data breaches.

Access Control: The system must implement stringent access control mechanisms, including user authentication and authorization protocols, to restrict access to sensitive functionalities and ensure that only authorized personnel can manage and monitor the system.

Data Privacy: The system should adhere to data privacy regulations and best practices, safeguarding the confidentiality of the store's inventory data and ensuring that customer information remains protected and anonymous throughout the monitoring and notification processes.

3.4.4 Software Quality Attributes

Accuracy: This refers to the system's ability to correctly detect void spaces and product positions on shelves. It is crucial for the system's effectiveness. Data can be collected through precision (percentage of correctly detected void spaces), recall (percentage of actual void spaces detected), and true positive/negative rates for product placement verification.

Real-time Performance: This attribute measures how quickly the system can process an image or video frame and provide results. It is important to ensure timely notifications and minimize delays. Data collection involves measuring frame processing time and latency between detection and notification.

Reliability: This refers to the system's consistency and dependability over time. A reliable system functions predictably with minimal errors. System uptime and mean time between failures (MTBF) can be measured to assess reliability.

Availability: This attribute reflects how often the system is accessible and operational for use. High availability ensures minimal downtime and ensures store staff can rely on the system. System downtime and uptime percentage are key metrics for availability.

Scalability: The system should be able to handle an increasing number of cameras or stores without significant performance degradation. Throughput with increasing camera load and resource utilization (CPU, memory) can be measured to assess scalability. As your project grows, this becomes increasingly important.

3.5 System Requirements

3.5.1 Database Requirements

The system will require a robust and scalable database management system capable of efficiently storing and managing a large volume of image data and metadata associated with the store's shelf layouts and product placements. The database should support fast data retrieval and querying capabilities to facilitate seamless integration with the system's real-time monitoring and notification functionalities. Additionally, the database should ensure data integrity and reliability, offering data backup and recovery mechanisms to prevent any potential data loss or corruption.

3.5.2 Software Requirements (Platform Choice)

The system will be developed using Python as the primary programming language, leveraging its extensive libraries and frameworks, including TensorFlow, PyTorch, and OpenCV, to facilitate advanced image processing and machine learning tasks. The system's user interface will be designed using HTML, CSS, and JavaScript to ensure a responsive and interactive dashboard accessible across different devices and platforms. Furthermore, the system will be deployed using the Flask web framework to enable seamless integration with the store's existing inventory management software and facilitate real-time data exchange and synchronization.

3.5.3 Hardware Requirements

The hardware components essential for the system's operation will include high-resolution cameras or image-capturing devices strategically positioned within the store's shelving units to capture detailed shelf images. Additionally, the system will rely on a robust computing infrastructure equipped with multi-core processors, ample RAM, and sufficient storage capacity to support the real-time processing and analysis of the captured images. The hardware configuration will prioritize processing speed and efficiency to enable rapid data processing and notification delivery, ensuring a seamless and uninterrupted monitoring experience for store personnel and management.

3.6 Analysis Models: SDLC Model to be applied

Given the dynamic nature of the project requirements and the need for continuous integration and adaptation, the Agile software development methodology will be adopted for the development and implementation of the automated void space detection and planogram compliance verification system.

The Agile approach will enable iterative development, frequent testing, and incremental enhancements, ensuring a flexible and adaptive development process that can readily accommodate evolving stakeholder requirements and incorporate feedback from end-users and store personnel.

By embracing the Agile methodology, the project team can foster a collaborative and cross-functional work environment, promoting effective communication, rapid decision-making, and a proactive response to changing market demands and technological advancements throughout the development lifecycle.

CHAPTER 4

SYSTEM DESIGN

4.1 System Architecture

4.2 Mathematical Model

he mathematical foundation of our project is grounded in the Information Bottleneck Principle and the innovative use of Reversible Functions, central to the design of YOLOv9. These principles address fundamental challenges in deep learning, ensuring efficient and accurate object detection performance.

- Information Bottleneck Principle: The Information Bottleneck Principle reveals the challenge of potential information loss as data propagates through successive layers of a network. Mathematically, it is represented as:

```
I(X, X) >= I(X, f_theta(X)) >= I(X, g_phi(f_theta(X)))
```

  Mutual information, denoted by I, quantifies the shared information between input data and its transformations through functions f and g, parameterized by theta and phi, respectively. YOLOv9 addresses this issue by introducing Programmable Gradient Information (PGI), which plays a vital role in maintaining critical data integrity throughout the network's layers. This mechanism enhances the reliability of gradient generation, leading to improved model convergence and overall performance.

  - Reversible Functions: Reversible Functions are those that can be inverted without loss of information, expressed as:

```
X = v_zeta(r_psi(X))
```

  with psi and zeta as parameters for the reversible and its inverse function, respectively. T

In our project, we integrate these principles into the architecture of YOLOv9 to ensure efficient and accurate object detection. Programmable Gradient Information (PGI) aids in preserving essential data across the network's depth, while reversible functions mitigate the risk of information degradation, particularly in deeper layers.These mathematical foundations enable YOLOv9 to maintain high efficiency and accuracy, even in lightweight models, ensuring reliable object detection performance across various applications.

4.3 Data Flow Diagrams

This provide a visual representation of the data transformation processes, highlighting the path of data from its initial capture on the store shelves to its analysis, processing, and presentation through the user interface for effective monitoring and management by store personnel.

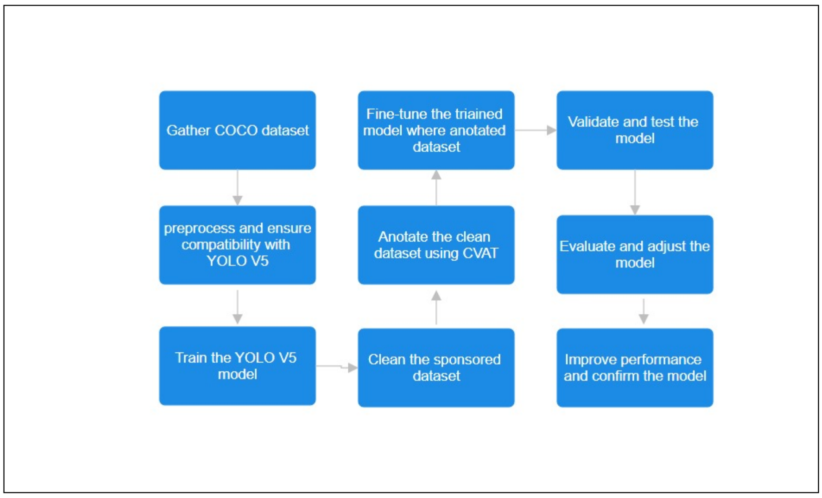

Figure 4.1: Data Flow Diagram

4.4 Entity Relationship Diagram

The entity relationship diagrams will delineate the relational structure and associations between the key data entities within the system, such as shelf layouts, product placements, and inventory databases. These diagrams will showcase the interconnected relationships between different data entities, emphasizing the dependencies and attributes that govern the system's data management and storage functionalities. By illustrating the data relationships and dependencies, the entity relationship diagrams will facilitate a comprehensive understanding of the data organization and management strategies implemented within the system.

