A COMPARATIVE ANALYSIS OF YOLOV12 MODEL SIZES FOR ROAD DAMAGE DETECTION ON AN INDONESIAN DATASET



Conduct by:

Name : Azrial Ahmad Haidar Daffi Zuhri
Student ID : 21523211

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2025

SUPERVISOR ENDORSEMENT PAGE

A COMPARATIVE ANALYSIS OF YOLOV12 MODEL SIZES FOR ROAD DAMAGE DETECTION ON AN INDONESIAN DATASET

THESIS

Conduct By:

Name : Azrial Ahmad Haidar Daffi Zuhri

Student ID : 21523211

Yogyakarta, October 16th, 2025

Advisor,

(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

EXAMINER ENDORSEMENT PAGE

A COMPARATIVE ANALYSIS OF YOLOV12 MODEL SIZES FOR ROAD DAMAGE DETECTION ON AN INDONESIAN DATASET

THESIS

Has been defended in front of the examiners as one of the requirements to obtain a Bachelor of Informatics degree from the Undergraduate Program in Informatics at the Faculty of Industrial Technology, Universitas Islam Indonesia

Yogyakarta, November 4th, 2025

Chair

Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D. _____

Examiner 1

Ahmad Fathan Hidayatullah, S.T., M.Cs., Ph.D. _____

Examiner 2

Arrie Kurniawardhani, S.Si., M.Kom. _____

Acknowledged by,

Head of Undergraduate Program in Informatics

Faculty of Industrial Technology

Universitas Islam Indonesia

(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

AUTHENTICITY STATEMENT

The undersigned:

Name	:	Azrial Ahmad Haidar Daffi Zuhri
Student ID	:	21523211

Final project with title:

A COMPARATIVE ANALYSIS OF YOLOV12 MODEL SIZES FOR ROAD DAMAGE DETECTION ON AN INDONESIAN DATASET

Stating that all components and contents in this final project are my own work. If in the future it is proven that some parts of this work are not my own work, the final project submitted as my own work is ready to be withdrawn and ready to bear any risks and consequences.

Thus this statement letter is made, hopefully it can be used properly.

Yogyakarta, October 16th, 2025



(Azrial Ahmad Haidar Daffi Zuhri)

DEDICATION

I dedicate this thesis to my beloved Father and Mother, for their endless love, prayers, and support. To my advisors who guided me with patience, and to my fellow students who always gave me encouragement every step of the way.

MOTTO

This section may be freely written with any form of motto as long as it does not violate ethics. Ideally, this page should consist of one full page.

FOREWORD

All praise be to Allah SWT, for His blessings and grace which made the completion of this thesis, "A Comparative Analysis of YOLOv12 Model Sizes for Road Damage Detection on an Indonesian Dataset," possible.

This research was undertaken not only to fulfill academic requirements but also from a desire to contribute to the modernization of infrastructure maintenance in Indonesia. The author hopes that the data-driven framework developed herein will serve as a practical and beneficial reference for stakeholders, particularly the Ministry of Public Works, in making strategic decisions regarding the implementation of road monitoring technology.

The completion of this work would not have been possible without the support of many individuals. The author extends profound gratitude to the thesis advisor, Mr. Ir. Dhomas Hatta Fudholi, S.T., M.Eng., Ph. D, for his invaluable guidance, patience, and constructive feedback throughout this project. Sincere thanks are also due to the Ministry of Public Works (PU) for providing the essential dataset that formed the foundation of this research. Finally, the author is grateful to all the lecturers at Universitas Islam Indonesia, beloved family, and friends for their unwavering support and encouragement.

I acknowledge that this thesis is far from perfect and welcome any constructive criticism and suggestions for future improvement.

Yogyakarta, October 16th, 2025



(Azrial Ahmad Haidar Daffi Zuhri)

ABSTRACT

The sustainable maintenance of road infrastructure is a cornerstone of economic development and societal well-being. This thesis investigates the application of deep learning–based object detection for automated road damage assessment, focusing on a comparative evaluation of five YOLOv12 model variants ranging from nano to extra-large. The models were trained and tested on a large-scale dataset of Indonesian road imagery, which presents a challenging real-world environment with four distinct categories of road damage. The central aim of this research is to examine the trade-offs between detection accuracy and computational efficiency across different model capacities. Experimental results demonstrate a non-linear relationship between model size and performance, with the YOLOv12l variant achieving the highest accuracy. However, the relatively modest performance gap between the smallest and largest models indicates that lightweight architectures remain highly practical for deployment on resource-constrained edge devices. These findings underscore the importance of aligning model capacity with dataset characteristics and deployment requirements. Overall, this study contributes a benchmark analysis of YOLOv12 for road damage detection and highlights the broader implications of model efficiency in real-world AI applications.

