



Automated Defect Product Detection

UNDERGRADUATE THESIS

**Submitted as one of the requirements to obtain
Sarjana Komputer**

By:

FITRA RAMDHAN HAFIDZ 001202200108

FACULTY OF COMPUTING INFORMATICS STUDY PROGRAM CIKARANG JULY, 2025

SIMILARITY INDEX



18% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- » Bibliography
- » Quoted Text

Match Groups

- 288** Not Cited or Quoted 17%
Matches with neither in-text citation nor quotation marks
- 17** Missing Quotations 1%
Matches that are still very similar to source material
- 0** Missing Citation 0%
Matches that have quotation marks, but no in-text citation
- 0** Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

- 9% Internet sources
- 6% Publications
- 17% Submitted works (Student Papers)

AI CHECK



*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify writing that is likely AI generated as AI generated and AI paraphrased or likely AI generated and AI paraphrased writing as only AI generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?

The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?

Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.



ABSTRACT

Traditional visual inspection of production lines is tedious, error-prone, and costly, and therefore inappropriate for high-volume production environments. The aim of this project is to investigate the potential of employing computer vision and Convolutional Neural Networks (CNNs) to produce automatic fault detection on manufacturing lines.

Since I was the AI engineer in charge of this project, model training, backend development, model testing, accuracy improvement, hyperparameter tuning, fine tune operations, and dataset expansion were activities that I was tasked with. The adopted strategy was large-scale testing of a series of architectures like Anomalib framework implementations, CNN models, HRNet, YOLO variants, and other deep learning methods. All the models were thoroughly evaluated on performance metrics including accuracy, precision, recall, F1-score, mean Average Precision (mAP), and processing speeds.

Key results indicate notable performance improvements in defect detection. Precise models achieved high accuracy with precision and recall maximized metrics with processing rates compatible with real-time industrial operations. The mAP scores and F1-scores validate consistent performance across different classes of defects and categories of products. Findings confirm the technical feasibility of implementing computer vision-based quality assurance systems in factories with the potential to transform factory quality assurance by eliminating human mistakes and saving huge sums of money through auto-defect detection.

Keywords : Anomalib, HRNet, YOLO, Defect, Hyperparameter, mAP

TABLE OF CONTENTS

STATEMENT OF ORIGINALITY.....	ii
SCIENTIFIC PUBLICATION APPROVAL FOR ACADEMIC INTEREST.....	
ADVISOR APPROVAL FOR JOURNAL/INSTITUTION'S REPOSITORY	iii
SIMILARITY INDEX REPORT.....	iv
GPT ZERO.....	v
ABSTRACT	vi
DEDICATION	vii
ACKNOWLEDGEMENTS	viii
TABLE OF CONTENTS	ix
LIST OF TABLES	x
LIST OF FIGURES	xi
I. INTRODUCTION	1
1.1 Problem Formulation	1
1. Background of the Problem	1
2. Problem Statement	2
3. Objective	3
4. Problem from a Customer Perspective	4
1.2 Problem Constraint Analysis.....	5
1. Aspect 1: Data Quality and Availability	5
2. Aspect 2: Model Performance.....	5
3. Aspect 3: Hardware Limitations	6
1.3 Problem Functional Analysis	6
Functionalities 1: Defect Detection Using Computer Vision	6
Functionalities 2: Real Time Defect Detection System	7
Functionalities 3: Data Logging & Reporting	7
1.4 Solution Selection.....	7
Alternative Solution 1: Convolutional Neural Networks (CNN).....	7

Alternative Solution 2: Recurrent Neural Networks (RNN)	8
Solution Selection.....	8
1.5 Solution Usage Scenarios	10
II. SYSTEM ANALYSIS.....	12
2.1 Existing Business Process.....	12
2.2 Proposed Business Process	13
2.3 Global Description of the Product.....	15
1. Main Functionality.....	16
2. User Characteristics	16
3. Product Development Environment.....	19
4. Product Operational Environment	20
2.4 Requirement Analysis.....	21
External Interface	21
Functional Description	22
Data Requirement	27
Functional Requirement.....	28
Activity Diagram	29
III. DESIGN	38
3.1 Alternative Solution Designs	38
3.2 Rational/Systematic Design	41
3.3 Hierarchical/Iterative Design	45
IV. IMPLEMENTATION.....	55
4.1 Functions/Procedure/Class Implementation.....	56
4.1.1 Add, Remove, Edit Data of Products	56
4.1.2 Capture and Analyze	58
4.1.3 Store Data.....	61
4.1.4 Generate Report.....	63
4.1.5 Monitor Defect Dashboard.....	64

4.1.6 Performance Monitoring.....	65
4.2 UI Implementation and Testing.....	150
V. CONCLUSION	152
REFERENCES	153
APPENDICES	154

LIST OF TABLES

1.1 Solution Selection.....	9
2.1 User Characteristics.....	16
2.2 Use Case: Add, Remove, Edit Data	22
2.3 Use Case: Capture and Analyze	23
2.4 Use Case: Store Data	24
2.5 Use Case: Generate Report	25
3.1 Alternative Solution Comparison Table.....	41
3.2 Quantitative Solution Selection.....	42

LIST OF FIGURES

2.1 Existing Business Process.....	13
2.2 Proposed Business Process	15
2.3 Use Case Diagram.....	22
2.4 Entity Relationship Diagram (ERD)	27
2.5 Context Diagram.....	28
2.6 Activity Diagram: Add, Remove, Edit Data	29
2.7 Activity Diagram: Capture and Analyze	30
2.8 Activity Diagram: Store Data	32
2.9 Activity Diagram: Generate Report	33
2.10 Activity Diagram: Monitor Defect Dashboard	34
2.11 Data Flow Diagram Level 1	35
3.1 Block Diagram Level 0	38
3.2 Data Flow Diagram ADPD	45
3.3 Use Case Diagram (Design).....	46
3.4 Activity Diagram: Add, Remove, Edit (Design).....	47
3.5 Activity Diagram: Capture and Analyze	48
3.6 Activity Diagram: Store Data (Design)	49
3.7 Activity Diagram: Generate Report (Design)	50
3.8 Activity Diagram: MonitorDashboard.....	51
3.9 Sequence Diagram: Add, Remove, Edit Data.....	52
3.10 Sequence Diagram: Capture, Analyze	53
3.11 Entity Relationship Diagram (Design)	54

4.1 Data Flow Diagram Implementation	55
4.2 UI Implementation Screenshot 1.....	150
4.3 UI Implementation Screenshot 2.....	151