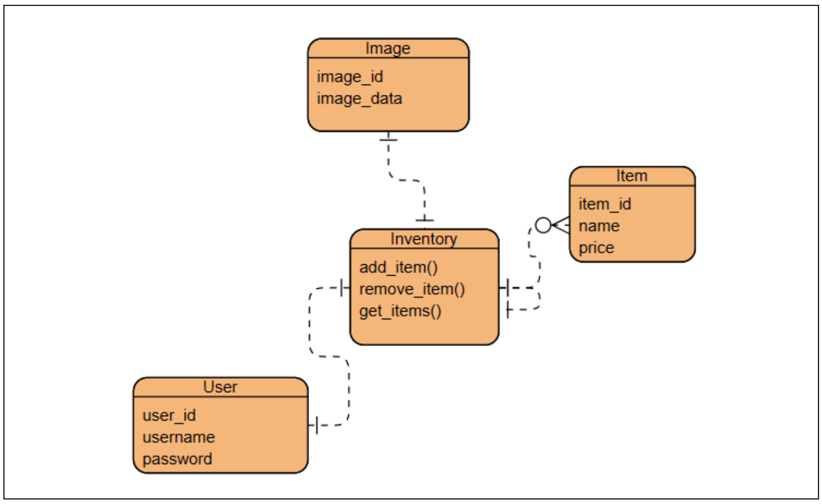

Figure 4.2: Entity Relationship Diagram

4.5 UML Diagrams

4.5.1 Use Case Diagram

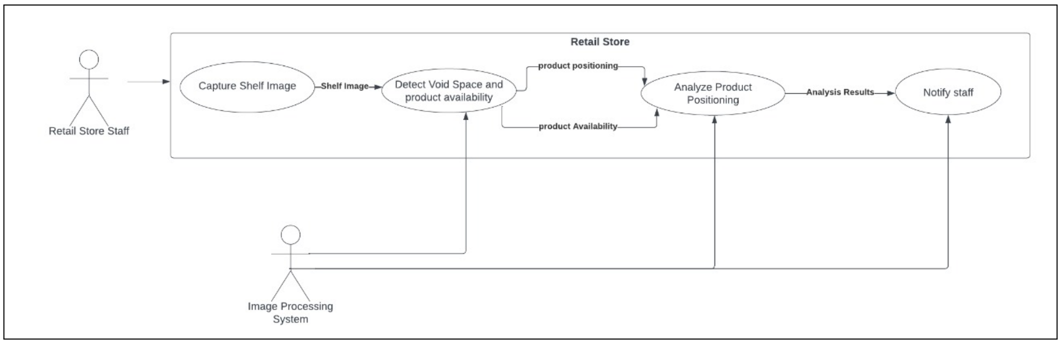

Figure 4.3: Use Case Diagram

4.5.2 Activity Diagram

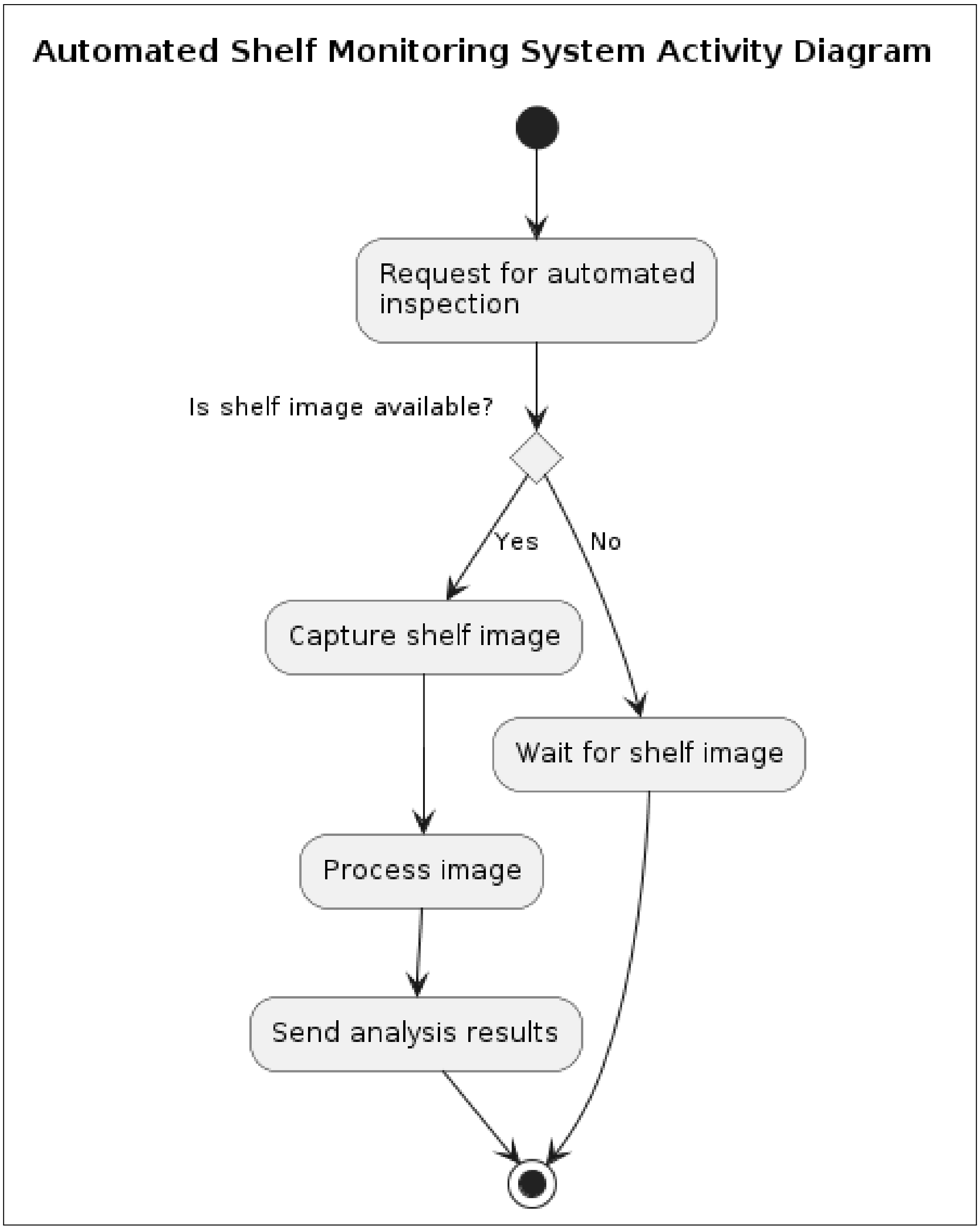

Figure 4.4: Activity Diagram

4.6 Sequence Diagram

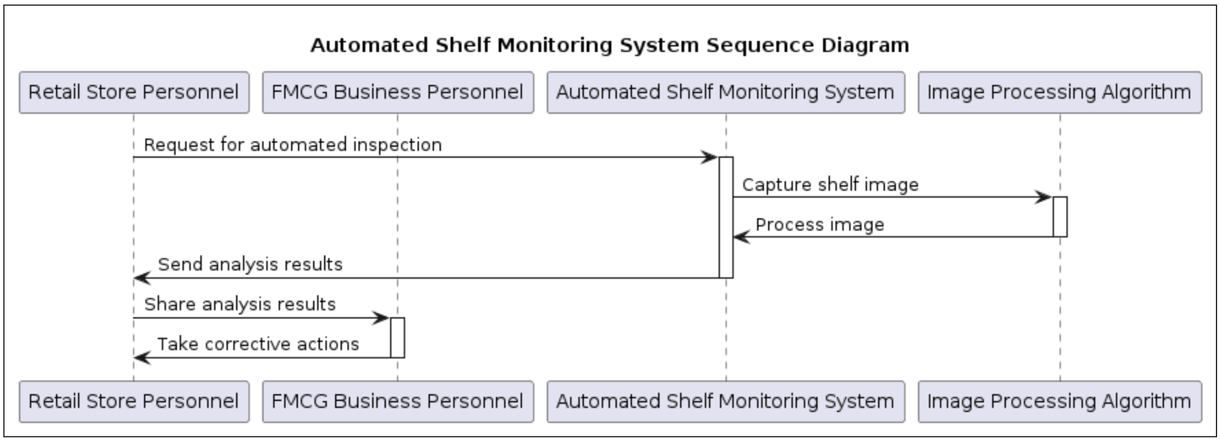

Figure 4.5: Sequence Diagram

Retail Store Personnel (Retailer): Initiates the process by requesting automated inspection of the shelves in the retail store.
Automated Shelf Monitoring System: Receives the request from the Retail Store Personnel and activates the monitoring process.
Image Processing Algorithm: Captures a snapshot of the shelf image using image processing techniques.
Automated Shelf Monitoring System: Receives the processed image from the Image Processing Algorithm and deactivates the image processing component.
Retail Store Personnel (Retailer): Receives the analysis results from the Automated Shelf Monitoring System.
FMCG Business Personnel (FMCG): Receives the analysis results from the Retail Store Personnel and takes corrective actions if necessary.

CHAPTER 5

PROJECT PLAN

5.1 Project Estimate

5.1.1 Reconciled Estimates

- Inventory Management: Reconciling inventory counts from different sources, such as manual counts, barcode scans, and sales transactions, to generate a more accurate estimate of stock levels. This could involve reconciling discrepancies between physical counts and system records to ensure inventory accuracy.

- Void Detection Using YOLOv9: Reconciling YOLOv9 void detection results with data from other sensors or sources to verify the presence or absence of products in specific areas of the store. By combining information from multiple sources, you can improve the accuracy and reliability of void detection.

- Product Positioning: Reconciling data on product positioning with sales data and customer behavior analytics to optimize product placement within the store. This could involve reconciling insights from store layout analysis with historical sales trends to identify the most effective product positioning strategies.

5.1.2 Project Resources

- Hardware: Inventory Management: CPU: Intel Core i5-X415 RAM: 32GB

- Model Training: GPU: Kaggle provided free GPU TPU: Not utilized

- Software and Development Tools: Operating System: Windows 11 IDE: Visual Studio Code (VSCode) Libraries: Python, Django, TensorFlow, PyTorch, Yolo Database Management: PostgreSQL,SQL

- Data Management:

  Datasets: We utilized Roboflow for preprocessing and managing datasets for inventory management and model training tasks. Database: PostgreSQL was used for storing and managing data, with a structured schema for efficient data organization.

- Model Training and Deployment:

  Model Training: Model training was conducted on Kaggle using GPU resources, with models trained using PyTorch frameworks. Model Deployment: The trained models were deployed on the Hugging Face platform for inference and integration into the application.

- Development Tools and IDEs:

  Development Tools: Visual Studio Code (VSCode), Django framework, Python programming language were utilized for application development. Version Control: Git was used for version control, allowing for collaborative development and tracking of code changes.

- Documentation and Collaboration:

  Documentation: Comprehensive project documentation, including user manuals, technical guides, and README files, was created to facilitate understanding and usage of the system. Collaboration: Communication and collaboration with stakeholders and team members were conducted using various platforms such as email,gmeet.

- Testing and Debugging:

  Testing Frameworks: Unit testing and integration testing were performed using frameworks such as pytest and TensorFlow/Keras testing utilities.

- Debugging Tools: Visual Studio Code (VSCode) debugger and logging utilities were utilized for identifying and resolving issues in the application code.

5.2 Risk Management

- Risk Management

5.2.1 Risk Identification and Analysis

- Technical Risks:
  Algorithm Complexity:
  The complexity of developing accurate image recognition algorithms can lead to delays and require additional resources.
  Impact: High
  Likelihood: Medium

- Integration Challenges: Difficulties in integrating image recognition solutions with existing retail systems and potential blockchain technology may cause disruptions.
  Impact: High
  Likelihood: Medium

- Data Quality Issues: Poor quality or insufficient training data for image recognition may result in suboptimal performance.
  Impact: Medium
  Likelihood: Low

- Operational Risks:
  Data Collection Challenges:
  Issues in collecting diverse and representative data from various retail environments may impact algorithm training effectiveness.
  Impact: Medium
  Likelihood: Medium

- Regulatory Compliance: Changes in data protection or retail compliance regulations may necessitate adjustments to the project approach.
  Impact: Medium
  Likelihood: Low

5.2.2 Overview of Risk Mitigation, Monitoring, Management

Risk Mitigation Strategies:
Algorithm Complexity:
To mitigate the complexity of developing image recognition algorithms, the team will conduct thorough feasibility studies before initiating development.

Integration Challenges:
Integration challenges with retail systems and potential blockchain technology will be mitigated through early integration testing, close collaboration with sponsoring company, and the development of contingency plans to address potential disruptions.

Data Quality Issues:
Mitigating data quality issues involves implementing data preprocessing techniques, regular validation and updates to training datasets, and ensuring diversity in data sources to enhance model robustness.

Risk Monitoring and Management:
Data Collection Challenges:
The team will actively monitor data collection progress against established milestones, regularly assessing the diversity and representativeness of collected data. Adjustments to data collection protocols and allocation of additional resources will be made as needed.

Scope Creep:
Ongoing monitoring of project scope will involve establishing clear boundaries and regularly reviewing project scope in comparison to initial specifications. Proposed scope changes will be documented, assessed, and communicated with stakeholders.

Market Dynamics:
Continuous monitoring of market trends and retail dynamics will inform adaptive strategies to respond to changing conditions. The project's modular approach allows for flexibility to accommodate future adjustments aligned with market needs.

Continued Risk Management:

Regulatory Compliance:
Rigorous monitoring involves staying updated on regulations, assessing project adherence to standards, and incorporating compliance checkpoints in the project timeline.

Skill Gaps:
Regular assessment of the team's skillset against project requirements will guide continuous learning and skill development initiatives.

Communication Breakdown:
Monitoring communication effectiveness involves regular assessments of channels and fostering open communication within the team. Implementation of collaboration tools and regular team meetings will ensure effective communication.

Continuous Improvement:

Equipment Failure:
Continuous monitoring includes regular checks and maintenance to prevent equipment failures. Protocols for swift response to issues and the availability of backup equipment during critical phases are integral to risk management.

