



**IMPLEMENTATION OF TRANSFER LEARNING ON
YOLOv8 FOR DETECTION AND CLASSIFICATION
OF 2016 RUPIAH COINS SERIES**

By
Stewart
001201800002

A Final Project
Submitted to the Faculty of Computing
President University
in Partial Fulfillment of the Requirements
for the Degree of Sarjana Komputer
Information Technology

Cikarang, Bekasi, Indonesia
July, 2025

SIMILARITY INDEX REPORT

Final Project Draft Chapter 1 - 7 (Stewart, 001201800002)

ORIGINALITY REPORT

10%

SIMILARITY INDEX

8%

INTERNET SOURCES

3%

PUBLICATIONS

4%

STUDENT PAPERS

PRIMARY SOURCES

1

repository.president.ac.id

Internet Source

3%

2

Submitted to President University

Student Paper

2%

3

www.mdpi.com

Internet Source

<1%

4

www.coursehero.com

Internet Source

<1%

5

Submitted to University of Sydney

Student Paper

<1%

6

getpodcast.com

Internet Source

<1%

7

www.likeforex.com

Internet Source

<1%

8

riunet.upv.es

Internet Source

<1%

9

S. Prabu, K. Joseph Abraham Sundar, G. Shanmukhanjali, N. Jawali, K. Sharvani, V. Nirmala. "Indian Coin Detection and Recognition Using Deep Learning Algorithm", 2022 6th Asian Conference on Artificial Intelligence Technology (ACAIT), 2022

Publication

<1%

AI TURNITIN REPORT



*% 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

The demand for automated and precise currency identification systems is on the rise, fueled by advances in banking operations, retail market activities, and accessibility opportunities for people with visual impairments. This research focuses on the detection and classification of the 2016 Indonesian Rupiah coins, which creates a particular challenge, as these coins have a high degree of similarity within class and have significant variation in environmental conditions (lighting and occlusion).

The writer proposes a robust solution utilizing transfer learning on the state-of-the-art YOLOv8 (You Only Look Once version 8) object detection model. This will incorporate the fine-tuning of a pre-trained YOLOv8 model on our custom dataset of 2016 Rupiah coins with high rates of precision and recall. This study will provide a complete image dataset of the Rupiah coin from numerous real-world scenarios, and then undergo a rigorous model training and validation process. The intention is not to create yet another model as output, but to obtain an accurate and efficient model working in real-time immediately on coins, and to demonstrate the effectiveness of transfer learning as a potential pathway for hyper-local adaptation of advanced deep learning models for classification and detection of particular regions and objects. The results will help to contribute to the realm of applied computer vision and support a significant real-world problem of developing automated systems for handling coins in Indonesia.

TABLE OF CONTENTS

ABSTRACT.....	i
ACKNOWLEDGMENT.....	ii
TABLE OF CONTENTS.....	iii
LIST OF TABLES.....	vii
LIST OF FIGURES	viii
CHAPTER I.....	1
1.1. Background.....	1
1.2. Problem Statement	2
1.3. Objectives	4
1.4. Significance of the Study	4
1.5. Scope and Limitations.....	5
1.5.1. Scope.....	5
1.5.2. Limitations	6
1.6. Final Project Methodology	6
1.7. Final Project Structure	8
CHAPTER II.....	10
2.1. Introduction to Computer Vision and Object Detection	10
2.2. Object Detection Frameworks and Architectures	13
2.3. Deep Learning and Convolutional Neural Networks.....	17
2.4. YOLO (You Only Look Once) Architecture	21
2.5. YOLOv8 Architecture and Features	24

2.6.	Transfer Learning with YOLOv8	26
2.7.	Related Works.....	28
2.7.1.	Indian Coin Detection and Recognition.....	29
2.7.2.	Automated Detection of Ancient Indonesian Coins	30
2.7.3.	YOLOv8 Algorithm for Rupiah Denomination Detection	31
2.7.4.	Summary Table of Relevant Works.....	32
CHAPTER III		34
3.1.	System Overview	34
3.2.	Functional Analysis	35
3.3.	System Requirements.....	35
3.3.1.	Hardware Requirements.....	36
3.3.2.	Software Requirements.....	36
3.4.	Use Case Diagram.....	37
3.5.	Use Case Narrative	38
3.6.	Swim Lane Diagram	47
CHAPTER IV		50
4.1.	User Interface Design	50
4.1.1	Home Page	50
4.1.2.	How to Use Page.....	52
4.1.3.	Main Detection Page.....	53
4.2.	Dataset Design	55
CHAPTER V		57