Keywords: road damage detection, deep learning, object detection, YOLOv12, computer vision, model comparison.

GLOSSARY

The glossary contains specific terms used in the report that require explanation, such as uncommon borrowed terms. Arrange the terms alphabetically. Example:

Anchor Boxes	pre-defined boxes of various sizes and aspect ratios that are used as a starting point for predicting the location of objects. The model learns to adjust these anchor boxes to fit the ground-truth objects in an image.
Batch Size	a hyperparameter that defines the number of training examples used in one iteration. Processing images in "batches" helps stabilize the learning process.
Epoch	one complete pass through the entire training dataset. A model is typically trained for multiple epochs to learn the patterns in the data.
Ground Truth	the accurate, human-verified data used to train and evaluate a model. In your case, it consists of the road images and their corresponding correct bounding boxes and class labels for each damage.
Hyperparameters	The configuration settings that are set before the model training process begins, such as the learning rate, batch size, and number of epochs.
Inference	The process of using a trained model to make predictions on new, unseen data.
Overfitting	A common problem where a model learns the training data too well, including its noise, which causes it to perform poorly on new, unseen data.
Optimizer	An algorithm, such as SGD (Stochastic Gradient Descent), that adjusts the internal parameters of a neural network to minimize the difference between its predictions and the ground truth.

TABLE OF CONTENTS

SUPERVISOR ENDORSEMENT PAGE ii
EXAMINER ENDORSEMENT PAGE iii
AUTHENTICITY STATEMENT iv
DEDICATION v
MOTTO vi
FOREWORD vii
ABSTRACT viii
GLOSSARY ix
TABLE OF CONTENTS x
LIST OF TABLES xii
LIST OF FIGURES xiii
CHAPTER I INTRODUCTION 1
1.1 Background and Motivation 1
1.1.1 The Socio-Economic Imperative for Road Infrastructure in Indonesia 1
1.1.2 The Problem of Pavement Distress: Cost, Causes, and Current Assessment Methods 2
1.1.3 The Technological Shift: Automated Detection with Computer Vision and Machine Learning 4
1.1.4 The YOLO Framework: A Revolution in Real-Time Object Detection 4
1.1.5 YOLOv12: State-of-the Art Architecture and The Principle of Model Scaling 5
1.2 Problem Statement and Research Questions 8
1.3 Scope of the Work 9
1.4 Research Objectives and Contributions 10
1.5 General Research Methodology 10
1.6 Report Structure 12
CHAPTER II LITERATURE REVIEW 13
2.1 Problem Contextualization: The Imperative for Automated Road Damage Detection in Indonesia 13
2.2 A Comparative Analysis of Automated Road Damage Detection Methodologies 15
2.2.1 Foundational Two-Stage Approaches: The R-CNN Family 15
2.2.2 The Rise of Real-Time Single-Stage Detectors: SSD and the YOLO Lineage 16
2.2.3 The Emergence of Transformer-Based Architectures: The DETR Family 17
2.2.4 Categorized Comparison of Detection Architectures 18
2.3 Justification for the Selected Solution: The YOLOv12 Framework 20
2.4 An In-Depth Examination of the YOLOv12 Architecture 21
2.4.1 The Evolution of the YOLO Family 21
2.4.2 Core Architectural Innovations in YOLOv12 22
2.4.3 Model Scaling in YOLOv12: 25
2.5 Core Evaluation Metrics in Object Detection 26
2.5.1 Intersection over Union (IoU) 26
2.5.2 Precision and Recall 27
2.5.3 Average Precision (AP) and mAP50 27
CHAPTER III RESEARCH METHODOLOGY 29