5.3 Project Schedule

5.3.1 Project Task Set

- **Task 1: Project Initiation (1 week)**
  - Define project goals and objectives.
  - Formulate the project team.
  - Set up communication and collaboration tools.
- **Task 2: Requirement Analysis (2 weeks)**
  - Gather project requirements and user stories.
  - Collect or create dataset of registered and input images.
  - Define software specifications such as which algorithms are best for the problem statement.
  - Identify functional and non-functional requirements.
- **Task 3: Design Phase (2 weeks)**
  - UI/UX Design.
  - Data Model Design.
- **Task 4: Development Phase (8 weeks)**
  - Implement Frontend using Streamlit.
  - Develop Data Processing.
  - Train Models using various ML algorithms like ORB, SIFT, CNN and ResNet50.
  - Implement an Authentication System.
  - Integrate ML Models.
- **Task 5: Testing and Quality Assurance (4 weeks)**
  - Address and resolve any bugs or issues.
  - Ensure the system is functioning as expected.
  - Perform testing.

5.3.2 Task Network

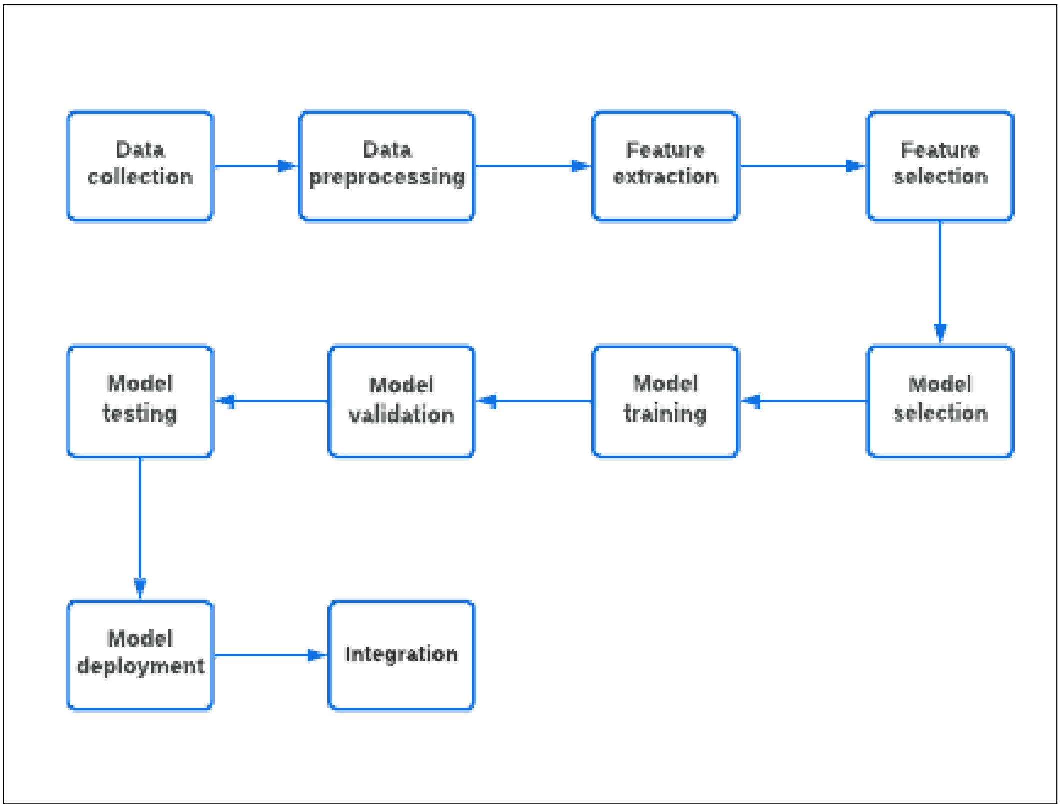

Figure 5.1: Task Network Diagram

5.3.3 Timeline Chart

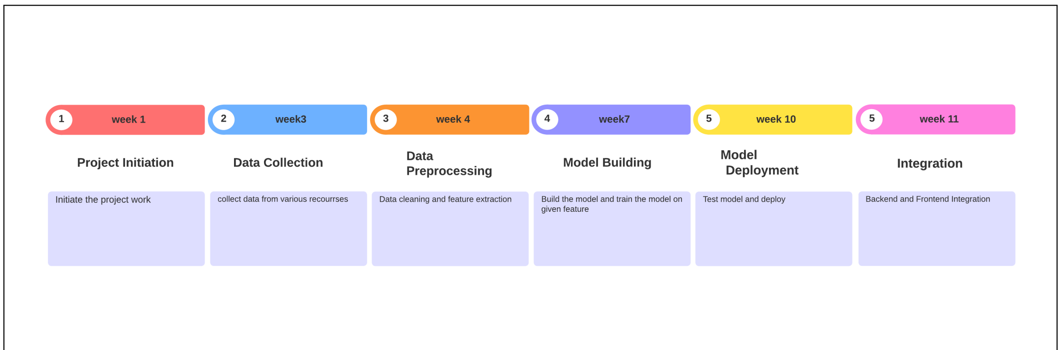

Figure 5.2: Timeline chart

5.4 Team Organization

5.4.1 Team structure

1. **Dataset Collection and Preprocessing**
   - Acquire relevant datasets for training and testing the hand gesture recognition models.
   - Clean and preprocess the datasets to ensure consistency and quality for machine learning tasks.
   - Proficient in data acquisition techniques.
   - Experience with data preprocessing tools and libraries (e.g., OpenCV, NumPy).

2. **Frontend Developer**
   - Design and develop the UI for the empty shelf detection system.
   - Implement responsive and user-friendly frontend components for interacting with the system.
   - Proficiency in frontend technologies (HTML/CSS, JavaScript).
   - Experience with frontend frameworks (e.g., React, Streamlit) for building interactive UIs.

3. **ML Specialist**
   - Develop and deploy machine learning models for empty shelf detection system.
   - Optimize and fine-tune models based on performance metrics and user feedback.
   - Experience with deep learning frameworks (e.g., TensorFlow, PyTorch) for model development.

4. **Database Connectivity and Integration**
   - Implement database connectivity to store user profiles, authentication data, and system logs.
   - Ensure secure and efficient data storage and retrieval for user authentication.
   - Proficient in database management systems (e.g., SQL, NoSQL, SQLite).
   - Experience with backend development and API integration for database operations.

5.4.2 Management reporting and communication

1. **Team Collaboration and Coordination:**
    - Utilize project management tools (e.g., Jira, Trello) for task assignment and tracking progress.
    - Conduct regular team meetings to discuss project updates, challenges, and milestones.
    - Foster open communication and collaboration among team members to ensure project alignment.
2. **Reporting and Documentation:**
    - Maintain detailed documentation of project requirements, designs, and implementation details.
    - Document lessons learned and best practices for future reference and knowledge sharing.
3. **Communication Channels:**
    - Establish clear communication channels (e.g., Microsoft Teams) for instant messaging and team discussions.
    - Encourage feedback and suggestions from team members to improve project efficiency and quality.

CHAPTER 6

PROJECT IMPLEMENTATION

6.1 Overview of Project Modules

- User Interface (UI) Module: The User Interface module serves as the front-end component of the application, providing users with an interactive platform to interact with the system. It is implemented using a combination of React.js and Django, offering a seamless user experience across different devices and browsers. The UI module facilitates user authentication, input data submission, and visualization of results.

- Data Processing Module: The Data Processing module is responsible for preprocessing and annotating datasets before they are used for model training and inference. It includes functionalities for data cleaning, augmentation, and labeling, ensuring that the input data is optimized for training the YOLOv9 algorithm. This module also handles the integration of annotated datasets with the algorithm for model training.

- Algorithm Module (YOLOv9): The Algorithm module comprises the YOLOv9 algorithm, which is used for object detection and recognition tasks within the application. YOLOv9 is a state-of-the-art deep learning model known for its efficiency and accuracy in real-time object detection. This module handles model training, inference, and integration with the User Interface module for displaying detection results to users.

- Database Module (SQLite): The Database module utilizes SQLite as the database management system for storing and managing application data. SQLite offers lightweight and efficient storage capabilities, making it suitable for small to medium-sized applications like ours. It stores user information, annotated datasets, detection results, and other relevant data entities.

- Reporting and Integration Module: The Reporting and Integration module facilitates the generation of reports and the integration of the application with external services such as WB (Weights Biases) and GitHub. Reports are generated based on processed data and detection results, providing insights and visualizations to users. Integration with services like WB and GitHub enables collaboration, version control, and performance tracking during the development and deployment phases.

6.2 Tools and Technologies Used

- 1. Development and Model Training:

  Python: Programming language used for implementing machine learning algorithms.

  PyTorch or TensorFlow: Deep learning frameworks for building and training neural networks.

  YOLOv9: Real-time object detection model for void space detection and product positioning verification.

  OpenCV: Library for image and video processing tasks.

  LabelImg or CVAT: Annotation tools for labeling images and annotating the dataset.

- 2. System Integration and Deployment:

  Django: Web frameworks for integrating the model into a user-friendly interface.

  HTML, CSS, JavaScript,ReactJs: Front-end languages for designing and developing the user interface.

  huggingFace: services for scalable and reliable deployment.

- 3. Testing and Quality Assurance:

  Selenium: Testing framework for web application testing.

- 4. Security and Privacy Tools:

  SSL/TLS Certificates: To ensure secure data transmission over the network.

  OAuth or JWT: Security protocols for authentication and authorization.

- 5. Monitoring and Analytics:

  Grafana: Tools for monitoring and visualizing system performance and metrics.

- 6. Version Control and Collaboration: Git and GitHub: Version control systems for managing and tracking changes in the project code.

Kaggle and Google Colab: Platforms for collaborative coding and sharing

6.3 Algorithm Details

6.3.1 Inventory Management System for Vendor

1. Input:

- Vendor details (vendor ID, name, contact information).
- Product delivery details (delivery ID, delivery date, list of products).
- Order details (order ID, list of products, quantity).
- Product details (product ID, name, category, brand, dimensions).

2. Process:

- Vendor adds product delivery information, specifying the delivery date and the list of products being delivered.
- Vendor places orders by specifying the products and quantities needed.
- Vendor adds new products to the inventory by providing product details such as name, category, brand, and dimensions.
- Orders placed by the vendor are forwarded to the warehouse for processing and fulfillment.

3. Output:

- Confirmation messages for successful addition of delivery, order placement, and product addition.
- Forwarded orders to the warehouse for fulfillment.

6.3.2 YOLOv9 Object Detection Algorithm

1. Input:

- Input image or video frame. 2. Process:
- Preprocess the input image or frame to prepare it for object detection.

- Pass the preprocessed image through the YOLOv9 model for object detection.
- Identify objects within the image by detecting bounding boxes and class labels.
- Filter the detected objects based on confidence scores and predefined thresholds.
- Return the detected objects along with their bounding boxes and class labels as output.

