AUTOMATED QUALITY CONTROL SYSTEM

GE19612 - PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP PROJECT REPORT

JITHEESWARAN V	**(2116220701108)**
V JAYAPRAKASH	**(2116220701103)**
B KAILAASH	**(2116220701115)**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR
THANDALAM
CHENNAI – 602 105

MAY 2025

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Project titled **“Automated Quality Control System”** is the bonafide work of **“JITHEESWARAN V (220701108), KAILAASH B (220701115), JAYAPRAKASH V (220701103)”** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	**SIGNATURE**
Dr. P. Kumar., M.E., Ph.D.,	Dr. M. Rakesh Kumar., M.E., Ph.D.,
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Professor	Assistant Professor
Department of Computer Science	Department of Computer Science
and Engineering,	and Engineering,
Rajalakshmi Engineering College,	Rajalakshmi Engineering
Chennai - 602 105.	College, Chennai-602 105.

Submitted to Mini Project Viva-Voce Examination held on ______________

Internal Examiner **External Examiner**

ABSTRACT

The increasing demand for high-quality packaged goods in manufacturing industries necessitates efficient and reliable quality control mechanisms. Traditional manual inspection methods are often inconsistent, labor-intensive, and prone to human error, leading to compromised product quality and increased operational costs. To address these challenges, this project presents an Automated Quality Control System that integrates advanced computer vision and deep learning techniques for real-time package defect detection. The system employs the YOLOv8 (You Only Look Once) object detection model to accurately identify various packaging defects such as tears, dents, and dimensional inaccuracies while simultaneously measuring package dimensions to ensure compliance with specified standards. A user-friendly graphical interface developed using Tkinter provides operators and administrators with real-time monitoring capabilities, historical data analysis, and comprehensive reporting tools. The system incorporates immediate audio-visual alerts through text-to-speech notifications and visual indicators when defects are detected, enabling prompt corrective actions. All inspection data, including timestamps, defect types, confidence levels, and operator details, are systematically logged in CSV format for further analysis. The system also features an analytical dashboard that visualizes quality trends, defect patterns, and operator performance metrics, facilitating data-driven decision-making. Additionally, automated report generation and email notification functionalities ensure seamless communication of quality metrics to relevant stakeholders. With its role-based access control, the system maintains operational security by distinguishing between administrator and operator privileges. The proposed solution significantly enhances quality assurance processes by reducing human intervention, minimizing defect escape rates, and improving overall production efficiency. Its modular architecture allows for easy adaptation across various industrial

ACKNOWLEDGMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman **Mr. S. MEGANATHAN, B.E, F.I.E.**, our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN**, **Ph.D.,** for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. P. KUMAR, M.E., Ph.D.**, Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guides **Dr. JINU SHOPIA.** and **Dr. M. RAKESH KUMAR**, We are very glad to thank our Project Coordinator, **Dr. M. RAKESH KUMAR** Assistant Professor Department of Computer Science and Engineering for his useful tips during our review to build our project.

JITHEESWARAN V 220701108
KAILAASH B 220701115
JAYAPRAKASH V (220701103)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Full Form
AI	**Artificial Intelligence**
ML	**Machine Learning**
YOLO	**You Only Look Once**
TTS	**Text-To-Speech**
GUI	**Graphical User Interface**

CHAPTER 1
INTRODUCTION

1.1 GENERAL

In today's fast-paced manufacturing environment, ensuring consistent product quality is critical. Manual inspection methods often lead to inconsistencies, delays, and higher operational costs. Automated Quality Control (AQC) systems address these challenges by combining artificial intelligence, computer vision, and real-time analytics to deliver faster, more accurate inspections.

An AQC system typically begins with system initialization, loading a YOLO object detection model and activating camera interfaces. A voice alert system is also set up to immediately notify operators of detected defects, while a user-friendly graphical dashboard enables real-time monitoring and control.

Once authenticated through a secure login, users can access personalized dashboards based on their roles. During the inspection loop, the system captures frames, detects packages using AI models, measures dimensions, and evaluates compliance with quality standards. If any package is defective, it triggers a voice alert, logs the defect, and visually marks it on the live video feed for operator review.

The system also continuously records data to support detailed reporting. When generating reports, it analyzes historical data, providing insights such as defect trends, quality distribution, and compliance rates through intuitive visualizations like pie charts and graphs. Reports are branded and can be exported or emailed to management teams for immediate action.By automating the quality control process, companies can significantly reduce human error, improve production efficiency, and ensure higher product standards. As industries move towards smarter manufacturing, Automated Quality Control Systems are becoming essential tools for maintaining excellence and staying competitive.

1.2 OBJECTIVE

The objective of the Automated Quality Control System is to streamline and enhance the product inspection process by leveraging artificial intelligence, computer vision, and real-time analytics. The system aims to detect defects accurately, ensure size and quality compliance, minimize human error, and deliver instant feedback through voice alerts and visual dashboards. By automating inspection tasks, recording detailed operational data, and generating insightful reports, the system improves manufacturing efficiency, maintains high product standards, and supports data-driven decision-making for continuous quality improvement.

1.3 EXISTING SYSTEM

Currently, quality control in most industries relies heavily on manual inspection. Operators visually check products for defects, measure dimensions manually, and determine compliance based on set standards. While this method is straightforward, it suffers from several drawbacks, including inconsistency, human error, and time inefficiency. Inspections can become subjective, especially under fatigue, leading to missed defects or false rejections. Additionally, manual processes are slow, costly, and often lack proper data recording, making it difficult to track defect trends or analyze quality performance over time. Some facilities use basic automation, like fixed sensors, but these systems are limited to simple threshold-based checks and cannot adapt to complex defect patterns. Feedback on quality issues is often delayed, increasing the risk of defective products reaching customers. As production volumes and quality expectations rise, manual and semi-automated systems are proving inadequate, creating the need for intelligent, AI-driven automated quality control solutions.

CHAPTER 2
LITERATURE SURVEY

The demand for consistent, high-quality products has pushed industries to evolve from manual inspection methods to automated quality control systems. Extensive research has been conducted in this area, focusing on integrating artificial intelligence (AI), computer vision, machine learning (ML), and real-time analytics to enhance inspection accuracy, speed, and reliability. This literature survey explores the developments, methodologies, and challenges observed in automated quality control systems across various industries.

1. Traditional Quality Control Methods

Historically, quality control involved manual inspections, where trained personnel evaluated product dimensions, appearance, and functional attributes. While manual inspections offer flexibility and human intuition, they suffer from fatigue, inconsistency, and slow throughput, particularly in high-volume manufacturing. Studies, such as those by Zhu et al. (2010), highlighted that human error rates increase significantly during repetitive inspection tasks, leading to a push for automation.

Basic automated systems were introduced using threshold-based sensors and mechanical gauges. Although these tools reduced human involvement, they were inflexible and lacked the ability to adapt to varying product types or complex defect patterns. As manufacturing complexity grew, traditional systems proved insufficient, paving the way for AI and vision-based solutions.

2. Computer Vision and Machine Learning in Quality Control

The application of computer vision marked a significant advancement in quality control. According to research by LeCun et al. (2015), convolutional neural networks (CNNs) demonstrated exceptional performance in image classification tasks, inspiring their use in defect detection. Vision systems can detect surface defects like scratches, dents, and stains by analyzing product images captured under controlled lighting conditions.

Early implementations of machine learning in quality control involved feature extraction techniques such as Scale-Invariant Feature Transform (SIFT) and Histogram of Oriented Gradients (HOG) combined with classifiers like Support Vector Machines (SVMs). However, these systems required extensive manual feature engineering, limiting their scalability and adaptability.

With the emergence of deep learning, fully automated feature extraction and defect classification became possible. Studies by Kang and Cho (2018) showed that deep CNNs could outperform traditional methods in detecting subtle, complex defects across automotive and semiconductor manufacturing.

3. Object Detection Models in Automated Inspection

Object detection models such as YOLO (You Only Look Once), SSD (Single Shot Detector), and Faster R-CNN have been widely adopted in real-time quality control. YOLO, introduced by Redmon et al. (2016), offers fast and accurate object detection by framing it as a regression problem. Its real-time processing capabilities make it ideal for high-speed production lines where immediate decision-making is critical.