3.1 Methodological Framework 29
3.2 Defining Indicators, Parameters, and Variables 31
3.2.1 Research Indicators 31
3.2.2 Research Parameters 31
3.3 Data Collection and Preprocessing 32
3.3.1 Data Source and Characteristics 32
3.3.2 Data Cleaning and Normalization 34
3.3.3 Image Transformation and Augmentation 34
3.3.4 Dataset Partitioning 35
3.4 Model Training 36
3.4.1 Training Environment and Configuration 36
3.5 Model Testing and Evaluation 38
3.5.1 Inference Environment 38
3.5.2 Testing Scenarios 38
CHAPTER IV RESULTS AND DISCUSSIONS 40
4.1 Quantitative Performance Analysis 40
4.1.1 Overall Model Comparison 40
4.1.2 Per-Class Detection Accuracy 41
4.1.3 Analysis of Training Process and Overfitting 42
4.2 Qualitative Performance Analysis 44
4.2.1 Case study 1: Pothole Detection 45
4.2.2 Case study 2: Complex Crack Detection 47
4.2.3 Case study 3: Alligator Crack Detection 49
4.3 Discussion 51
4.3.1 The Performance-Efficiency Trade-off 51
4.3.2 Impact of Dataset Characteristics 51
4.3.3 Practical Implications for Deployment 52
CHAPTER V CONCLUSION AND FUTURE WORK 53
5.1 Conclusion 53
5.2 Limitations 54
5.3 Future Work 54
REFERENCE 56

LIST OF TABLES

Table 1.1 The scale and Economic Impact of Indonesia's Road Network 2
Table 1.2 Overview of YOLOv12 Model Variants 7
Table 2.1 A Comparative Analysis of Deep Learning Architectures for Road Damage Detection 19
Table 3.1 Regional Distribution of labeled images in the dataset 32
Table 3.2 Distribution of damage instances by class in the dataset 33
Table 3.3 Data augmentation techniques applied during model training 35
Table 3.4 Final Training Hyperparameters 37
Table 4.1 Performance and Efficiency of YOLOv12 Variants 41
Table 4.2 Per-Class Detection Performance (mAP50) Across YOLOv12 Variants 42

LIST OF FIGURES

Figure 2.1 Architectural comparison of feature aggregation modules, highlighting the introduction of the residual connection in R-ELAN (d) used in YOLOv12.....24
Figure 2.2 Precision-Recall curves for two object classes, illustrating how Average Precision (AP) is computed. 28
Figure 3.1 Systematic workflow diagram illustrating the research methodology 30
Figure 4.1 Training and Validation loss curves for all five YOLOv12 Variants. 43
Figure 4.2 Original Test Image for Pothole Detection 45
Figure 4.3 Visual Output for All Five Models on the Test Imager for Pothole Detection 46
Figure 4.4 Original Test Image for Complex Crack Detection 47
Figure 4.5 visual output for all five models on the complex crack test image 48
Figure 4.6 Original Image for Alligator Crack Detection.....49
Figure 4.7 Visual output for all five models on the alligator crack test image.....51

CHAPTER I
INTRODUCTION

1.1 Background and Motivation

1.1.1 The Socio-Economic Imperative for Road Infrastructure in Indonesia

A well-maintained road network is not a luxury but a fundamental pillar of Indonesia's economic development, social cohesion, and national resilience. As the world's largest archipelagic nation, Indonesia faces unique and profound transportation challenges that directly impact its economic trajectory and the welfare of its population. The national road network serves as the primary circulatory system for commerce, enabling the efficient movement of goods and people across its vast and diverse islands. The sheer scale of this network is immense; while official reports in 2012 cited a total classified road network of over 477,000 km (World Bank, 2012), more recent data from collaborative mapping projects suggest a total length of approximately 1.5 million km, underscoring the monumental task of management and maintenance (Humanitarian OpenStreetMap Team (HOT), 2024). The government continues to invest in expansion, with plans to construct an additional 104 km of new national roads in 2025 alone (Investor Daily, 2024).

