COLLEGE CODE: 3105

COLLEGE NAME: dhanalakshmi srinivasan college of engineering and technology

DEPARTMENT: computer science and engineering

STUDENT NM-ID: A9C02D326F750DA88195BB524F4FF8C0

ROLL NO : 310523104012

DATE: 07.05.2025

TECHNOLOGY-PROJECT NAME : quality control in manufacturing

SUBMITTED BY,

Archana. S

Phase 5: Project Demonstration & Documentation

Title: AI-Based Quality Control in Manufacturing

Abstract:

This project introduces an AI-powered quality control system tailored for manufacturing units. It leverages computer vision and deep learning (YOLOv5) for defect detection, integrates with IoT devices for real-time image feeds, and uses secure logging for traceability. This final report details the full project lifecycle, from live demonstrations and system architecture to testing metrics, feedback, and deployment readiness. Screenshots, performance results, and annotated code are included.

Index

---

1. Project Demonstration

Overview:

The AI Quality Control system is presented in a live demo, illustrating its real-time defect detection capabilities, dashboard analytics, data logging, and user interactions.

Demonstration Details:

- **System Walkthrough:** Live demo of the platform detecting defects in products using camera feeds.
- **AI Detection Accuracy:** Shows YOLOv5 detecting scratches, missing parts, and misalignments in real-time.
- **IoT Integration:** Live camera inputs feed the inspection model continuously with defect feedback.

- **Performance Metrics:** Show model speed (FPS), defect detection accuracy, and dashboard load response.
- **Security & Traceability:** Display encrypted log files, defect records, and audit trails.

Outcome:
Demonstrates reliability, real-time capability, and readiness for industrial deployment.

---

2. Project Documentation

Overview:

Complete technical documentation and user guides to support system deployment and future development.

Documentation Sections:

- **System Architecture:**
  - YOLOv5 model integrated with OpenCV camera input.
  - Flask API for dashboard and data interface.
  - SQLite for logging inspection outcomes.
- **Code Documentation:**
  Scripts include:
  - `defect_detect.py`: Handles real-time detection
  - `iot_stream.py`: Captures camera input
  - `log_writer.py`: Encrypts and logs results
- **User Guide:**
  Interface walkthrough and instructions for factory floor operators.
- **Administrator Guide:**
  Configuration setup, performance tuning, and troubleshooting.
- **Testing Reports:**
  Precision, recall, throughput benchmarks, and error rate analytics.

Outcome:
Complete, accessible documentation for all users and maintainers.

---

3. Feedback and Final Adjustments

Overview:

Stakeholder feedback was collected and used to refine performance and usability.

Steps:

- **Feedback Collection:** Surveys from production engineers and operators.
- **Refinement:** Adjusted camera angles, improved model training, and added alert sounds for detections.
- **Final Testing:** Simulated high-speed production lines with multiple defect types.

Outcome:
Improved user interaction, higher detection reliability, and robust real-world performance.

---

4. Final Project Report Submission

Overview:

Summary of each project phase, major milestones, and outcomes.

Report Sections:

- **Executive Summary:**
  High-speed, AI-driven inspection system improving QC consistency and speed.
- **Phase Breakdown:**
  Covers data collection, model training, dashboard development, and performance testing.
- **Challenges & Solutions:**
  - Lighting variance: Solved with histogram equalization
  - False positives: Improved with extended training dataset
- **Outcomes:**
  Consistent >90% detection accuracy in test environments.

Outcome:
System is ready for scale and integration with manufacturing ERPs.

---

5. Project Handover and Future Works

Overview:

The project is handed over with instructions and future enhancement suggestions.

Handover Details:

- **Next Steps:**

    - Integrate barcode reading for product traceability
    - Expand dataset for more defect categories
    - Implement multilingual dashboard
- **Outcome:**
  Official handover complete with training materials and documentation package.

Screenshots and Sample Code

Sample: YOLOv5 Inference Code

```
import torch
from PIL import
```

```
model = torch.hub.load('ultralytics/yolov5', 'custom', path='best.pt')
img = Image.open('sample_product.jpg')
results = model(img)
results.show()
results.save()  # Save output image with bounding boxes
```

Screenshot Placeholders:

- YOLO model detecting defects
- Real-time dashboard alert popup
- Inspection log with encrypted timestamp