  3. Output:

- Detected objects with bounding boxes and class labels.

6.3.3 Real-time CCTV Void Detection and Image Capture

1. Input:

- Video stream from CCTV cameras. 2. Process:
- Continuously capture frames from the CCTV video stream.
- Process each frame to detect voids using the YOLOv9 object detection algorithm. If a void is detected in a frame, capture the image containing the void area.
- Send the captured images containing voids to the void detection model for analysis.
- Display the results of void detection, indicating whether a void is present in the captured images. 3. Output:
- Results of void detection, indicating whether voids are present in the captured images.

CHAPTER 7

SOFTWARE TESTING

7.1 Type of Testing

1. **Unit Testing:**

   - **Description:** Testing individual components or functions of the system to ensure they perform as expected.
   - **Purpose:** Identify and fix bugs in isolated components and validate their behavior.
   - **Example:**
     [label=○]Component: Hand gesture feature extraction function. Test Case: Verify that the function correctly extracts hand landmarks from input images and returns expected feature vectors.

2. **Integration Testing:**

   - **Description:** Testing the interaction and integration between different modules/components of the system.
   - **Purpose:** Ensure that integrated components work together seamlessly.
   - **Example:**
     [label=○]**Scenario:**Integration between the frontend UI and backend authentication system. **Test Case:** Validate that user actions (e.g., hand gesture inputs) from the UI trigger appropriate authentication processes in the backend.

3. **System Testing:**

   - **Description:** Testing the entire system as a whole to validate its functionality against specified requirements.
   - **Purpose:** Assess the system's behavior under different use cases and scenarios.
   - **Example:**
     [label=○]**Scenario:** User authentication using hand gestures. **Test Case:** Simulate various hand gestures and verify that the system correctly identifies and authenticates users based on predefined gestures.

4. **Performance Testing:**

- **Description:** Evaluating the system's responsiveness, scalability, and resource usage under different load conditions.
- **Purpose:** Ensure that the system performs efficiently and reliably under expected and peak loads.
- **Example:**
  [label=○]**Load Test:** Simulate multiple concurrent users performing hand gestures for authentication. **Metric:** Measure the system's response time and resource utilization to ensure it meets performance requirements.

5. **Security Testing:**

- **Description:** Identifying vulnerabilities and weaknesses in the system's security mechanisms.
- **Purpose:** Mitigate security risks and ensure robust authentication and data protection.
- **Example:**
  [label=○]**Scenario:** Verify user authentication security. **Test Case:** Attempt unauthorized access using invalid hand gestures and ensure the system rejects unauthorized attempts while granting access to valid users.

7.2 Test cases & Test Results

Test Case Description	**Expected Result**	**Actual Result**	**Pass/Fail**
Retail personnel initiates a request for automated inspection.	Monitoring system receives the request and begins the inspection process.	Monitoring system receives the request and begins the inspection process.	Pass
Image processing algorithm captures a snapshot of the shelf.	An image of the shelf is successfully captured.	An image of the shelf is successfully captured.	Pass
Image processing algorithm analyzes the captured image to identify products, empty spaces, and correct product orientations.	The image processing algorithm accurately detects products, empty spaces, and their orientations.	The image processing algorithm accurately detects products, empty spaces, and their orientations.	Pass
Analysis results are transmitted from the monitoring system to the retail personnel.	Retail personnel receive the analysis results, including information about product availability, shelf arrangement, and any corrective actions needed.	Retail personnel receive the analysis results, including information about product availability, shelf arrangement, and any corrective actions needed.	Pass

Table 7.1: Test Cases and Results for Void-space Detection

CHAPTER 8

RESULTS

8.1 Results

8.1.1 Training Process

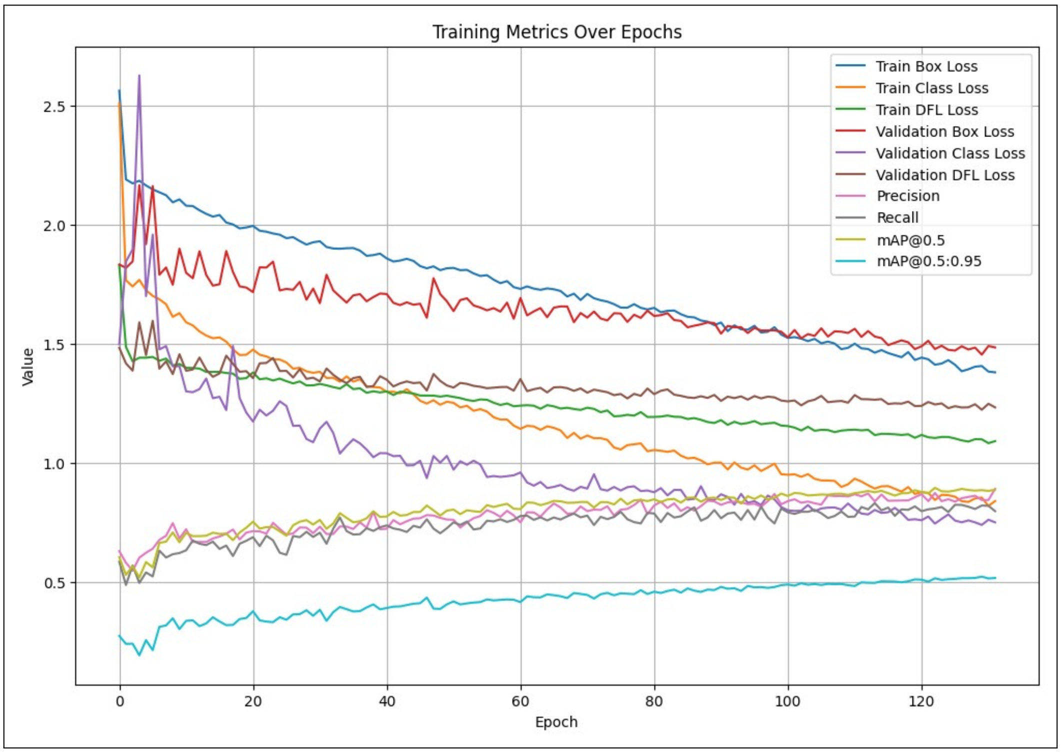

Figure 8.1: Training Metrics

- Number of Epochs:

  The YOLOv9 model was fine-tuned for a total of 131 epochs. This extended training duration allowed the model to learn intricate patterns and details within the dataset, potentially improving its ability to detect objects accurately.

8.1.2 Loss Functions

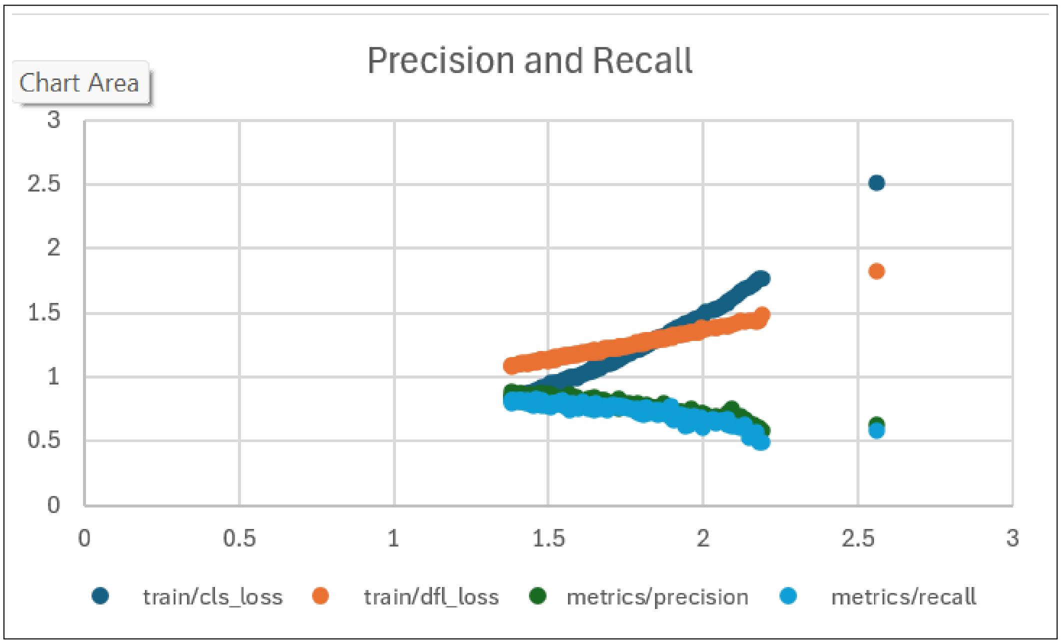

Figure 8.2: Precision and Recall

Two primary loss functions were utilized during the fine-tuning process:
1. Box Loss: This loss function measures the discrepancy between the predicted bounding box coordinates and the ground truth bounding box coordinates. Minimizing this loss encourages the model to accurately predict the locations of objects in the image.
2. Class Loss: The class loss evaluates the difference between the predicted class probabilities and the true class labels associated with each bounding box. Minimizing this loss helps the model learn to classify objects correctly.