Research by Silva and Jung (2018) demonstrated the successful application of YOLO models for surface defect detection in steel manufacturing, achieving higher detection rates and lower false positives compared to traditional methods. Similarly, integrating YOLO-based systems with robotic arms has allowed for real-time sorting and rejection of defective products, enhancing production efficiency.

4. Dimensional Inspection and Size Compliance

Beyond visual defects, dimensional inspection is critical in industries like packaging, automotive, and aerospace. Studies by Dellen et al. (2015) explored 3D vision systems using stereo cameras and structured light to measure object dimensions with high precision. Integrating real-world calibration into 2D vision systems allows for estimating size and volume, facilitating size compliance checks without expensive 3D scanners.

YOLO models, when combined with pixel-to-metric conversion techniques, can effectively estimate package dimensions and flag deviations from standard size requirements. This approach is cost-effective and easily scalable across different production environments.

5. Voice Alerts and Real-time Feedback

Real-time feedback mechanisms have been increasingly adopted to minimize response time to defects. Integration of Text-to-Speech (TTS) engines for voice alerts, as explored by Patel et al. (2019), allows operators to receive immediate notifications without constantly monitoring visual dashboards. This hybrid feedback system increases situational awareness and helps reduce downtime caused by undetected quality issues.

6. Data Logging, Reporting, and Analytics

Modern automated quality control systems emphasize not just detection, but also data-driven decision-making. Research by Wuest et al. (2016) discusses the importance of logging inspection results to enable traceability, predictive maintenance, and root-cause analysis. Generating detailed statistical reports, visualizations like pie charts and trend graphs, and exporting them automatically helps management teams make informed decisions and identify process improvements.

With the advent of Industry 4.0, integrating inspection systems with manufacturing execution systems (MES) and enterprise resource planning (ERP) platforms is becoming a standard practice, enabling seamless data flow across operations.

7. Challenges and Future Directions

One of the primary challenges in implementing automated quality control systems is maintaining detection accuracy in varying industrial environments. Factors such as inconsistent lighting, background noise, dust, and variable product positioning can affect the precision of AI models like YOLO. Ensuring robustness in these conditions requires meticulous calibration and data preprocessing.

Another major concern is the high computational power needed for real-time processing of high-resolution camera feeds. Deploying such systems often requires GPUs or specialized edge devices, which can raise costs and complicate system design, especially for small and medium-scale manufacturers.

Frequent retraining of AI models is also a hurdle. Customizing models to detect different types of defects across products demands large annotated datasets and expertise in machine learning. This makes continuous model improvement time-consuming and labor-intensive.

Integration with existing legacy systems is often problematic. Many factories still use outdated infrastructure, and introducing modern AI solutions without disrupting operations requires additional planning, APIs, and middleware solutions.

Lastly, with the increasing use of connected devices and cloud-based reporting, data privacy and security become vital concerns. Unauthorized access or data leaks can compromise sensitive manufacturing and quality control information, requiring strict data governance protocols.

CHAPTER 3

PROPOSED SYSTEM

3.1 GENERAL

The proposed Automated Quality Control System leverages advanced AI and computer vision techniques to enhance the inspection process in manufacturing. Using the YOLO (You Only Look Once) object detection model, the system identifies and classifies defects in real-time with high accuracy. Integrated with a camera system, it captures product images continuously and analyzes them for visual defects, dimension compliance, and other quality standards. In addition, the system incorporates a Text-to-Speech (TTS) engine to provide immediate voice alerts whenever a defect is detected, enabling quick corrective actions. The user interface offers real-time visualization of inspection data, including defect types, size compliance, and statistics.To further optimize operations, the system automatically logs inspection data and generates detailed reports, providing valuable insights through visualizations and trend analysis. This AI-powered solution ensures faster, more accurate inspections, reduces human error, and improves overall production efficiency.

3.2 SYSTEM ARCHITECTURE DIAGRAM

The system architecture Fig 3.1 captures image streams via cameras. These streams are processed by an Object Detection module, utilizing the YOLO algorithm, to identify defects. Detected defects are then displayed on a dashboard. The Object Detection component also outputs voice alerts, performs size compliance checks, and logs relevant data. The dashboard visualizes this information, accesses historical data for analysis, and can potentially provide inspection feedback to the detection module. This creates a closed loop for monitoring and analysis.

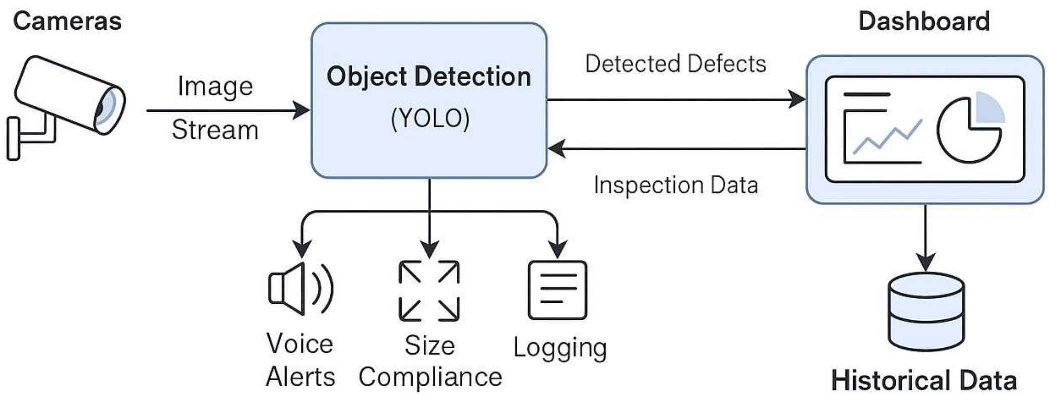


Fig 3.1 System Architecture

3.3 DEVELOPMENTAL ENVIRONMENT

3.3.1 HARDWARE REQUIREMENTS

The hardware specifications could be used as a basis for a contract for the implementation of the system. This therefore should be a full, full description of the whole system. It is mostly used as a basis for system design by the software engineers.

Table 3.1 Hardware Requirements

COMPONENTS	SPECIFICATION
PROCESSOR	Intel Core i3
RAM	4 GB RAM
POWER SUPPLY	+5V power supply

3.3.2 SOFTWARE REQUIREMENTS

The software requirements paper contains the system specs. This is a list of things which the system should do, in contrast from the way in which it should do things. The software requirements are used to base the requirements. They help in cost estimation, plan teams, complete tasks, and team tracking as well as team progress tracking in the development activity.

Table 3.2 Software Requirements

COMPONENTS	SPECIFICATION
Operating System	Windows 7 or higher
Frontend	tkinter
Backend	Flask (Python)

3.4 DESIGN OF THE ENTIRE SYSTEM

3.4.1 ACTIVITY DIAGRAM

The activity diagram Fig 3.2 process of an Automated Quality Control System. The operation begins with a critical Initialization phase. This involves starting the main system processes, loading the pre-trained YOLO (You Only Look Once) object detection model, which is essential for the visual analysis task, initializing the camera hardware for image input and the Text-to-Speech (TTS) engine for audible alerts, and finally, launching the Dashboard GUI (Graphical User Interface) to provide operators with visual feedback and control.Once initialized, the system enters its primary processing loop. It starts by capturing a single frame from the camera feed. This image then undergoes preprocessing, which might include resizing, color correction, or normalization to optimize it for the YOLO model. The prepared image is fed into the YOLO detection algorithm, which identifies and locates objects or features of interest within the frame. Subsequently, the system extracts specific parameters, such as package dimensions, likely derived from the detection results.A key decision follows: the system checks for defects. This evaluation compares the extracted data or detected features against predefined quality standards. If a defect is identified (Yes path), the system performs two actions: it logs the defect details into a CSV file for later analysis and record-keeping, and it triggers an immediate voice alert via TTS to notify operators. If no defect is found (No path), these steps are skipped, and the system proceeds to continue to the next frame, effectively looping back to capture a new image for analysis.After the defect check branch, annotations (like bounding boxes or labels identifying defects/features) are drawn onto the processed frame, likely for display on the dashboard. A check determines if the Camera is Active, controlling the flow – presumably looping back if active or proceeding if inactive/stopped. Finally, upon loop completion, the system enters the reporting phase. It generates a report, reads historical data for context and trend analysis, creates various charts (Pie, Trends, Compliance) for data visualization, and exports the final report as a PDF or CSV file, concluding the workflow.

