



TABLE OF CONTENT

PREFACE.....iv

TABLE OF CONTENT.....v

LIST OF FIGURES viii

LIST OF TABLESx

CHAPTER I 1

INTRODUCTION..... 1

 1.1 Background..... 1

 1.2 Problem Statement..... 2

 1.3 Problem Limitation 2

 1.4 Research Objectives..... 3

 1.5 Research Benefit 3

 1.6 Writing System 3

CHAPTER II 5

LITERATURE REVIEW 5

 2.1 Helmet Detection in Traffic Safety..... 5

 2.2 YOLOv8 in Helmet Detection 6

 2.3 Faster R-CNN in Helmet Detection 8

 2.5 Performance Comparison between YOLOv8 and Faster R-CNN 10

 2.6 Review of Previous Studies 12

CHAPTER III 17

RESEARCH METHODOLOGY 17

 3.1 Research Methodology Overview..... 17

 3.2 Research Framework..... 18

 3.3 Dataset and Annotation Process..... 20

 3.4 Data Augmentation 23

 3.5 Model Architecture and Configuration 24

 3.5.1 YOLOv8 Architecture..... 24

 3.5.2 Faster R-CNN Architecture 25

 3.5.3 Configuration Comparison..... 25

 3.6 Model Training Process 26

 3.7 Evaluation Metrics 26

 3.7.1 Precision, Recall, and F1-Score 26

 3.7.2 Mean Average Precision (mAP) 27

 3.7.3 Matthews Correlation Coefficient (MCC) 28

Robustness Evaluation	28
Video Inference Testing.....	29
CHAPTER IV	30
IMPLEMENTATION	30
4.1 Overview of Research Implementation.....	30
4.1.1. Tool Selection	31
4.1.2. Account Setup and Access	32
4.1.3. Google Colab Environment Setup	32
4.2 Dataset Collection and Annotation	34
4.2.1. Source of Dataset	34
4.2.2. Upload and Annotation in Roboflow	35
4.2.3. Dataset Splitting and Export	36
4.3 YOLOv8 Model Implementation.....	38
4.3.1. Environment and File Setup.....	38
4.3.2. YOLO Dataset Configuration (data.yaml).....	39
4.3.3. Training Parameters and Execution	39
4.4 Faster R-CNN Model Implementation.....	39
4.4.1. Dataset Preparation in COCO Format.....	40
4.4.2. Model Initialization and Modification	41
4.4.3. Training Script and Parameter Settings.....	41
4.5 Performance Evaluation.....	43
4.5.1. Classification Metrics	43
4.5.2. Detection Accuracy Metrics.....	44
4.5.3. Confidence Analysis	46
4.5.4. Inference Efficiency.....	49
4.5.5. Robustness	52
4.5.6. Video Based Performance.....	53
CHAPTER V	57
RESULT AND DISCUSSION	57
5.1 Introduction.....	57
5.2 Baseline Results.....	57
5.2.1. Classification Performance	58
5.2.2. Detection Accuracy.....	59
5.2.3. Confidence Analysis	60
5.2.4. Inference Efficiency.....	61
5.2.5. Robustness	61



5.2.6.	Video Inference.....	63
5.3	Summary of Results.....	64
5.4	Discussion.....	66
CHAPTER VI	69
CONCLUSION	69
6.1	Conclusion	69
6.2	Suggestions	69
REFERENCES	71



LIST OF FIGURES

UNIVERSITAS
GADJAH MADA

Figure 2. 1 YOLO 7

Figure 2. 2 R-CNN 8

Figure 3. 1 Research Workflow Diagram 20

Figure 3. 2 Distribution of the dataset across training, validation, and testing subsets 21

Figure 3. 3 Sample annotated image with bounding boxes for head, licensePlate, twowheeler, withHelmet, and withoutHelmet classes 22

Figure 3. 4 Sample robustness image Low Light with Blur, Overexposure, Rotation, Visual Noise Source: Google Colab 28

Figure 4. 1 Implementation Flowchart of the Helmet Detection System 31

Figure 4. 2 Google Colab runtime settings interface with Tesla T4 GPU selected 33