8.1.3 Other Relevant Training Parameters

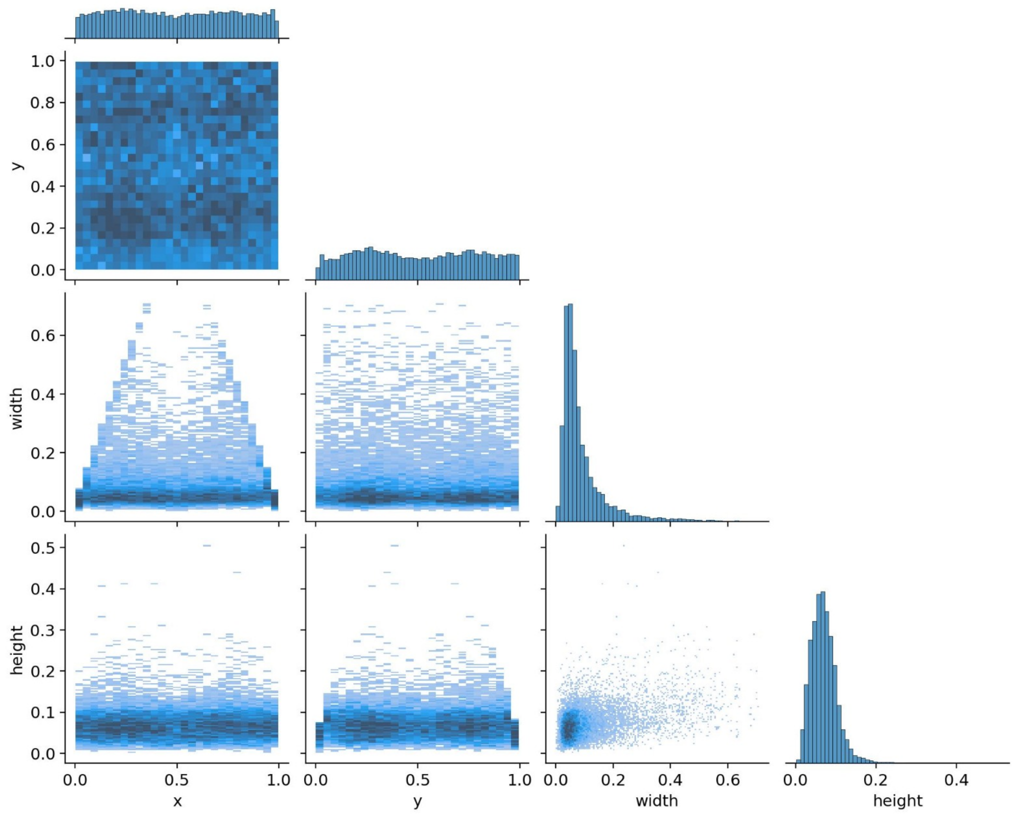

Figure 8.3: Input Data Pattern

Other Relevant Training Parameters:
1.Learning Rate: The learning rate determines the step size at which the model updates its parameters during training. It's essential to strike a balance between a learning rate that's too high (which may cause unstable training) and one that's too low (which may result in slow convergence). Experimentation with different learning rates may have been conducted to identify an optimal value.

2.Batch Size: The batch size specifies the number of training examples processed in each iteration before updating the model's parameters. A larger batch size can lead to faster training but may require more memory.

3.Data Augmentation: Techniques such as random cropping, flipping, and rotation may have been employed to augment the training data, thereby enhancing the model's robustness and generalization ability.

4.Optimizer: The choice of optimizer, such as Adam or SGD (Stochastic Gradient Descent), plays a crucial role in determining how the model parameters are updated during training.

8.1.4 Observed Trends

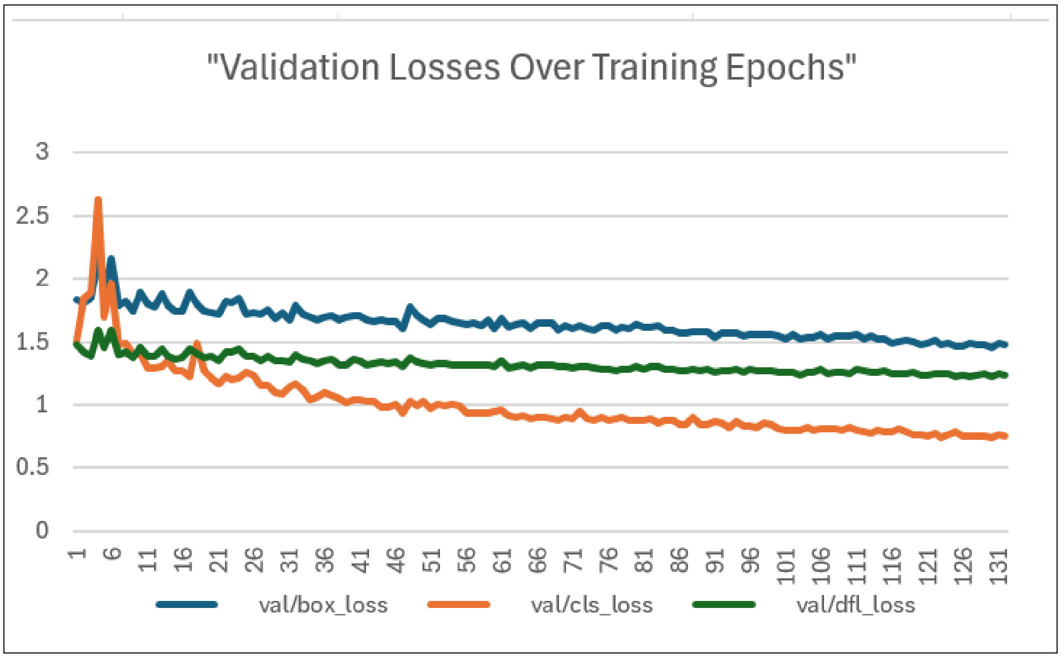

Figure 8.4: Validation losses

During the training process, a steep curve for the error loss was observed initially, followed by stabilization around the 50th to 70th epochs. This trend suggests that the model rapidly improved in the early stages of training as it learned to detect objects more accurately. Subsequently, the rate of improvement slowed down, indicating that the model approached convergence and reached a stable performance level.

Understanding these trends is essential for assessing the model's training progress and determining when to halt training to prevent overfitting or unnecessary computational costs.

This line chart illustrates the variation of different loss components (val/box, val/clsloss, val/dfloss) during the training process. The losses are computed based on the validation dataset, providing insights into the model's performance in terms of localization (val/box), classification (val/clsloss), and regression (val/dfloss) tasks. The x-axis represents the training epochs, while the y-axis represents the corresponding loss values. Analyzing these loss trends can help in assessing the effectiveness of the model training and identifying potential areas for improvement.

8.2 Outcomes

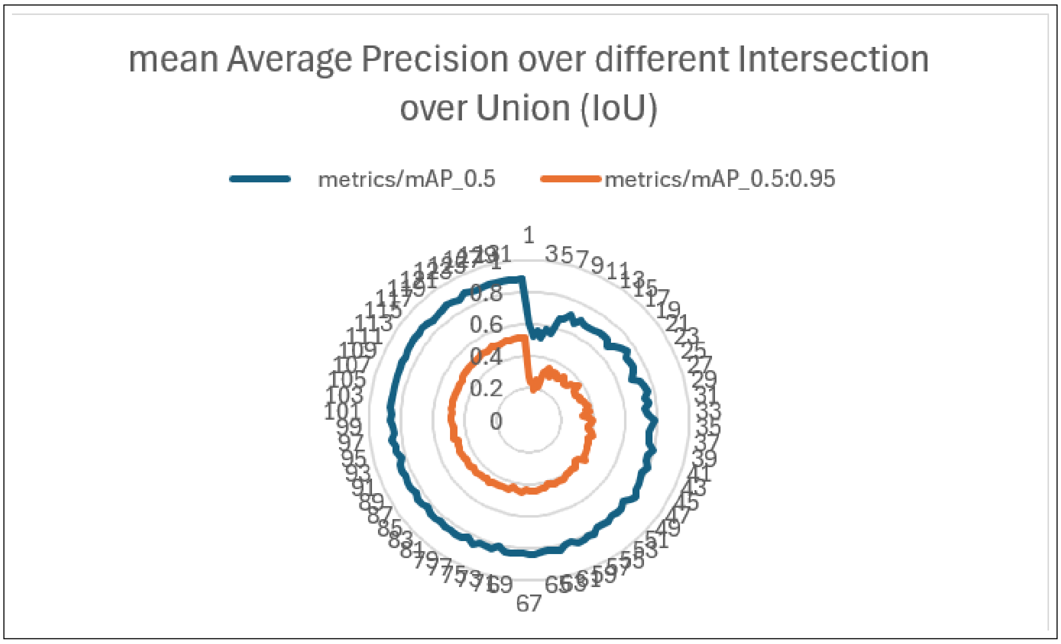

Figure 8.5: mean Average Precision over different Intersection over Union (IoU)

1. **Cost Savings:** By automating the inspection process, retailers and FMCG companies can reduce the expenses associated with visual checks and human errors.

2. **Improved Accuracy:** The image processing algorithms can provide more accurate and consistent monitoring of product placement and shelf visibility, compared to manual inspections.

3. **Enhanced Shelf Management:** The ability to identify empty spaces and verify product orientation enables better shelf management, ensuring optimal product visibility and availability.

4. **Increased Sales:** By maintaining proper product placement and availability, the automated shelf inspection system can contribute to increased sales for both retailers and product manufacturers.

8.3 Screen Shots

8.3.1 Training notebook

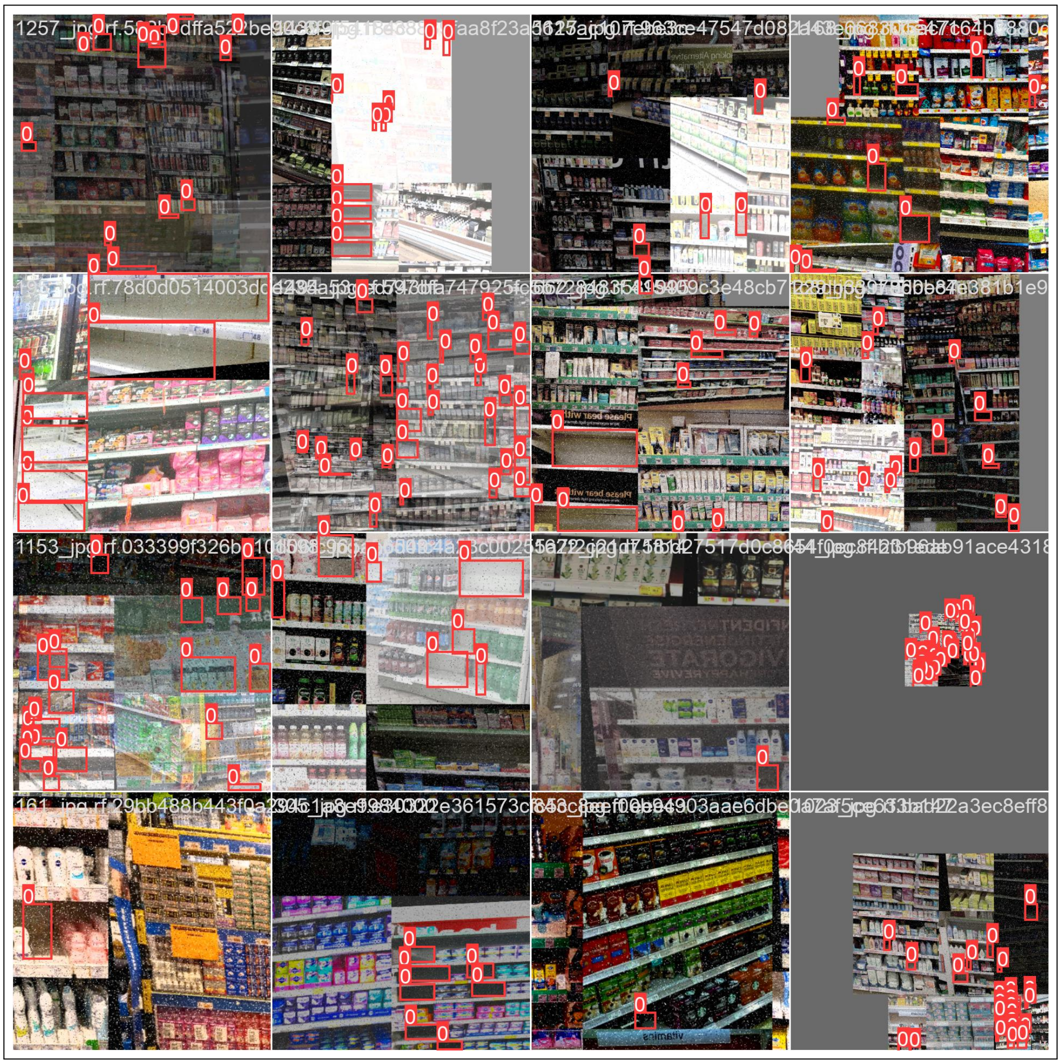

Figure 8.6: training-dataset

This screenshot displays a segment of the training dataset used for training the YOLOv9 object detection model. The dataset consists of annotated images representing various scenes, with bounding boxes drawn around objects of interest. In this particular subset of the dataset, labeled as "s 0," annotations for void regions are highlighted.