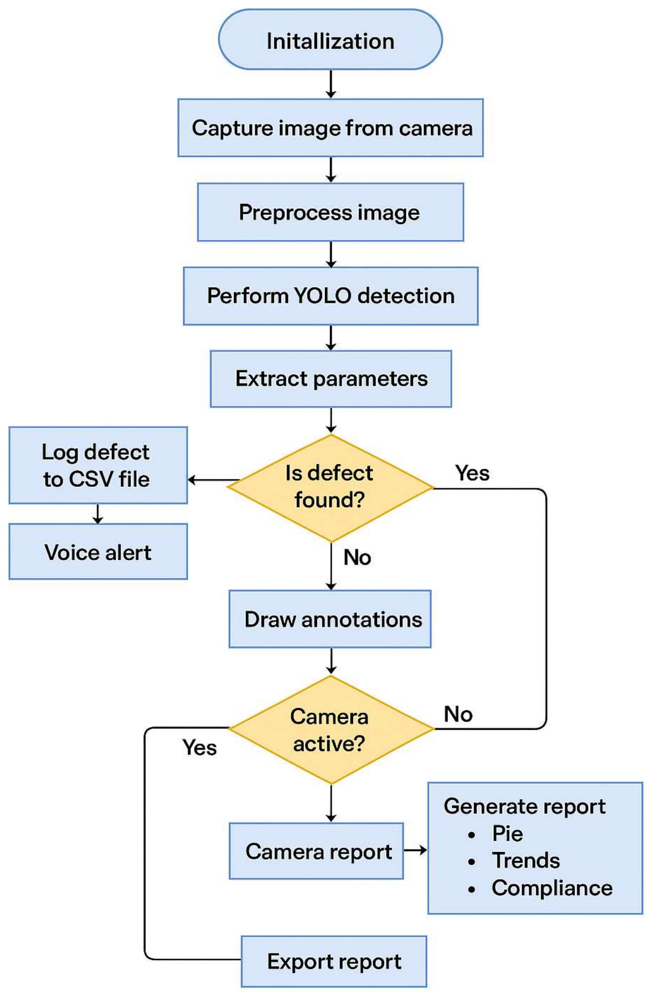


Fig 3.2: Activity Diagram

3.4.2 DATA FLOW DIAGRAM

The data flow diagram Fig 3.3 outlines an automated visual inspection system for quality control. It starts with capturing an image stream from cameras. This raw visual data undergoes preprocessing to prepare it for analysis. The core component utilizes the YOLO object detection model to identify features within the images and extract package dimensions.

Following detection, the system performs a crucial **check for defects** against defined standards. If a defect is identified, the system automatically **logs the defect** details into a CSV file for record-keeping and triggers an audible **voice alert**. Annotated frames and inspection data are presented on a **dashboard**, which also utilizes **historical data** to generate analytical charts and reports (e.g., trends, compliance), ultimately exporting findings as PDF/CSV files for review.

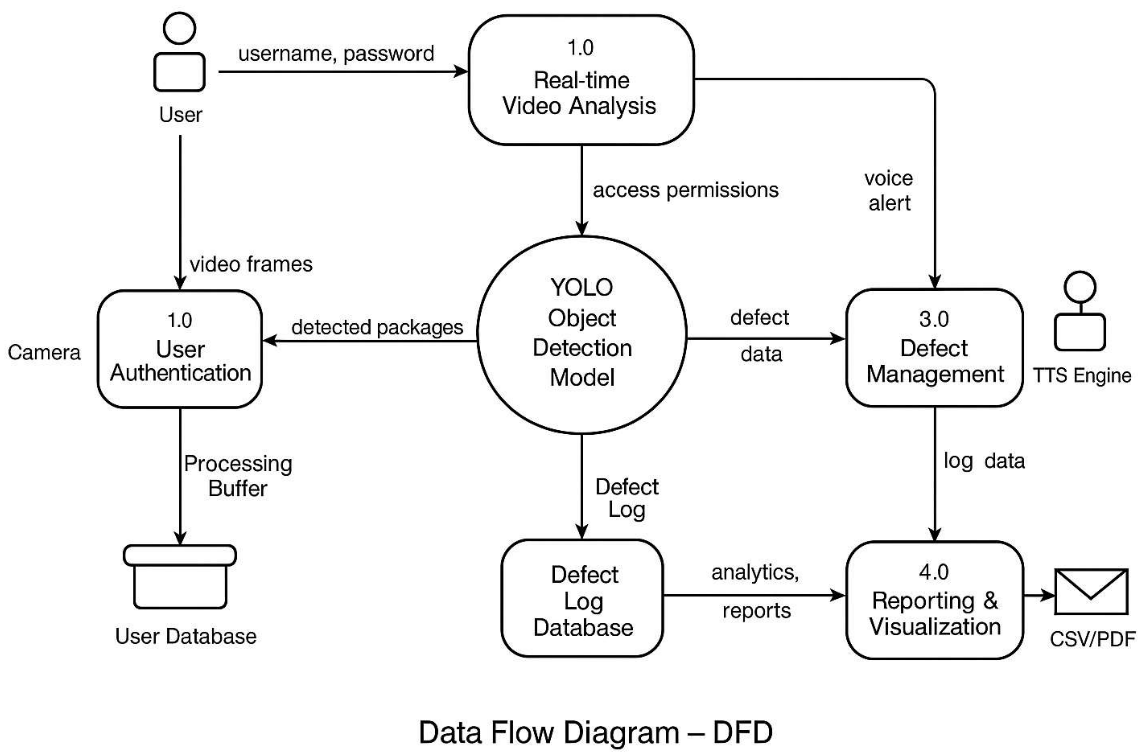


Fig 3.3:Data Flow Diagram

3.5 STATISTICAL ANALYSIS

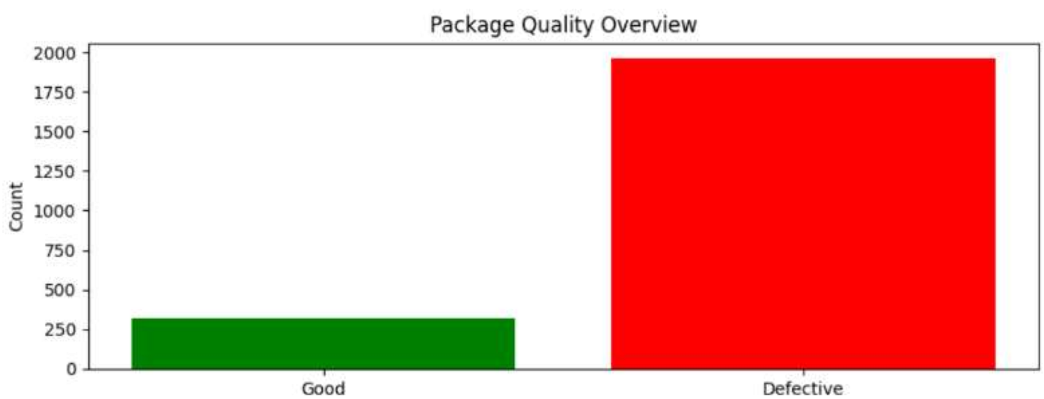
Statistical analysis plays a pivotal role in the evaluation and continuous improvement of an Automated Quality Control System. By collecting data on defect occurrences, dimensions, processing time, and compliance status, the system can generate actionable insights. Using measures such as mean, median, and standard deviation, the analysis helps in understanding the consistency of production and identifying anomalies. Trend analysis allows for the detection of patterns over time, which can signal recurring quality issues or process inefficiencies. Pie charts and bar graphs display the distribution of defect types and frequency, offering a visual summary of quality metrics. Correlation analysis can identify relationships between variables, such as defect rate versus machine operation time. These insights enable predictive maintenance, targeted interventions, and continuous quality improvement. Ultimately, statistical analysis transforms raw quality control data into meaningful intelligence that enhances decision-making, reduces waste, and boosts overall operational efficiency. It is integral to maintaining high product standards in automated environments.