However, decades of under-investment have resulted in a significant infrastructure deficit, imposing substantial economic and social costs on the nation (Ray & Ing, 2016). Persistently high logistics costs, which account for an estimated 21% to 23% of the Gross Domestic Product (GDP), are a direct consequence of these infrastructural inefficiencies and serve as a major impediment to economic growth (Rubix Data Science, 2024). These high costs translate directly into elevated consumer prices and a reduction in the competitiveness of Indonesian industries in the global market

The relationship between road quality and economic prosperity is direct and quantifiable. Poor road conditions do not merely cause inconvenience; they actively suppress economic mobility and entrench regional inequality. Research indicates a strong causal link, showing that improvements in road quality can yield significant welfare gains. A rigorous analysis found that a 1% improvement in road quality in Indonesia increases household welfare by an estimated 0.1%. This effect is not driven by marginal time savings but by a fundamental economic shift: better roads enable workers to transition from low-wage, informal employment

into higher-wage formal sector jobs, particularly in manufacturing (Rothenberg, Gonzalez-Navarro, & Getler, 2022). This demonstrates that investing in road maintenance is a direct investment in poverty reduction and human capital development. By facilitating access to markets, education, and healthcare, a robust road network can unlock the economic potential of rural communities and foster more inclusive national growth. Therefore, the research presented in this thesis, which aims to enhance the efficiency of road maintenance, is directly aligned with these critical socio-economic objectives. The immense scale and profound economic importance of this challenge are summarized in Table 1.1.

Table 1.1 The scale and Economic Impact of Indonesia's Road Network

Metric	Value / Status	Source(s)
Total Road Network Length	> 477,000 km (classified, 2012) / ~1.5 million km (OSM, 2024)	(World Bank, 2012)
Road Condition (2019 Baseline)	National: 92% good; Provincial: 68% good; Regency/City: 57% good	(Asian Transport Observatory, 2025)
Road Condition (2024 Target)	National: 97% good; Provincial: 75% good; Regency/City: 65% good	(Asian Transport Observatory, 2025)
Budget for Regional Road Repair	2023: Rp14.7 trillion; 2024: Rp15 trillion	(antaranews.com, 2024)
Estimated Logistics Costs	21-23% of GDP	(Rubix Data Science, 2024)
Estimated Cost of Road Crashes	~3% of GDP (2021)	(Asian Transport Observatory, 2025)

1.1.2 The Problem of Pavement Distress: Cost, Causes, and Current Assessment Methods

Deficiencies in road infrastructure, ranging from minor surface defects like cracks to severe structural failures like potholes, impose immense direct and indirect costs on the

Indonesian economy and society. The most tragic of these are the human and economic losses from traffic accidents. Poor road conditions are a significant contributing factor to road traffic injuries (RTIs), which are a leading cause of death in the country (Kementerian PPN/Bappenas, 2020). The economic burden of these crashes is staggering, estimated at roughly 3% of Indonesia's GDP in 2021, a figure comparable to the nation's entire healthcare expenditure for the same year (Asian Transport Observatory, 2025). The National Development Planning Agency (Bappenas) underscores this crisis, noting that the majority of crash victims are of productive age, resulting in devastating losses to household welfare and national productivity (Kementerian PPN/Bappenas, 2020).

In response, the Indonesian government has allocated substantial financial resources toward road maintenance. In 2023, the budget for regional road repair was Rp14.7 trillion, followed by an

even larger allocation of Rp15 trillion (approximately US$953.51 million) in 2024 (antaranews.com, 2024). This massive investment is guided by Presidential Instruction to prioritize repairs on roads connecting production areas to markets, supporting tourism, and addressing heavily damaged routes (antaranews.com, 2024). However, a critical disparity exists between this high level of expenditure and the on-the-ground reality. Baseline data from 2019 revealed that only 68% of provincial roads and a mere 57% of regency/city roads were in good condition (Asian Transport Observatory, 2025). This points to a fundamental "maintenance efficiency gap," where the problem is not merely a lack of funding but a critical inefficiency in how those funds are identified, prioritized, and allocated.

The root cause of this efficiency gap lies in the nation's reliance on outdated data collection methodologies. The primary data source for crucial planning frameworks like the Integrated Road Management System (IRMS) remains the manual visual survey. This traditional process, where inspection teams physically travel roads to observe and record damage, is fundamentally ill-suited to the scale of Indonesia's challenges. It is notoriously labor-intensive, time-consuming, and expensive, making it impractical for the continuous monitoring of an extensive network. Furthermore, manual surveys are inherently subjective; the classification of damage type and severity can vary significantly between different inspectors, leading to inconsistent and unreliable data. This lack of objective, standardized information directly undermines effective planning and leads to the inefficient allocation of the multi-trillion Rupiah maintenance budget. While citizen-sourcing applications like "Jalan Kita" supplement these efforts, the foundational reliance on manual surveys remains a critical

bottleneck, perpetuating a cycle of reactive maintenance rather than proactive asset management.

