

AUTOMATED DEFECT PRODUCT DETECTION



UNDERGRADUATE THESIS

Submitted as one of the requirements to obtain

Sarjana Komputer (S.Kom)

By:

SATRIA IBNU PAMUNGKAS

001202200071

FACULTY OF COMPUTER SCIENCE

INFORMATICS STUDY PROGRAM

CIKARANG

JULY, 2025

SIMILARITY INDEX REPORT 1



Page 1 of 132 - Cover Page

Submission ID: trn:oid::3618:109797551

Satria Ibnu Pamungkas

Individual Report - Satria ibnu pamungkas - 001202200071

Thesis Computing

Document Details

Submission ID

trn:oid::3618:109797551

Submission Date

Aug 27, 2025, 4:07 PM GMT+7

Download Date

Aug 27, 2025, 4:13 PM GMT+7

File Name

Individual Report - Satria ibnu pamungkas - 001202200071.docx

File Size

5.7 MB

123 Pages

11,062 Words

71,623 Characters



Page 1 of 132 - Cover Page

Submission ID: trn:oid::3618:109797551

SIMILARITY INDEX REPORT 2



Page 2 of 132 - Integrity Overview

Submission ID: tm:oid::3618:109797551

13% Overall Similarity

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

Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

	119 Not Cited or Quoted 13%
	Matches with neither in-text citation nor quotation marks
	1 Missing Quotations 0%
	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
2%		Publications
12%		Submitted works (Student Papers)



Page 2 of 132 - Integrity Overview

Submission ID: tm:oid::3618:109797551

AI WRITING 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.

TABLE OF CONTENTS

TABLE OF CONTENTS	xi
LIST OF TABLES	xiv
LIST OF FIGURES.....	xv
CHAPTER I INTRODUCTION	17
1.1.Problem Formulation	17
1.1.1 Background of the Problem	17
1.1.2 Problem Statement	19
1.1.3 Objective	19
1.1.4 Problem From Customer Perspective	19
1.2.Problem Constraint Analysis	20
1.2.1 Data Quality and Availability	20
1.2.2 Model Performance.....	21
1.2.3 Hardware Limitation	21
1.3.Problem Functional Analysis	22
1.3.1 Defect Detection Using Computer Vision	22
1.3.2 Realtime Defect Detection System	22
1.3.3 Data Logging and Reporting.....	23
1.4.Solution Selection	23
1.4.1 Convolutional Neural Networks (CNN)	23
1.4.2 Recurrent Neural Networks (RNN)	24
1.4.3 Selected Solution	24
1.5.Solution Usage Scenarios	25
1.5.1 Scenario for Defect Detection Using Computer Vision	25
1.6.Scenario for Data Logging and Reporting	26
CHAPTER II SPECIFICATION	27
2.1.Existing System	27

2.1.1	Existing Business Process.....	27
2.1.2	Proposed Business Process	29
2.2.	Global Description	30
2.2.1	Main Functionality	30
2.2.2	User Characteristics	32
2.2.3	Constraints	33
2.2.4	Product Development Environment.....	34
2.2.5	Product Operational Environment	35
2.3.	Requirement Analysis.....	36
2.3.1	External Interface	36
2.3.2	Functional Description.....	37
2.3.3	Data Requirement	41
2.3.4	Functional Requirement.....	42
2.3.5	Non-Functional Requirement.....	49
CHAPTER III	DESIGN	51
3.1.	Alternative Solution Designs	51
3.1.1	Algorithm.....	51
3.1.2	Implementations.....	52
3.2.	Rational/Systematic Design	53
3.2.1	Comparison Table	53
3.2.2	QUANTITATIVE SOLUTION SELECTION	53
3.2.4	Solution Selection	60
3.3.	Hierarchical/Iterative Design	61
3.3.1	Interface Information Between Block Diagram.....	61
3.3.2	Scrum Methodology to Implement steps in Software Engineering Design	63
3.3.3	Detail of Component and Library Used.....	64
3.3.4	Unified Modeling Language 2.0	67
CHAPTER IV	IMPLEMENTATION.....	76
4.1.	Functions/Procedure/Class implementation.....	76
4.1.1	Add, Remove, Edit and Save Data of the Products	76
4.1.2	Capture and Analyze	78
4.1.3	Store Data.....	79

4.1.4	Generate Report	85
4.1.5	Monitor Defect Dashboard	95
4.2.	UI Implementation and Testing	106
4.2.1	Registration & Login	106
4.2.2	Single Image Analysis Feature	110
4.2.3	Batch Image Analysis	112
4.2.4	Real-Time Analysis	115
4.2.5	Generate Report Feature	117
4.2.6	Detail Analysis	119
CHAPTER V CONCLUSION		122
REFERENCES		123
APPENDICES		124

LIST OF TABLES

Table 1.1 Advantage and Disadvantage of Solutions	25
Table 2.1 User Characteristics	32
Table 3.1 Entity Comparison	53
Table 3.2 Quantitative Rubric Evaluation	53
Table 3.3 Quantitative Solution Selection	58