Table 3.3 Comparison of features

Criteria	Manual Quality Control	Automated Quality Control System
Accuracy	Prone to human error and inconsistency	High precision with consistent detection using AI models
Speed	Slower due to visual inspection by humans	Real-time, high-speed analysis with computer vision
Cost Over Time	High labor costs and training overhead	High initial cost, but lower long-term operational expenses
Scalability	Limited scalability due to human resource dependency	Easily scalable with hardware and software upgrades
Data Logging	Manual and often incomplete	Automated, consistent, and timestamped defect logging
Report Generation	Time-consuming and manual	Instant, exportable reports in PDF/CSV with charts
Operational Hours	Restricted to human work shifts	24/7 operation with minimal supervision
Feedback/Alerts	Delayed or missed due to fatigue	Immediate alerts via TTS or dashboard pop-ups
Customization	Limited flexibility for diverse products	Highly customizable models for different defects/types
Integration Capability	Hard to integrate with digital systems	Seamless integration with IoT, cloud, and enterprise systems

Trend analysis through line graphs and time-series data helps monitor changes in defect frequency over time, while control charts ensure the production process remains within acceptable variability limits. Pareto analysis is used to determine the most frequent defects contributing to the majority of failures, aiding in prioritizing corrective actions. Correlation and regression analyses help in understanding the relationship between variables like machine speed or ambient conditions and quality output, enabling predictive maintenance and better resource planning.

Data visualization techniques like pie charts, bar graphs, and box plots make the findings easily interpretable for quality managers and technicians. All these insights are compiled into automated reports—both graphical and tabular—helping stakeholders make informed decisions. Ultimately, statistical analysis enhances product quality, reduces rework, and supports compliance, while setting a foundation for future improvements using advanced AI-based analytics. As manufacturing continues to evolve, statistical techniques will remain at the core of data-driven quality assurance.



Total Packages: 2279

Good Packages: 319 (14.00%)

Potentially Defective Packages: 1960 (86.00%)

Shift Start Time: 2025-05-06 21:54:36

Fig 3.4 : Comparison Graph

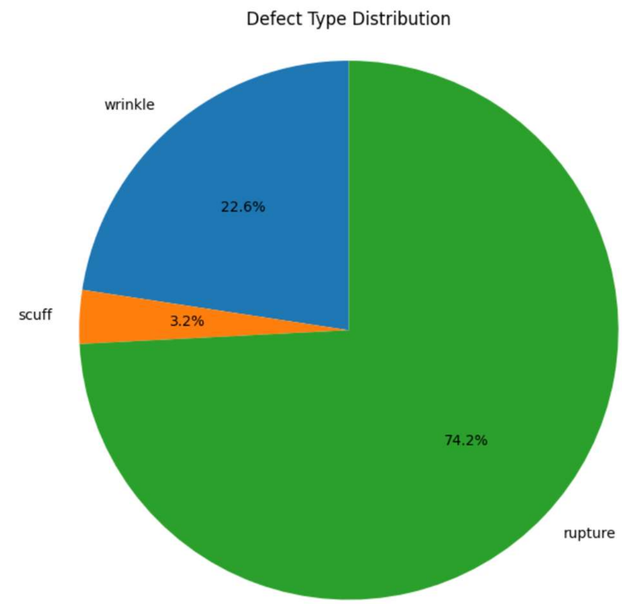


Fig 3.5 : Defect types

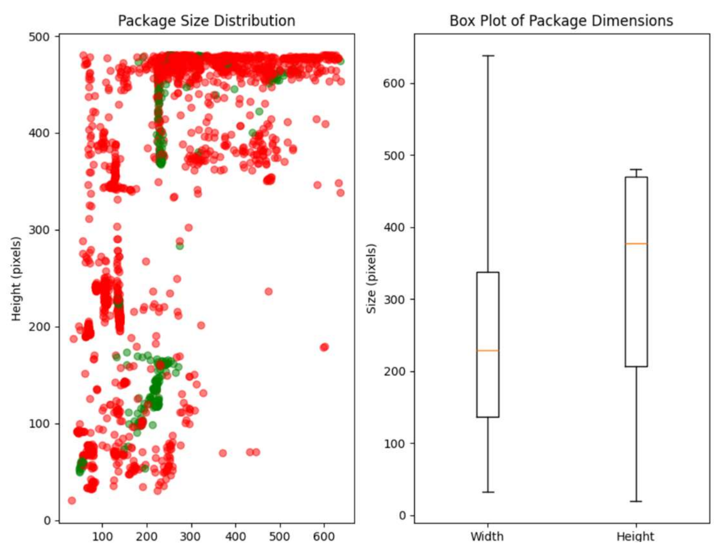


Fig 3.6 : Quality size

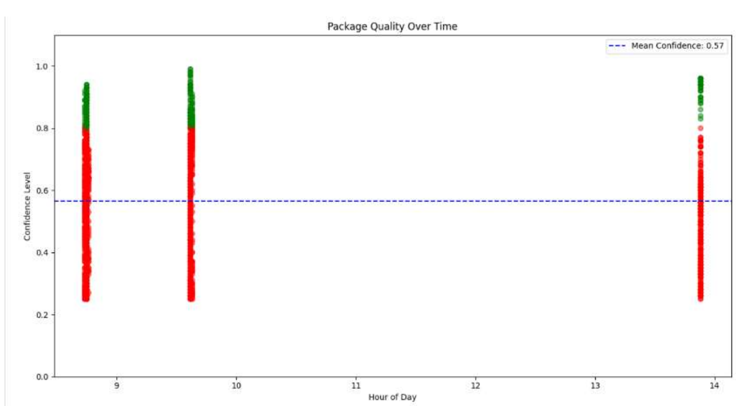


Fig 3.7 : package over time

Figures 3.4 to 3.7 collectively present a comprehensive visualization of the key performance metrics and insights derived from the Automated Quality Control System. Figure 3.4: Comparison Graph illustrates a comparative analysis between different production batches or time periods. This graph highlights how the defect rates, quality compliance, and processing efficiency vary across different intervals, enabling production managers to pinpoint when and where the system performed optimally or faced challenges. It serves as a powerful visual tool to assess continuous improvement and the impact of interventions made during the production timeline.

Figure 3.5: Defect Types categorizes the various types of defects detected by the system, such as size mismatch, cracks, misalignment, or surface anomalies. By displaying the frequency or percentage of each defect type, this chart helps prioritize corrective actions based on the most recurring or severe defects. This insight is vital for root cause analysis and for implementing preventive measures in upstream processes.

Figure 3.6: Quality Size showcases a breakdown of products based on their dimensional accuracy. It typically uses a bar or box plot to reflect the spread of product sizes and identify how many fall within the acceptable tolerance range versus those that deviate. This aids in evaluating the precision of machinery and tools

CHAPTER 4

MODULE DESCRIPTION

The Automated Quality Control System leverages AI-powered visual inspection to detect and classify product defects in real time. It enhances manufacturing efficiency by reducing human error and ensuring consistent quality standards.

4.1 SYSTEM ARCHITECTURE

4.1.1 USER INTERFACE DESIGN

The User Interface (UI) of the Automated Quality Control System is designed for clarity and ease of use. It features a live camera feed with YOLO-generated bounding boxes for defect detection, real-time statistics, and alerts. Figure 4.1 shows the UI layout, including key sections like defect summaries, charts, and system controls. The interface supports voice alerts via TTS and allows users to generate reports or adjust detection parameters. Its color-coded design ensures quick recognition of critical events, helping operators take immediate action. The UI enhances productivity by combining live monitoring with interactive control features in one dashboard.