Figure 8.7: Results in notebook

This screenshot showcases the output of the trained YOLOv9 object detection model applied to a set of test images. Each resulting image displays the original scene overlaid with bounding boxes representing the model's predictions for detected objects.

8.3.2 User Interface

This screenshot depicts the user interface (UI) of the application/software under development. The UI design aims to provide a seamless and intuitive user experience, facilitating interaction and navigation within the application.

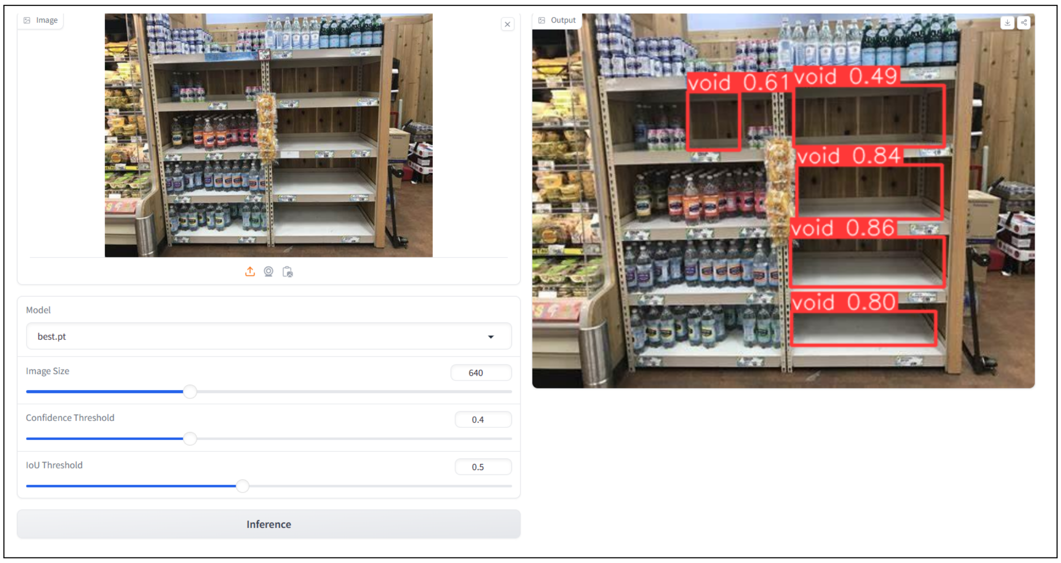

Figure 8.8: Screenshot of UI

CHAPTER 9

CONCLUSIONS

9.1 Conclusions

The development and implementation of the automated void space detection and planogram compliance verification system represent a significant advancement in the realm of retail stock management and customer-centric store operations. By leveraging cutting-edge machine learning algorithms and real-time monitoring capabilities, the system offers a robust solution for small retail businesses to streamline their inventory control processes, minimize stock discrepancies, and ensure consistent planogram compliance. The project's successful execution underscores the potential of integrating advanced technology with traditional retail practices, emphasizing the importance of data-driven insights and proactive interventions to foster a seamless and engaging shopping experience for customers. Through the collaborative efforts of the project team and the valuable insights gained from stakeholder engagement, the project has laid a strong foundation for future advancements in retail automation and the continued evolution of stock management practices within the retail industry.

9.2 Future Work

While the current implementation of the automated void space detection and planogram compliance verification system demonstrates its efficacy and practical value in small retail environments, several avenues for future research and development warrant exploration. These include : Enhanced Image Recognition Algorithms: Further research into advanced image recognition algorithms and deep learning models can refine the system's void space detection and product positioning verification capabilities, enhancing its accuracy and adaptability to complex shelf layouts and product arrangements. Integration of IoT Devices: The integration of Internet of Things (IoT) devices and sensors can enable a more comprehensive and interconnected approach to retail automation, facilitating real-time data collection and analysis for predictive stock replenishment and dynamic planogram adjustments based on customer foot traffic and product demand. Intelligent Inventory Forecasting: By incorporating predictive analytics and demand forecasting models, the system can anticipate future stock requirements and customer preferences, enabling proactive inventory management strategies and minimizing the occurrence of stockouts or overstock situations. The pursuit of these avenues for future work holds the potential to further revolutionize the retail landscape, empowering small retail businesses to embrace data-driven decision-making and automation technologies to deliver enhanced customer experiences and drive sustainable business growth.

9.3 Applications

The automated void space detection and planogram compliance verification system finds diverse applications across various retail environments, offering comprehensive solutions for efficient stock management and improved store operations. Some of the key applications of the system include:

- Small Retail Shops and Convenience Stores: The system is well-suited for small-scale retail businesses and convenience stores seeking to optimize their stock management processes and ensure the accurate placement of products on the shelves, enhancing the overall shopping experience for customers within a limited physical space.

- Supermarkets and Hypermarkets: In larger retail establishments, such as supermarkets and hypermarkets, the system serves as a valuable tool for maintaining planogram compliance across multiple product categories and store sections, facilitating efficient inventory control and contributing to the seamless navigation of customers throughout the store.

- Pharmacies and Specialty Stores: Within pharmacies and specialty stores, the system's accurate void space detection capabilities assist in preventing stockouts of essential medications and products, ensuring the availability of critical items for customers and promoting a well-organized and easily accessible store layout.

- Department Stores and Chain Retail Outlets: In the context of department stores and chain retail outlets, the system's scalability enables centralized inventory management and planogram compliance verification across multiple store locations, fostering a standardized approach to stock replenishment and product placement that aligns with the brand's visual merchandising guidelines and customer experience standards.

By acknowledging these advantages, limitations, and diverse applications, stakeholders and decision-makers can effectively evaluate the system's feasibility and potential impact on enhancing stock management practices and delivering an enhanced shopping experience for customers across various retail settings.

CHAPTER 10

APPENDIX

10.1 Appendix A

In this appendix, we delve into the feasibility assessment of our project's problem statement using satisfiability analysis and complexity classification. We apply concepts from modern algebra and relevant mathematical models to evaluate the computational complexity and feasibility of automating retail store inventory management, with a focus on void detection, correct product positioning, and overall inventory management optimization.

1. Recap of Problem Statement: Our project aims to automate the inventory management process in retail stores by addressing three main objectives: - Void Detection: Identifying areas in the store where products are out of stock or missing. - Correct Product Positioning: Ensuring that products are placed in their designated positions on shelves or display racks. - Inventory Management Optimization: Optimizing the overall inventory management process to minimize stockouts, overstocking, and operational inefficiencies.

2. Satisfiability Analysis: We approach the problem statement as a logical formula representing the constraints and objectives of inventory management. By conducting satisfiability analysis, we assess the feasibility of satisfying these constraints and objectives through automated algorithms and methodologies.

3. Complexity Classification: - P Type: We aim to design algorithms and methodologies that can solve the problem within polynomial time, making it tractable and feasible for practical implementation. - NP-Hard and NP-Complete: Given the combinatorial nature of inventory management and the potential for exponential growth in problem instances, certain aspects of the problem, such as optimal inventory allocation and routing, may fall into the NP-hard or NP-complete complexity classes. These areas may require approximation algorithms or heuristic approaches for efficient solution finding.

4. Utilization of Modern Algebra and Mathematical Models: We leverage concepts from modern algebra, such as group theory, ring theory, and field theory, to formalize the problem statement and analyze its properties mathematically. Additionally, we employ relevant mathematical models, such as optimization models, graph theory, and combinatorial algorithms, to devise efficient algorithms and approaches for solving specific aspects of the inventory management problem.

5. Conclusion: Through the feasibility assessment conducted in this appendix, we gain insights into the computational complexity of automating retail store inventory management. By leveraging modern algebra and mathematical models, we establish a rigorous framework for analyzing the problem's complexity and devising effective solutions. This assessment lays the groundwork for the subsequent development and implementation of algorithms and methodologies to address the project's objectives.

10.2 Glossary

- 1. Object Detection: Object detection is a computer vision task that involves identifying and locating objects of interest within an image or video frame.
- 2. YOLO (You Only Look Once): YOLO is a popular object detection algorithm that divides an image into a grid and predicts bounding boxes and class probabilities for each grid cell simultaneously. YOLO is known for its speed and accuracy.
- 3. YOLOv9: YOLOv9 is an improved version of the YOLO (You Only Look Once) object detection model. It includes enhancements such as a more advanced architecture, improved training techniques, and better performance.
- 4. Model: In the context of machine learning, a model is a mathematical representation of a real-world process or phenomenon. In object detection, the model is trained to identify and locate objects within an image or video.
- 5. Confidence Threshold: The confidence threshold is a parameter that determines the minimum confidence score required for a detected object to be considered valid. Objects with confidence scores below this threshold are typically discarded.
- 6. IoU (Intersection over Union) Threshold: The IoU threshold is a parameter used to evaluate the overlap between predicted bounding boxes and ground truth bounding boxes. It measures the intersection of the predicted and ground truth bounding boxes divided by their union.

- 7. Bounding Box: A bounding box is a rectangular box that encloses an object detected by an object detection model. It is defined by its coordinates (x, y) and dimensions (width, height).

- 8. Interval: In the context of video processing, the interval refers to the time duration between consecutive frames or screenshots captured from the video.

- 9. Image Size: The image size refers to the dimensions (width and height) of the input images processed by the object detection model. Larger image sizes may result in better detection accuracy but require more computational resources.

- 10. Input: In the context of Gradio, an input refers to the data provided to a machine learning model for processing. Inputs can include images, videos, text, sliders, buttons, etc.