Figure 4. 3 Mounting of Google Drive in Google Colab environment 33

Figure 4. 4 Installing required libraries in Google Colab using pip commands 34

Figure 4. 5 Example of video frame containing motorcyclists 35

Figure 4. 6 Annotations generation flowchart 35

Figure 4. 7 Interface displaying 2,525 uploaded source images in Roboflow 35

Figure 4. 8 Sample annotated image with bounding boxes for five object classes 36

Figure 4. 9 Dataset split interface in Roboflow showing distribution into training, validation, and testing subsets 37

Figure 4. 10 Roboflow export interface showing selection of YOLO and COCO JSON formats 37

Figure 4. 11 Model Implementation YOLOv8 flowchart 38

Figure 4. 12 Command used to install Ultralytics and Roboflow libraries in Google Colab .. 38

Figure 4. 13 Python script to access and download YOLOv8-compatible dataset from Roboflow 38

Figure 4. 14 data.yaml configuration file for YOLOv8 training 39

Figure 4. 15 YOLOv8 training command using custom dataset and adjusted hyperparameters 39

Figure 4. 16 Model Implementation Faster R-CNN flowchart 40

Figure 4. 17 Directory structure of the COCO-formatted dataset used for Faster R-CNN training 40

Figure 4. 18 Python code to load and modify the Faster R-CNN model for custom class detection 41

Figure 4. 19 Manual training loop for Faster R-CNN in PyTorch using SGD optimizer 42



Universitas Gadjah Mada, 2025 | Diunduh dari <http://etd.repository.ugm.ac.id/>

Figure 4. 20 Performance Evaluation flowchart	43
Figure 4. 21 Python code for computing classification metrics.....	44
Figure 4. 22 Command to evaluate YOLOv8 model and compute mAP scores	45
Figure 4. 23 Python script using pycocotools to evaluate mAP for Faster R-CNN	45
Figure 4. 24 YOLOv8 implementation for extracting and analyzing confidence scores	47
Figure 4. 25 Faster R-CNN implementation for extracting and summarizing prediction confidence	48
Figure 4. 26 YOLOv8 inference time and FPS calculation using time module	50
Figure 4. 27 Faster R-CNN inference time and FPS measurement using PyTorch.....	51
Figure 4. 28 Python functions for generating robustness test cases: dark-blur, brightness, noise, and rotation.....	53
Figure 4. 29 YOLOv8 video inference implementation and performance measurement.....	55
Figure 4. 30 Faster R-CNN video inference and confidence tracking implementation.....	56
Figure 5. 1 Confusion matrix of YOLOv8 and Faster R-CNN	58
Figure 5. 2 Comparison of mAP scores between YOLOv8 and Faster R-CNN	59
Figure 5. 3 Distribution of confidence scores for YOLOv8 and Faster R-CNN	60
Figure 5. 4 Comparison of inference speed metrics between YOLOv8 and Faster R-CNN	61
Figure 5. 5 Average confidence (blur, lighting, noise, rotation)	62
Figure 5. 6 Video inference comparison between YOLOv8 and Faster R-CNN	63
Figure 5. 7 Comparison of consolidated evaluation metrics between YOLOv8 and Faster R- CNN	65
Figure 5. 8 Helmet detection using YOLOv8 in various real-world conditions.	67
Figure 5. 9 Helmet detection using Faster R-CNN in various real-world conditions.	67



LIST OF TABLES

Table 2. 1 Previous Research on Helmet Detection	14
Table 3. 1 Distribution of images across training, validation, and testing subsets	22
Table 3. 2 Comparison of model configurations used for YOLOv8m and Faster R-CNN.	25
Table 4. 1 Tool and Platform Used for Implementation	32
Table 5. 1 Classification metrics comparison between YOLOv8 and Faster R-CNN.....	58
Table 5. 2 Detection accuracy comparison between YOLOv8 and Faster R-CNN.....	59
Table 5. 3 Confidence score analysis of YOLOv8 and Faster R-CNN	60
Table 5. 4 Inference speed comparison between YOLOv8 and Faster R-CNN	61
Table 5. 5 Average confidence under visual perturbations (Robustness Testing).....	62
Table 5. 6 Video inference performance comparison between YOLOv8 and Faster R-CNN	63
Table 5. 7 Consolidated metrics across all dataset variations.....	64