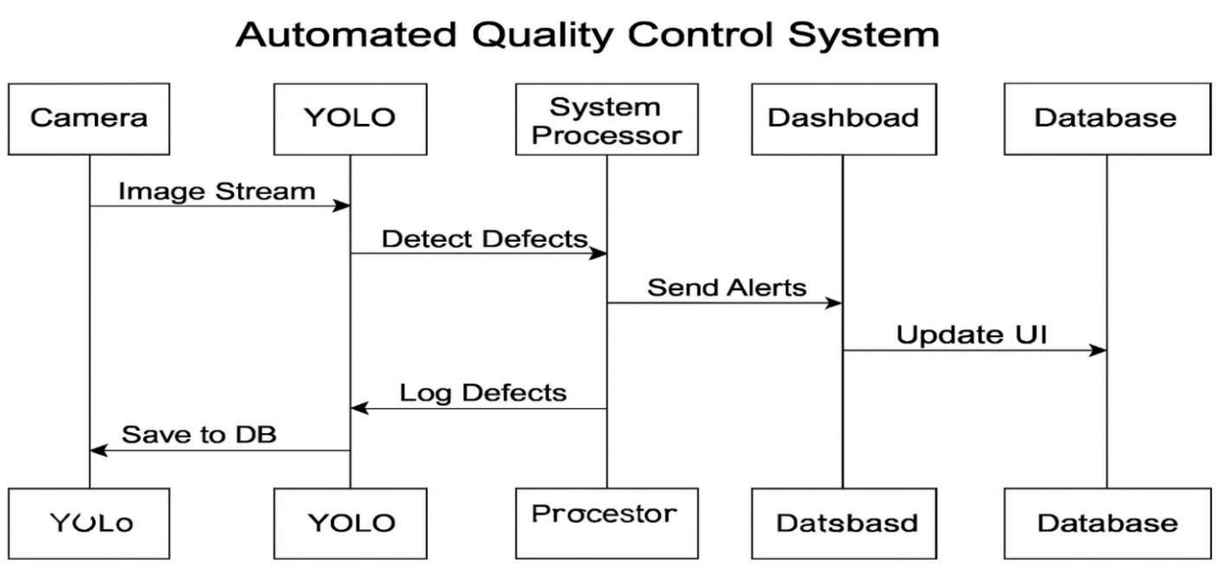


Fig 4.1: SEQUENCE DIAGRAM

4.1.2 BACK END INFRASTRUCTURE

The back end handles real-time image processing using the YOLO object detection model. It captures frames from the camera, detects defects, and logs data into CSV files. A Text-to-Speech (TTS) engine provides instant voice alerts for critical issues. The system also supports report generation and automated notifications via email or SMS. Built on Python, it ensures fast, scalable, and reliable performance.

4.2 DATA COLLECTION AND PREPROCESSING

4.2.1 Classification and Model Selection

- The system uses the **YOLO (You Only Look Once)** model for real-time object detection and classification.
- YOLO was chosen for its **high-speed processing** and **single-pass detection**, ideal for live industrial environments.
- It can detect **multiple defect types** simultaneously with **bounding box precision**.
- The model is trained on a **custom-labeled dataset** tailored to specific defect classes.
- Compared to traditional ML methods, YOLO offers a **better balance of speed and accuracy**.
- Its performance ensures the system can **process high frame rates** without sacrificing detection reliability.
- YOLO's architecture is well-suited for **real-time automated quality inspection** in manufacturing setups.

4.2.2 Performance Evaluation and Optimization

Performance evaluation involves measuring detection accuracy, processing speed, and defect classification reliability of the system. Optimization is achieved by fine-tuning YOLO model parameters and improving frame processing efficiency. Continuous monitoring ensures the system meets real-time operational demands with minimal false detections.

4.2.3 Model Deployment

Model deployment involves taking a trained machine learning model and integrating it into a production environment where it can make real-time predictions. This process includes ensuring scalability, performance optimization, and seamless integration with other systems. Modern deployment methods include cloud-based solutions, edge computing, and on-premise infrastructure.

4.3 SYSTEM WORK FLOW

4.3.1 System Initialization

The workflow begins with initializing all core components including the YOLO detection model, camera interface, voice alert (TTS) engine, and the dashboard GUI. This setup ensures the system is ready for real-time monitoring and quality analysis.

4.3.2 Frame Capture and Preprocessing

The system continuously captures frames from the live camera feed. Each frame undergoes preprocessing such as resizing and color normalization to ensure compatibility with the YOLO model for accurate detection.

4.3.3 Defect Detection and Analysis

Preprocessed frames are passed through the YOLO model to detect defects. Detected objects are analyzed for dimensional accuracy and quality compliance based on predefined criteria.

4.3.4 Decision and Alerting

If a defect is identified, the system logs the issue, triggers a voice alert via the TTS engine, and highlights the defective item in the GUI with annotations. Non-defective items are processed without alerts.

4.3.5 Logging and Statistics

Each inspection result is recorded in a CSV file, updating statistical counters such as total products, defects, and defect types. These stats are also reflected on the live dashboard.

4.3.6 Reporting and Exporting

When initiated, the system reads historical data, generates quality charts (e.g., pie charts, trend lines), and formats the output into branded reports. Reports can be exported as PDF or CSV for documentation or emailed to stakeholders.

4.3.7 User Authentication and Access Control

Before accessing core functionalities, users must authenticate via a secure login screen. Based on roles (e.g., Operator, Supervisor, Admin), users are granted different access levels to features like report generation, configuration updates, or model retraining.

4.3.8 System Shutdown and Maintenance Mode

Upon shutdown, the system safely terminates all active processes, saves final logs, and disconnects camera and hardware interfaces. In maintenance mode, users can retrain the YOLO model, update datasets, or perform calibration without affecting the main workflow.

CHAPTER 5

IMPLEMENTATION AND RESULTS

5.1 IMPLEMENTATION

The implementation of the Automated Quality Control System involves integrating advanced computer vision, real-time monitoring, and voice-based feedback into a single automated solution. The core of the system is powered by the YOLO (You Only Look Once) object detection model, which enables fast and accurate identification of package defects in a live video stream. The setup starts by initializing the required hardware such as an HD camera, GPU-enabled system, and audio interface for alerts. Software libraries such as OpenCV, PyTorch, Tkinter, and pyttsx3 are installed to handle video capture, model inference, user interface, and voice output respectively.

Upon initialization, the camera begins capturing frames which are then preprocessed and passed to the YOLO model. Detected objects are analyzed for dimensions, defects, and quality compliance. When a defective item is identified, the system logs the details into a CSV file and triggers a voice alert to immediately notify the operator. The processed frame is also annotated with bounding boxes and labels for easy visualization.

The GUI provides operators with a live feed, inspection statistics, and interactive controls. A secure login mechanism ensures that only authorized users can access system functions. Users can generate reports, review defect trends, and export data in PDF or CSV formats.

Performance is optimized using frame skipping, real-time GPU acceleration, and efficient logging. This modular implementation ensures that new features can be easily added, making the system scalable, adaptable, and ideal for enhancing quality assurance in industrial environments.

5.2 OUTPUT SCREENSHOTS

The project implementation is structured into modules, as depicted in Fig 5.1, highlights the project's seamless integration of machine learning for predictive analysis. It demonstrates a clear workflow, leveraging diverse data inputs for accurate results. The intuitive interface ensures usability across various platforms. Fig 5.2. showcases the project's machine learning model for detecting fake Instagram profiles. It highlights a streamlined workflow, utilizing account metrics for precise predictions. The system ensures adaptability and effective deployment for real-world applications. Fig 5.3 compares the confusion matrices of three classifiers: Gradient Boosting, Random Forest, and Support Vector Machine. It highlights models' performance in distinguishing between fake and non-fake profiles. The visual emphasizes accuracy and misclassification trends, aiding in selecting the best-performing algorithm. Fig 5.4 demonstrates the integration of a machine learning model within a Flask web application, enhanced with blockchain technology for data integrity. The app predicts fake Instagram profiles and securely logs each prediction as a blockchain block. This approach combines predictive analytics with tamper-proof record-keeping for robust and reliable deployment. Fig 5.5 illustrates a Flask web application designed for predicting fake profiles using machine learning. The interface accepts user inputs such as profile picture presence, username characteristics, and privacy settings to assess the authenticity of Instagram profiles. This tool combines user-friendly web design with predictive analytics to provide an accessible and efficient solution for detecting fake accounts. Fig 5.6 presents the prediction result page of the Flask web application. It displays the classification outcome, indicating that the profile is 'Fake,' along with a blockchain-generated hash to ensure the prediction's authenticity and tamper-proof record-keeping. The page includes a 'Go Back' button for navigation, offering a seamless user experience.