1.1.3 The Technological Shift: Automated Detection with Computer Vision and Machine Learning

The inherent limitations of manual surveys, such as their high cost, subjectivity, and inefficiency, create a compelling case for a paradigm shift towards automated road damage detection (RDD) systems. The advent of Artificial Intelligence (AI), specifically within the domain of computer vision, presents a transformative alternative. AI-driven systems offer the potential to automate the inspection process with a level of objectivity, speed, and consistency that is simply unattainable through manual means, enabling authorities to transition from a reactive to a proactive maintenance strategy.

The technological evolution of automated RDD has progressed rapidly. Early systems often relied on specialized and expensive hardware, such as 3D laser scanning (LiDAR) or ground-penetrating radar, which, while detailed, were characterized by slow data acquisition and complex post-processing that limited their widespread adoption. Concurrently, early computer vision approaches used traditional image processing algorithms based on edge detection or texture analysis, but these methods were highly sensitive to variations in lighting and pavement conditions, often resulting in poor reliability.

A significant breakthrough occurred with the rise of deep learning, particularly the use of Convolutional Neural Networks (CNNs). CNNs marked a paradigm shift in computer vision by eliminating the need for manual, domain-expert-driven "feature engineering." Instead, these networks can learn hierarchical feature representations directly from raw pixel data, leading to dramatic improvements in detection accuracy and the ability to generalize across diverse conditions. This technological leap has made it feasible to develop robust RDD systems that can analyze imagery captured from ubiquitous and cost-effective devices, such as vehicle-mounted cameras or even standard smartphones. This flexibility is crucial for developing scalable solutions appropriate for the varied economic and logistical landscapes across Indonesia.

1.1.4 The YOLO Framework: A Revolution in Real-Time Object Detection

Within the broad landscape of deep learning, the "You Only Look Once" (YOLO) family of models has emerged as a particularly influential and powerful framework for object

detection tasks. Its unique architecture makes it exceptionally well-suited for applications like RDD, which demand a strong balance between processing speed and detection accuracy (Ali & Zhang, 2024).

Introduced in 2016, the original YOLOv1 revolutionized the field by reframing object detection not as a multi-stage classification problem but as a single, unified regression problem. Unlike two-stage detectors (e.g., the R-CNN family) that first propose potential regions of interest and then classify them, YOLO processes an entire image in a single pass through the neural network to predict all bounding boxes and their corresponding class probabilities simultaneously. This unified architecture resulted in unprecedented inference speeds, making real-time object detection on standard hardware a practical reality for the first time (Viswanatha, Chandana, & Ramachandra, 2025).

The success of this approach spurred a series of rapid and continuous innovations. Subsequent versions, from YOLOv2 to the widely adopted YOLOv8, have systematically addressed the limitations of their predecessors, progressively closing the accuracy gap with slower two-stage methods while maintaining or even improving upon real-time performance. These advancements have included more powerful backbone networks, the introduction of anchor boxes to improve localization, and multi-scale feature prediction to enhance the detection of small objects, which is a common challenge in identifying incipient road damage like fine cracks. This relentless evolution has cemented the YOLO framework as a leading choice for real-world applications that cannot afford to compromise on either speed or accuracy.

1.1.5 YOLOv12: State-of-the Art Architecture and The Principle of Model Scaling

This research specifically employs YOLOv12, a recent and state-of-the-art advancement in this influential lineage. The selection of YOLOv12 is motivated not only by its superior performance but, more importantly, by its advanced architecture and inherent design philosophy, which directly enable the central comparative investigation of this thesis.

The development of object detection architectures shows a clear trend of convergence, where the best ideas from previously distinct architectural families are being integrated. YOLOv12 is a prime example of this convergence. It is described as an "attention-centric" model that incorporates concepts traditionally associated with the Transformer family of architectures (e.g., DETR) but in a way that preserves the computational efficiency of the YOLO framework (Alif & Hussain, 2025). Its key architectural innovations include:

a. Residual Efficient Layer Aggregation Network (R-ELAN): An optimized backbone network designed to improve gradient flow and enhance feature extraction, allowing the model to learn more robust representations of damage from input images (Alif & Hussain, 2025).
b. Area Attention: A novel attention mechanism, often accelerated with FlashAttention, that allows the model to focus on the most salient regions of an image with significantly less computational overhead than traditional self-attention. This is crucial for accurately localizing damage in cluttered visual scenes (Alif & Hussain, 2025).
c. 7x7 Seperable Convolutions: An efficient method for preserving spatial context and location information with fewer parameters and computations compared to standard convolutional layers, further enhancing the model's balance of performance and efficiency (Alif & Hussain, 2025).