5.1.	User Interface Development	57
5.1.1.	Home Page	57
5.1.2.	How to Use Page.....	58
5.1.3.	Main Detection Page.....	59
5.2.	Application Details	61
5.2.1.	Packages and Modules.....	61
5.2.2.	YOLOv8 Model Loading.....	62
5.2.3.	Camera Management and Video Stream	63
5.2.4.	Coin Detection Loop.....	64
5.2.5.	Data Management and Results Processing	66
5.3.	Model Development.....	68
5.3.1.	YOLOv8 Transfer Learning Training.....	68
5.3.2.	Model Training Results.....	69
CHAPTER VI		72
6.1.	Testing Environment.....	72
6.2.	Testing Scenario.....	72
6.2.1.	Home Page Testing Scenario	72
6.2.2.	Main Detection Page Testing Scenario	75
6.3.	Performance and Limitations Testing.....	82
6.3.1.	Qualitative Performance Testing	82
6.3.2.	Limitations Testing	90
CHAPTER VII		92

7.1.	Conclusion	92
7.2.	Future Work.....	93
REFERENCES		96

LIST OF TABLES

Table 2.1 Comparison Table of Relevant Works.....	32
Table 3.1 Functionality Table	35
Table 3.2 Hardware Requirements Table	36
Table 3.3 Software Requirements Table.....	36
Table 3.4 View Home Page – Use Case Narrative	38
Table 3.5 View Application Info – Use Case Narrative	39
Table 3.6 Access How to Use Guide – Use Case Narrative	39
Table 3.7 Start Detection Application – Use Case Narrative	40
Table 3.8 Provide Video Input – Use Case Narrative.....	41
Table 3.9 Detect Coin Objects – Use Case Narrative.....	42
Table 3.10 Classify Denominations – Use Case Narrative.....	43
Table 3.11 Count Coins by Denomination – Use Case Narrative	44
Table 3.12 Display Annotated Results – Use Case Narrative	45
Table 3.13 Export Detection Report – Use Case Narrative	46
Table 4.1 Home Page Description	51
Table 4.2 How to Use Page Description.....	52
Table 4.3 Main Detection Page Description	54
Table 6.1 Home Page Testing Scenario	74
Table 6.2 Main Detection Page Testing Scenarios	82
Table 6.3: System Limitations Identified During Testing	90

LIST OF FIGURES

Figure 1.1 Rapid Application Development (RAD) Diagram	6
Figure 2.1 Object Detection and Bounding Boxes	11
Figure 2.2 Object Detection Evolution Diagram	17
Figure 2.3 CNN Layers Figure	19
Figure 2.4 CNN Image Classification Flow	21
Figure 2.5 Illustration of YOLO Algorithm	22
Figure 2.6 YOLO Innovation Timeline	24
Figure 2.7 YOLOv8 architecture	26
Figure 2.7 Indian Coin Detection.....	29
Figure 2.8 Automated Detection of Ancient Indonesian Coin	30
Figure 2.9 Rupiah Banknotes Denomination Detection	31
Figure 3.1 Use Case Diagram	37
Figure 3.2 Swim Lane Diagram – Coin Detection Application.....	48
Figure 3.3 Swim Lane Diagram – Coin Detection Activity	49
Figure 4.1 User Interface - Home Page	50
Figure 4.2 User Interface – How to Use Page	52
Figure 4.3 User Interface - Main Detection Page	53
Figure 4.4 2016 Rupiah Coins Dataset	55
Figure 5.1 Welcome Page of the Application	58
Figure 5.2 How to Use Page of the Application	59
Figure 5.3 Main Detection Page of the Application	61
Figure 5.4 Packages and Modules Importing Code	62
Figure 5.5 Check GPU access Code	62

Figure 5.6 Loading the YOLOv8 Model Code	63
Figure 5.7 Start Camera Function Code	64
Figure 5.8 Real-time Detection Loop Code	65
Figure 5.9 Hold Current Batch Function Code	66
Figure 5.10 Show Total Function Code	67
Figure 5.11 Save Result Function Code	67
Figure 5.12 Model Training Code	68
Figure 5.13 Model Training Early Stopping Log	70
Figure 5.14 Final Model Performance Metrics	70
Figure 6.1 Application Home Page on Launch.....	73
Figure 6.2 Navigating to the "How to use" Page	73
Figure 6.3 Navigating to "Main Detection" Page	74
Figure 6.4 Activating the Camera and Real-Time Detection	75
Figure 6.5 Testing the "Hold Current Count" Feature	76
Figure 6.6 Testing the "Total" function	77
Figure 6.7 Testing the "Save" Feature	78
Figure 6.8 Generated Text File Result	78
Figure 6.9 Testing the "Clear" Feature	80
Figure 6.10: Testing the "Stop Camera" Functionality	81
Figure 6.11 Coins detected under Ideal Conditions.....	83
Figure 6.12 Incorrect Coin prediction in a low-light environment.....	84
Figure 6.13 Coins Predicted Incorrectly because of Direct Light Reflection.....	85
Figure 6.14 Detection of clustered and overlapping coins	86
Figure 6.15 Incorrect Prediction on a Tarnished Coin.....	88
Figure 6.16 Distance around 20cm effect on detection accuracy	89

Figure 6.17 Distance more than 20cm effect on detection accuracy	89
--	----