- 11. Output: In the context of Gradio, an output refers to the result or prediction generated by a machine learning model based on the input data. Outputs can include images, text, numerical values, etc.

10.3 Appendix B

Paper has been successfully published in the JETIR International Journal(ISSN: 2349-5162) and presented in Impetus and Concepts (InC) ,a flagship technical event of SCTR's Pune Institute of Computer Technology (PICT), Pune , which was held during the 1st week of April 2024.

- Registration ID : JETIR533935
- Paper ID: JETIR2403136
- Paper Title: Applying Image Processing to Determine Product Placement and Availability on the Shelf in Retail Stores
- Authors: Sakshi Rathi, Pratik More, Harsh Dhawale, Sanket Jhavar, Bhumesh Masram
- JETIR Impact Factor: 7.95
- Unique Contents:85

© 2024 JETIR March 2024, Volume 11, Issue 3 www.jetir.org (ISSN-2349-5162)

Applying Image Processing to Determine Product Placement and Availability on the Shelf in Retail Stores

Harsh Dhawale*
Computer Engineering
Pune Institute of Computer Technology
Pune, India

Sanket Jhavar*
Computer Engineering
Pune Institute of Computer Technology
Pune, India

Pratik More*
Computer Engineering
Pune Institute of Computer Technology
Pune, Maharashtra

Sakshi Rathi*
Computer Engineering
Pune Institute of Computer Technology
Pune, India s

Bhumesh Masram[†]
Assistant Professor
Pune Institute of Computer Technology
Pune, India

***Abstract*—The absence of product availability and the im- proper arrangement of items on the shelves in retail stores can lead to reduced sales for the retailer. personnel of Fast-Moving Consumer Goods (FMCG) businesses, whose products are stocked on these shelves, as well as personnel of the retailer undertake visual checks to detect products that are lost or out of stock are expensive and tends to commit mistakes. A technique for automating the manual inspection process is presented in this study. The research also shows that it is possible to identify empty spaces on the shelves in addition to recognizing and counting products that face front by using image processing algorithms. Moreover, this approachcan be applied to images captured from a video stream, such as those from security cameras, to count the number of visible units of a particular product on a shelf and determine whether theyare correctly oriented, as required. The image processing methodproposed in this paper primarily facilitates the proper positioningof products on the front row of the shelves. While this might appear to be a limitation in terms of inventory management,it holds substantial significance for product manufacturers who lease shelf space and allocate specific shelf positions, compelling retailers to place particular products in designated locations. The innovative aspect of this paper lies in the extension of feature extraction in image processing to highlight and rectify incorrect placements and arrangements of items on the shelves. Importantly, the implemented solution does not entail significant additional infrastructure costs.**

I. Introduction

The term "On-Shelf Availability" (OSA) refers to the measurement of products that are available to customers at the expected location, at the time of their choosing, and in a condition suitable for sale[7]. On average, retailers allocate 5% of their sales revenue to logistics, with the majority of these costs stemming from in-store inventory handling (38%) and storage (7%) [1]. Consequently, improving themanagement of in-store inventory is crucial for enhancinga retailer's profitability. Both the retailer and the productmanufacturer face an opportunity cost associated with OSA.Hence, maintaining OSA or reducing instances of Out-of-Stock (OOS) serves as an indicator of a retailer's performance.

Out-of-Stocks (OOS), the counterpart of OSA, occur whena consumer visiting a retail store finds that the specific productthey are looking for is unavailable on the shelf [2].

When an OOS situation arises, customers react in the following ways [7]: • 31% purchase the needed product from an alternative source (a different store or online) • 26% choose a different brand • 19% still buy the same brand but opt for a different variant, size, or flavor • 15% delay their purchase to a later date • 9% leave the store without making a purchase.

A loss of EUR 4 billion is incurred by the 9% of consumers who choose not to make a purchase [9]. With so many options nowadays, a client may decide to permanently move to a different store if OOS problems happen frequently. A "loyal" family's weekly revenue loss from leaving a particular store is about EUR 150 , which adds up to EUR 150,000 in lost revenue over a 20-year period [8]. Furthermore, 20% of all OOS cases are not handled for three days [7], which has a substantial negative impact on retailer profitability and performance [6].

For manufacturers, a 3% increase in OSA equates to a 1% increase in sales, while for retailers, a 2% increase in OSA results in a 1% sales boost [7]. Therefore, retail establish- ments must ensure high OSA to retain their loyal customer base. Accumulating excess inventory is not a viable solution, as it can lead to losses due to product wastage. Moreover,even if a product is available in the inventory, inefficient shelfmanagement may prevent it from reaching the shelf [6].

Retailers are now using a variety of strategies to guarantee a high degree of On-Shelf Availability (OSA) for their merchandise. Performing store audits, which entail manual visual checks of store shelves, is one of these strategies. However, because the assessment is dependent on the auditor's visit schedule, which might be arbitrary, this labor-intensive method cannot reliably assess Out-of-Stock (OOS) problems [3]. Moreover, this approach might not be economical in areas with high labor costs.