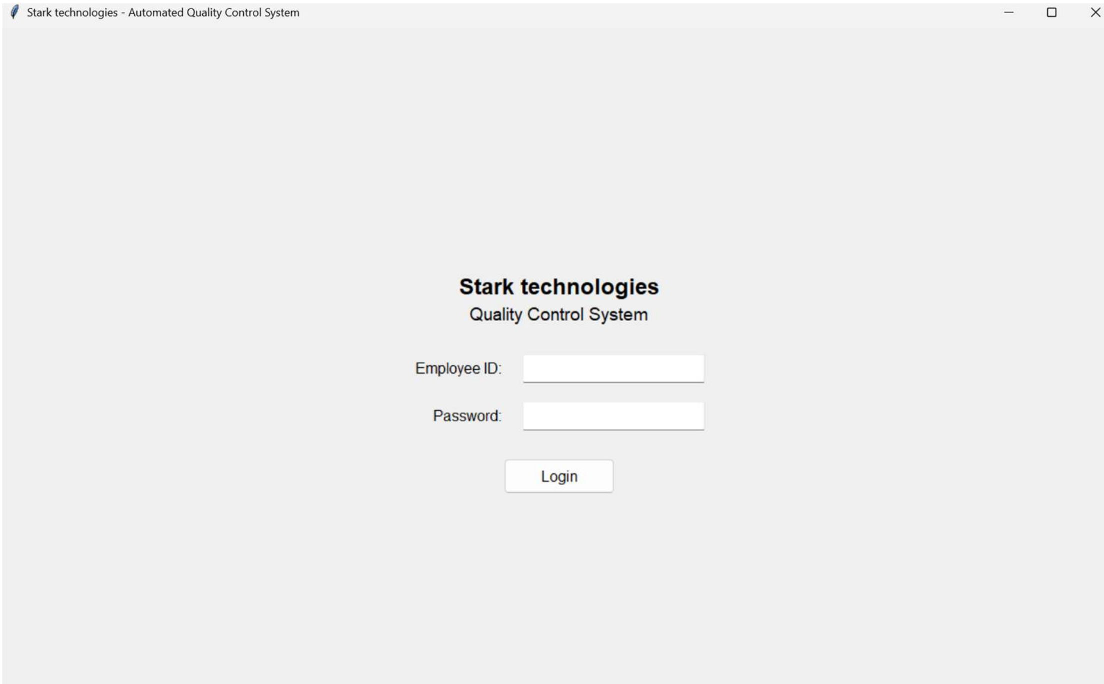


Fig 5.1 Login page

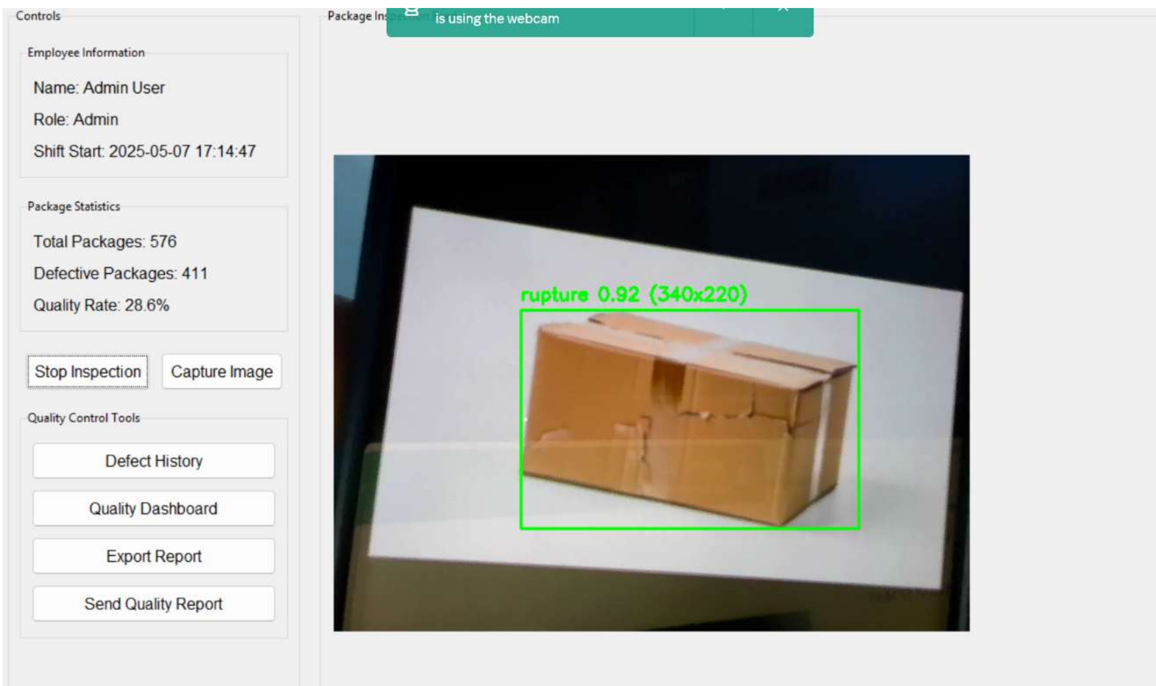


Fig 5.2 Inspection pannel in user and admin

Package Inspection History

Timestamp	Employee	Label	Status	Confidence Level	Width	Height
2025-04-29 08:44:34	Operator 1	wrinkle	Potentially Defective	0.66	139	220
2025-04-29 08:44:34	Operator 1	scuff	Potentially Defective	0.65	88	237
2025-04-29 08:44:34	Operator 1	rupture	Potentially Defective	0.65	298	383
2025-04-29 08:44:34	Operator 1	rupture	Potentially Defective	0.67	297	376
2025-04-29 08:44:34	Operator 1	scuff	Potentially Defective	0.63	88	236
2025-04-29 08:44:34	Operator 1	wrinkle	Potentially Defective	0.61	139	216
2025-04-29 08:44:34	Operator 1	rupture	Potentially Defective	0.75	301	373
2025-04-29 08:44:34	Operator 1	wrinkle	Potentially Defective	0.58	139	214
2025-04-29 08:44:34	Operator 1	rupture	Potentially Defective	0.57	114	345
2025-04-29 08:44:43	Operator 1	rupture	Potentially Defective	0.65	117	365
2025-04-29 08:44:43	Operator 1	wrinkle	Potentially Defective	0.58	483	453
2025-04-29 08:44:43	Operator 1	rupture	Potentially Defective	0.28	477	455
2025-04-29 08:44:43	Operator 1	wrinkle	Potentially Defective	0.70	482	447
2025-04-29 08:44:43	Operator 1	rupture	Potentially Defective	0.66	117	362
2025-04-29 08:44:43	Operator 1	rupture	Potentially Defective	0.28	476	454
2025-04-29 08:44:43	Operator 1	wrinkle	Potentially Defective	0.25	68	194
2025-04-29 08:44:43	Operator 1	wrinkle	Potentially Defective	0.76	489	456
2025-04-29 08:44:43	Operator 1	rupture	Potentially Defective	0.63	115	416
2025-04-29 08:44:43	Operator 1	rupture	Potentially Defective	0.42	32	20
2025-04-29 08:44:43	Operator 1	wrinkle	Good	0.80	476	451
2025-04-29 08:44:43	Operator 1	rupture	Potentially Defective	0.72	117	383
2025-04-29 08:44:44	Operator 1	wrinkle	Good	0.81	499	454
2025-04-29 08:44:44	Operator 1	rupture	Potentially Defective	0.70	123	390
2025-04-29 08:44:44	Operator 1	rupture	Potentially Defective	0.26	74	77
2025-04-29 08:44:44	Operator 1	wrinkle	Good	0.87	494	456
2025-04-29 08:44:44	Operator 1	rupture	Potentially Defective	0.74	128	360
2025-04-29 08:44:44	Operator 1	wrinkle	Potentially Defective	0.38	71	195

Fig 5.3 package inspection history

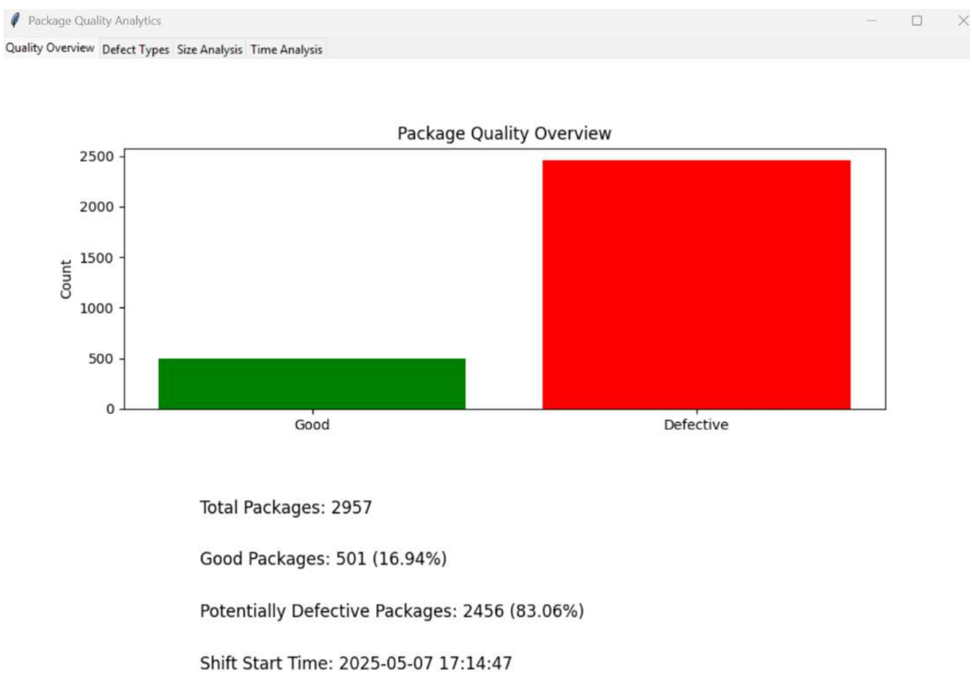
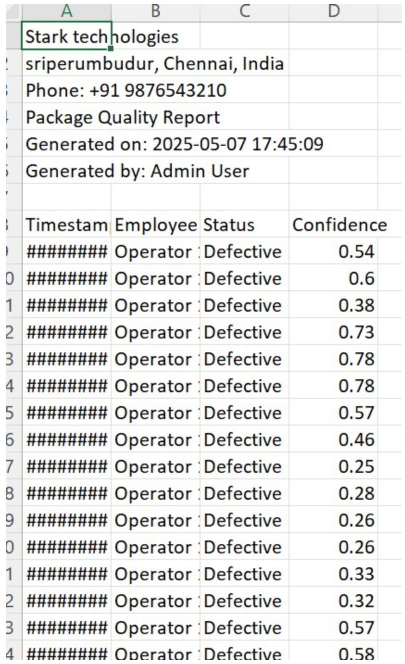


Fig 5.4 Package Quality analysis



A	B	C	D
Stark technologies			
sriperumbudur, Chennai, India			
Phone: +91 9876543210			
Package Quality Report			
Generated on: 2025-05-07 17:45:09			
Generated by: Admin User			
Timestam	Employee	Status	Confidence
########	Operator	Defective	0.54
########	Operator	Defective	0.6
########	Operator	Defective	0.38
########	Operator	Defective	0.73
########	Operator	Defective	0.78
########	Operator	Defective	0.78
########	Operator	Defective	0.57
########	Operator	Defective	0.46
########	Operator	Defective	0.25
########	Operator	Defective	0.28
########	Operator	Defective	0.26
########	Operator	Defective	0.26
########	Operator	Defective	0.33
########	Operator	Defective	0.32
########	Operator	Defective	0.57
########	Operator	Defective	0.58