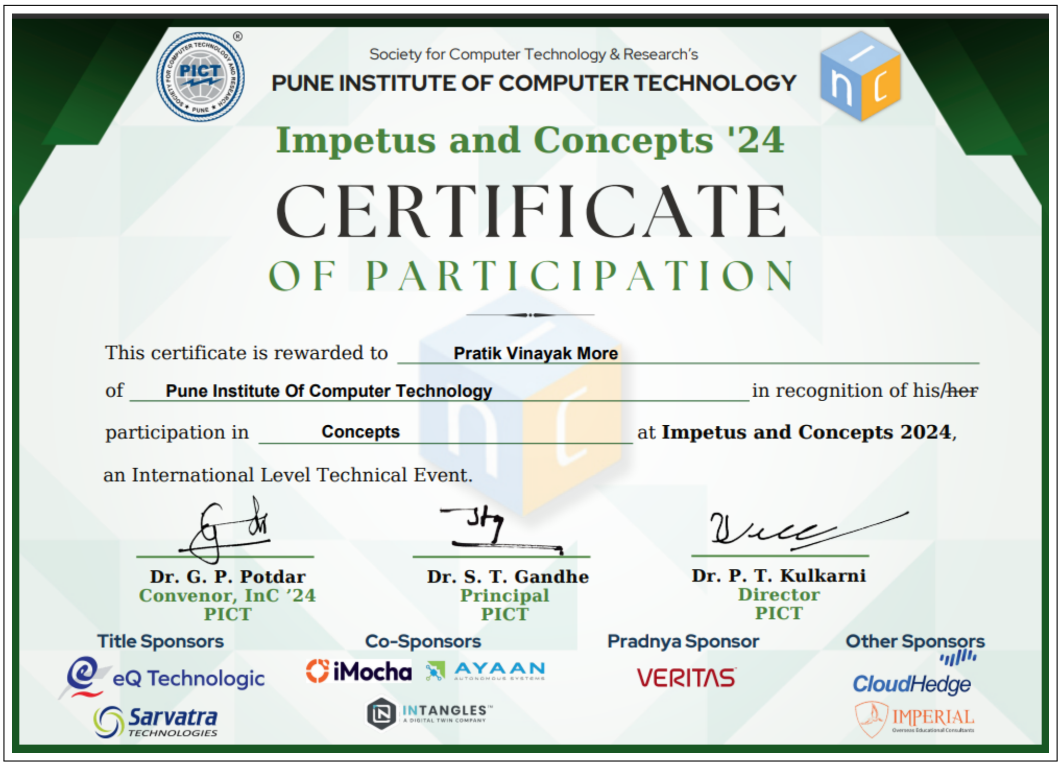

Figure 10.1: Certificate-Pratik

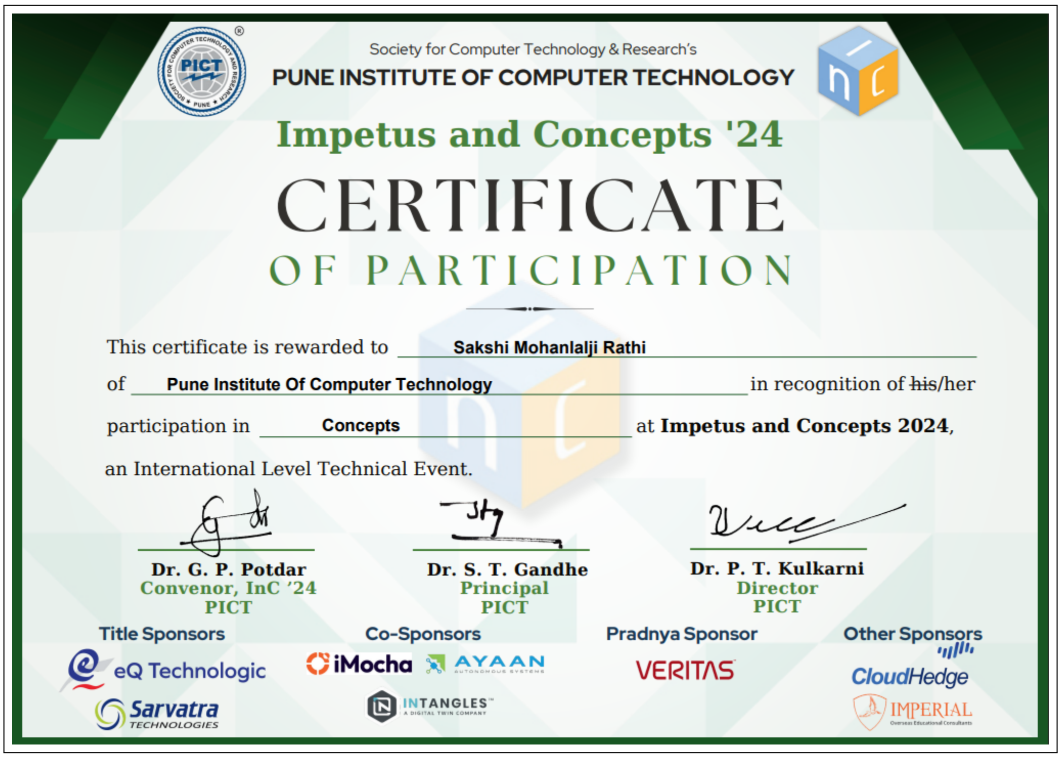

Figure 10.2: fig:Certificate-Sakshi

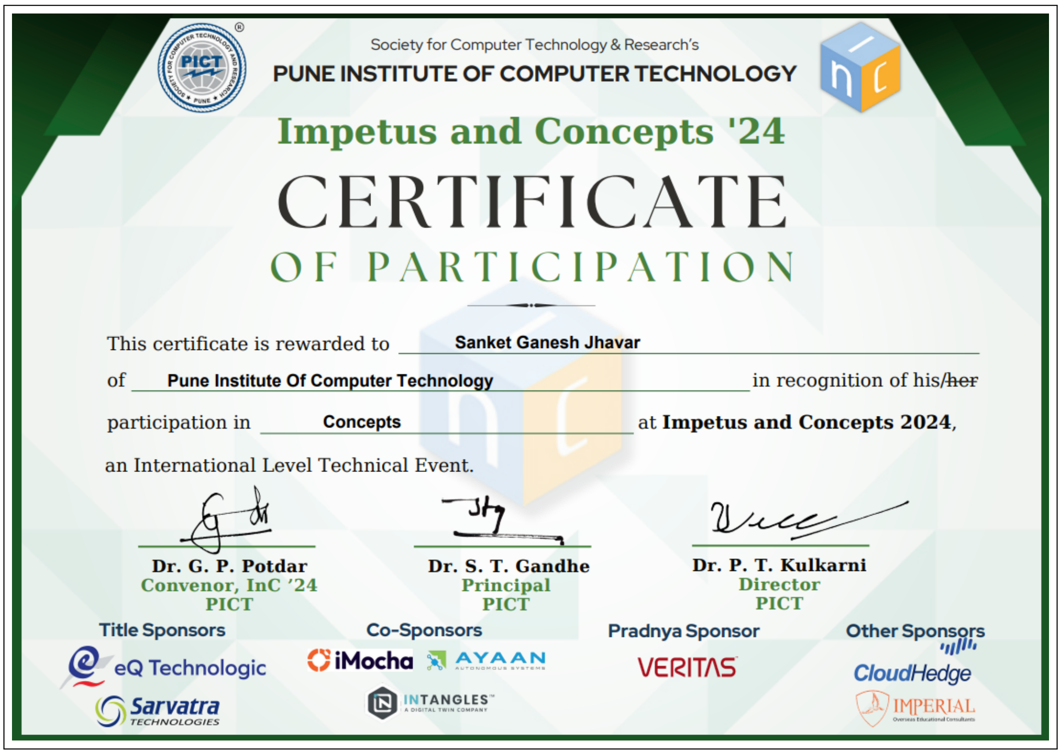

Figure 10.3: Certificate-Sanket

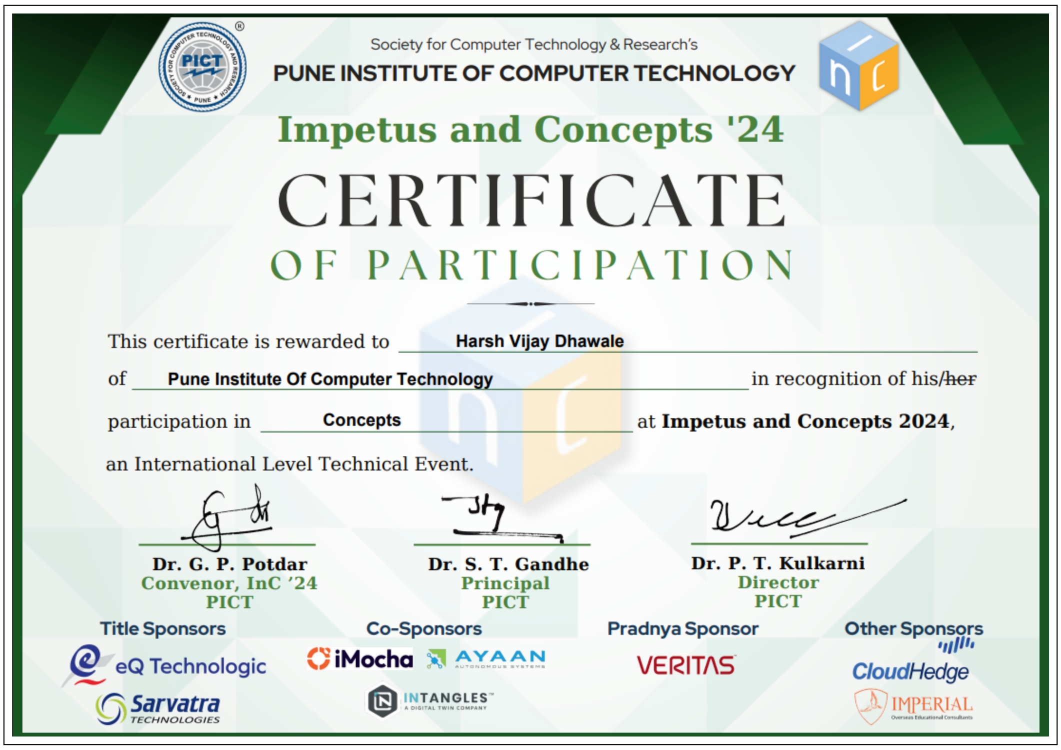

Figure 10.4: Certificate-Harsh

10.4 Appendix C

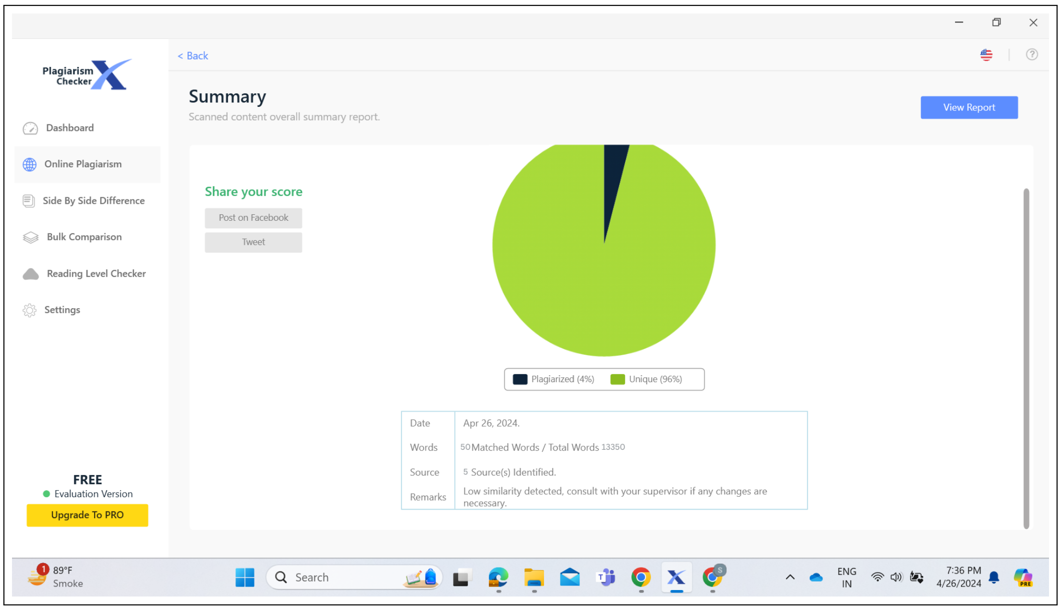

Figure 10.5: Plagiarism Report of project report.

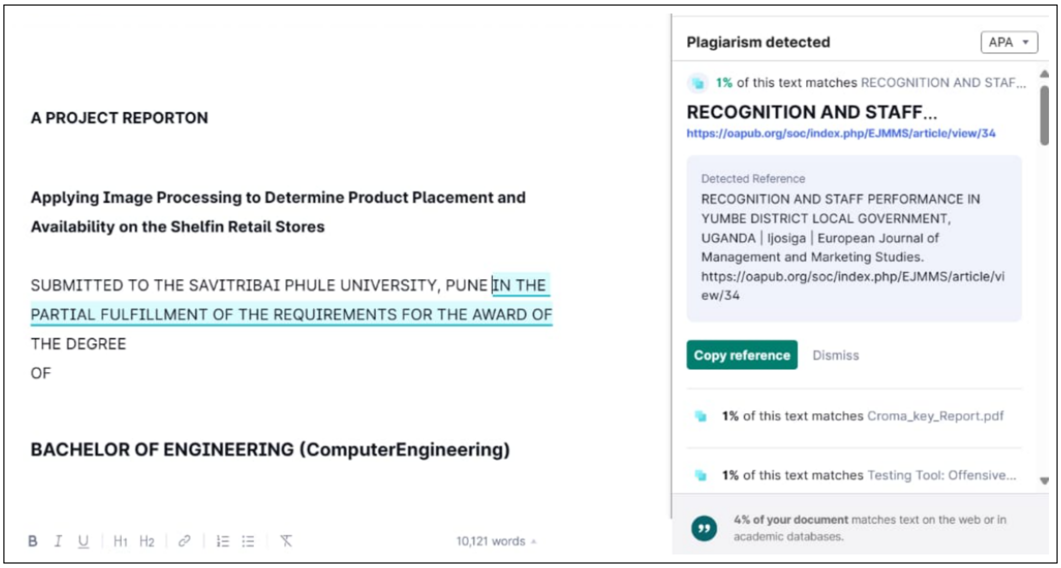

Figure 10.6: Grammerly-Pro Report for Plagarism

CHAPTER 11
REFERENCES

[1] Chu, C., Niu, J., Zheng, W., Su, J., Wen, G. (2021). A time-efficient protocol for unknown tag identification in large-scale RFID systems. IEEE Internet of Things Journal.

[2] Rambe, A., et al. (2020). Analysis SURF feature extraction and SVM classification for facial image recognition from various angles. IOP Conference Series: Materials Science and Engineering.

[3] C. C. (2023). Design and application of a UHF RFID tag for monitoring pollution in high voltage towers. Electric Power Systems Research.

[4] C. C. (2021). Out-of-stock justifications and consumers' behavioral outcomes: explores the role of product type and sales level information in out-of-stock situations. Journal of Retailing and Consumer Services.

[5] Submitted on 21 Feb 2024 (v1), last revised 29 Feb 2024 (this version, v2)] YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information Chien-Yao Wang, I-Hau Yeh, Hong-Yuan Mark Liao

[6] YOLOv9 for Fracture Detection in Pediatric Wrist Trauma X-ray Images 17 Mar 2024 · Chun-Tse Chien, Rui-Yang Ju, Kuang-Yi Chou, Jen-Shiun Chiang ·

[7] Kala, S. (2021, October). Void Dataset [Open Source Dataset]. Roboflow Universe. Retrieved from https://universe.roboflow.com/shivam-kala/void-61xpg

[8] Abhinaya, A. (2021, November). Using machine learning to detect voids in an underground pipe using in-pipe ground penetrating radar (Master's thesis)

Table 11.1: LIST OF ABBREVATIONS

Abbreviation	Illustration
OOS	Out-of-Stock
YOLO	You Only Look Once
POI	Point of interest
ROI	Return on Investment
RFID	Radio-Frequency Identification

LIST OF FIGURES

LIST OF TABLES