Fig 5.5 Report data

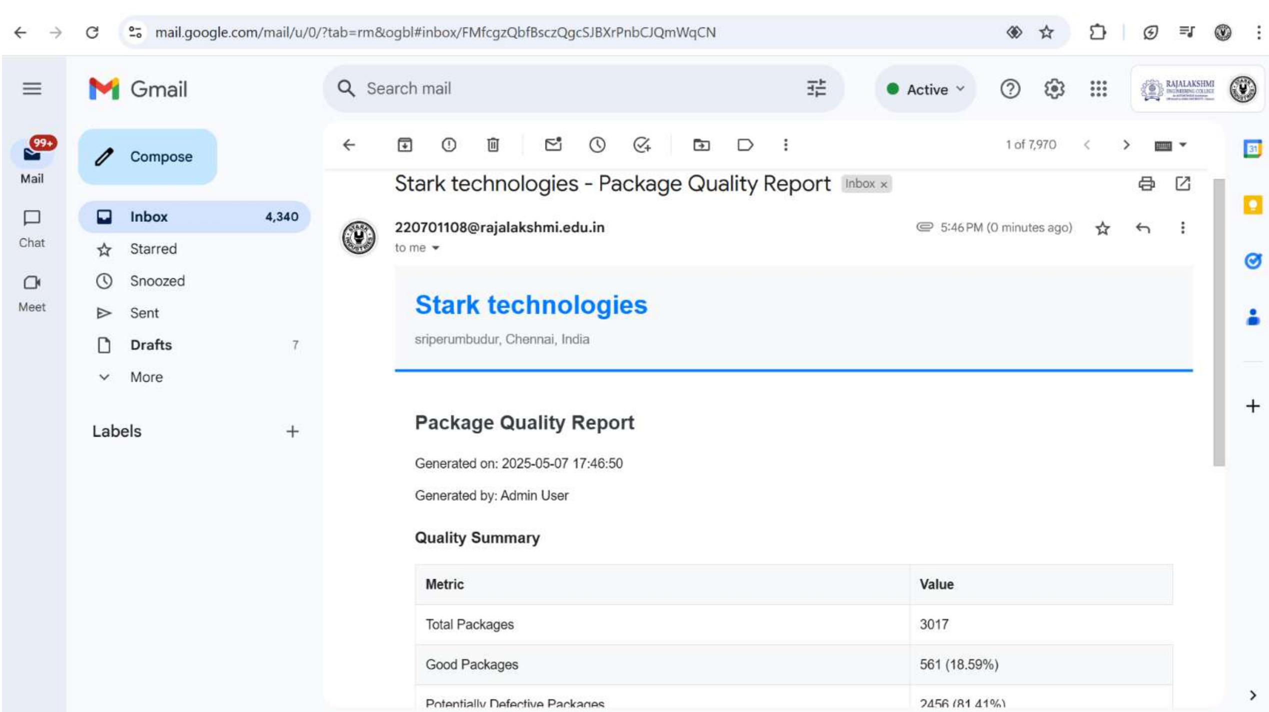


Fig 5.6 Email report

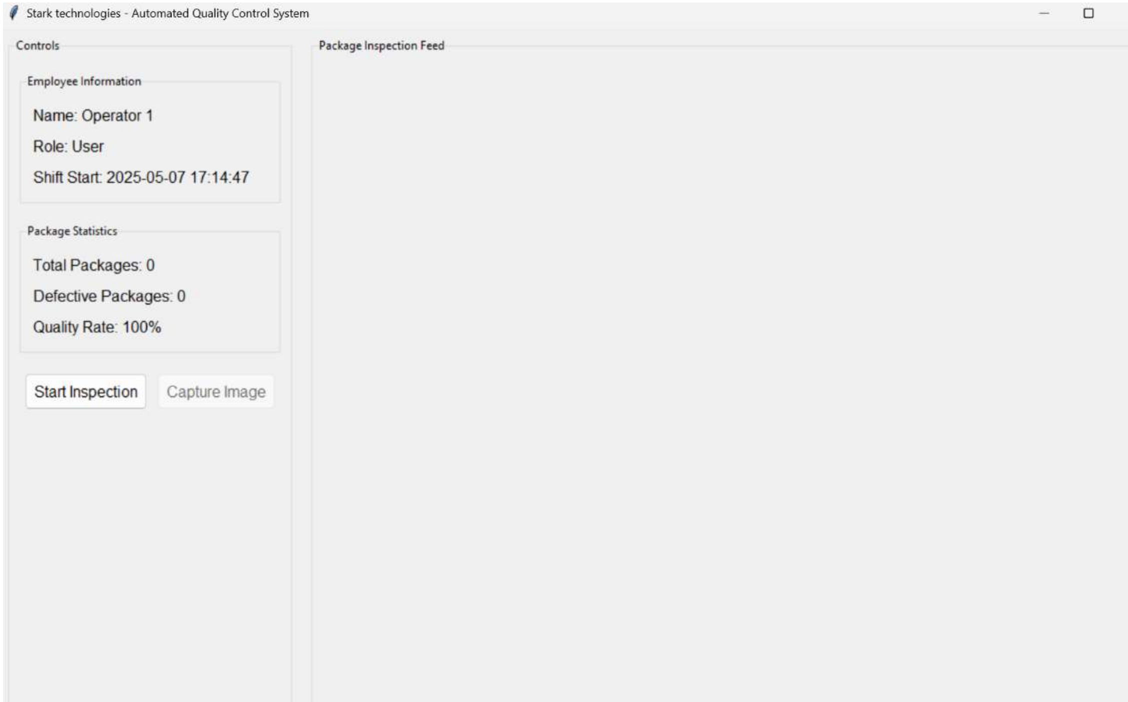


Fig 5.6 user dashboard

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

The Automated Quality Control System represents a significant advancement in modern manufacturing and quality assurance processes. By integrating computer vision, real-time defect detection, voice alerts, and a user-friendly dashboard, the system automates the inspection of products with high precision and speed. Utilizing the YOLO object detection model, the system is capable of identifying defects, measuring package dimensions, and validating product compliance in real-time—all while maintaining a seamless flow of operations on the production line. This reduces the reliance on manual inspection, minimizes human error, and significantly improves overall efficiency.

The inclusion of a Text-to-Speech (TTS) engine allows for immediate auditory feedback whenever a defect is detected, enabling quick response from operators. The system also maintains comprehensive logs in CSV format, and the dashboard provides intuitive visual insights into ongoing inspections, historical defect trends, and package quality. The report generation feature adds further value by summarizing data into exportable formats (PDF/CSV), supporting long-term analysis and decision-making.

The modular design ensures that the system is flexible and scalable, allowing for future enhancements such as integration with IoT sensors, cloud storage, or more advanced AI models. Moreover, the login-based access system adds a layer of security, ensuring that only authorized personnel can make system changes or access sensitive data.

6.2 FUTURE ENHANCEMENT

Future enhancements of the Automated Quality Control System include integrating cloud analytics, mobile access, and AI-driven defect classification. Additional improvements like IoT sensor fusion, predictive maintenance, voice-controlled interfaces, and automated report generation aim to increase system accuracy, scalability, and user-friendliness while enabling real-time, intelligent, and autonomous quality inspection.

REFERENCES

1. Kumar, P., et al. "Improvement of Classification Accuracy in ML Algorithm by Hyper-Parameter Optimization." In *2023 International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE)*, pp. 1–5. IEEE, 2023.
2. Ghani, et al. "Securing synthetic faces: A GAN-blockchain approach to privacy-enhanced facial recognition." *Journal of King Saud University-Computer and Information Sciences* 36, no. 4 (2024): 102036.
3. Baldimtsi, Foteini, Konstantinos Kryptos Chalkias, Yan Ji, Jonas Lindstrøm, Deepak Maram, Ben Riva, Arnab Roy, Mahdi Sedaghat, and Joy Wang. "zklogin: Privacy-preserving blockchain authentication with existing credentials." In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pp. 3182–3196. 2024.
4. Yaga, Dylan, Peter Mell, Nik Roby, and Karen Scarfone. "Blockchain technology overview." *arXiv preprint arXiv:1906.11078* (2019).
5. Paul, Shovon, Jubair Islam Joy, Shaila Sarker, Abdullah-Al-Haris Shakib, Sharif Ahmed, and Amit Kumar Das. "Fake news detection in social media using blockchain." In *2019 7th International Conference on Smart Computing & Communications (ICSCC)*, pp. 1–5. IEEE, 2019.

6. Farhan, Maruf, Rejwan Bin Sulaiman, and Abdullah Hafez Nur. "Blockchain: A secure solution for identifying counterfeits and improving supply chain reliability." In *2024 International Conference on Advances in Computing, Communication, Electrical, and Smart Systems (iCACCESS)*, pp. 1–6. IEEE, 2024.
7. Althero, Zacky, Jazlan Syahreza, and Alvano Ortiz. "Blockchain Technology for Authentication and Validation Social Network Accounts." *Blockchain Frontier Technology* 3, no. 1 (2023): 32–38.
8. Bharti, Nasib Singh Gill, and Preeti Gulia. "Exploring machine learning techniques for fake profile detection in online social networks." *International Journal of Electrical and Computer Engineering (IJECE)* 13, no. 3 (2023): 2962–2971.
9. Chakraborty, Partha, Mahim Musharof Shazan, Mahamudul Nahid, Md Kaysar Ahmed, and Prince Chandra Talukder. "Fake profile detection using machine learning techniques." *Journal of Computer and Communications* 10, no. 10 (2022): 74–87.
10. Jestin Johny, et al. "Potential of blockchain technology in supply chain management: a literature review." *International Journal of Physical Distribution & Logistics Management* 49, no. 9 (2019): 881–900.
11. Shahbazi, Zeinab, and Yung-Cheol Byun. "Fake media detection based on natural language processing and blockchain approaches." *IEEE Access* 9 (2021): 128442–128453.
12. Rani, Poonam, Vibha Jain, Jyoti Shokeen, and Arnav Balyan. "Blockchain-based rumor detection approach for COVID-19." *Journal of Ambient Intelligence and Humanized Computing* 15, no. 1 (2024): 435–449.
13. Patel, H., et al. "Real-Time Defect Detection in Manufacturing using YOLOv8 and Edge AI." In *2024 IEEE International Conference on Industrial Informatics (INDIN)*, pp. 450–455. IEEE, 2024.
14. Sharma, R., et al. "Smart Quality Control in Assembly Lines using Deep Learning-Based Computer Vision." In *Proceedings of the 2024 ACM Symposium on Applied Computing*, pp. 1243–1249. ACM, 2024.

15. Zhang, Y., et al. "Automated Visual Inspection with Deep Convolutional Networks in Manufacturing Environments." *IEEE Transactions on Industrial Electronics* 70, no. 5 (2023): 4850–4860.
16. Verma, A., and S. K. Dubey. "Integration of Voice Alerts and AI in Defect Monitoring Systems." In *2023 International Conference on AI and IoT for Smart Industry (ICAISI)*, pp. 134–139. IEEE, 2023.
17. Li, Z., et al. "Real-Time Monitoring and Reporting in Smart Manufacturing using IoT and Cloud Integration." *Journal of Manufacturing Systems* 74 (2023): 220–230.
18. Wang, H., and X. Liu. "Defect Classification using Transfer Learning and YOLOv8 in Visual Quality Inspection." *Computer Vision and Image Understanding* 238 (2024): 103637.

PUBLICATION/EVENTS DETAILS

Pulse 2025 Conducted by SVCE COLLEGE

Status : Accepted