Crucially for this research, a defining feature of the modern YOLO family, continued in YOLOv12, is its release as a scalable family of models. These variants, typically ranging from Nano (n) to Small (s), Medium (m), Large (l), and Extra-Large (x), are created from a single base architecture by systematically adjusting its depth (number of layers) and width (number of channels). This design choice creates a clear and fundamental trade-off. Larger models, like YOLOv12x, have a higher parameter count and greater computational complexity, giving them a higher learning capacity to achieve superior accuracy, but at the cost of slower inference speeds and greater resource requirements. Conversely, smaller models, like YOLOv12n, are significantly faster and more lightweight, making them ideal for deployment on resource-constrained edge devices, though potentially with a compromise in accuracy.

This principle of model scaling is the central pillar of this thesis. By empirically evaluating each of these variants on a consistent dataset, this research aims to map out this exact performance-versus-cost curve. This provides a data-driven framework for selecting the optimal model for a given set of real-world constraints, a critical and practical question for Indonesian road authorities. The models that form the basis of this comparative study are detailed in Table 1.2.

Table 1.2 Overview of YOLOv12 Model Variants

Model Variant	Parameters (Millions)	Computational Load (GFLOPs)	Brief Description
YOLOv12n	2.6	6.5	Nano: Optimized for maximum speed and deployment on edge devices.
YOLOv12s	9.3	21.4	Small: A balanced model offering high speed with improved accuracy.
YOLOv12m	20.2	67.5	Medium: A general-purpose model with a strong balance of performance and cost
YOLOv12l	26.4	88.9	Large: A high-capacity model designed for superior detection accuracy.
YOLOv12x	59.1	199	Extra-Large: The highest-capacity variant, for applications where accuracy is paramount.

Source: Adapted From (Tian, Ye, & Doermann, 2025)

The main motivation for this comparative study stems from a critical, practical gap in existing research. While many studies propose a single model for road damage detection, they often fail to address the diverse operational and financial constraints faced by infrastructure managers. A high-accuracy model requiring powerful hardware may be suitable for a national-level survey, but completely impractical for regional agencies or for deployment on low-cost edge devices.

The YOLOv12 framework, with its inherent family of scalable models (nano, small, medium, large, and extra-large), provides a unique and ideal testbed to investigate this performance-versus-cost trade-off. This research, therefore, moves beyond simply asking "is YOLO good?" and instead addresses the more vital question: "which YOLO size offers the optimal balance of accuracy and efficiency for a given Indonesian deployment scenario?" This

provides a direct, data-driven guide for stakeholders to make informed, evidence-based technology adoption decisions.

1.2 Problem Statement and Research Questions

The preceding background analysis highlights several critical issues at the intersection of infrastructure management, economic necessity, and technological capability in Indonesia. These issues can be distilled into the following formal problem statements.

The current reliance on manual visual surveys for road condition assessment in Indonesia creates a significant data collection bottleneck. This method is inefficient, subjective, and costly, hindering the effective and timely allocation of national maintenance funds and perpetuating the cycle of infrastructure decay.

While deep learning models offer a promising solution for automated road damage detection, there is a critical lack of empirical, comparative analysis of state-of-the-art architectures, specifically concerning the trade-offs between different model sizes when applied to the unique and challenging conditions of Indonesian roads.

This absence of practical benchmarks leaves road authorities and infrastructure stakeholders without a data-driven framework for selecting the most appropriate model that balances detection accuracy with the computational and financial constraints of their specific deployment scenarios (e.g., high-end national survey vehicles vs. low-cost regional monitoring systems).

To address these problems, this thesis seeks to answer the following primary research question: How do the five scaled variants of the YOLOv12 object detection model (nano, small, medium, large, and extra-large) compare in terms of detection accuracy (Precision, Recall, mAP) and computational performance (inference speed, resource usage) when trained and evaluated on a large-scale, real-world Indonesian road damage dataset?

1.2.1 Research Gap and Positioning

Existing research in automated road damage detection (as detailed in Chapter 2) has largely focused on two streams: (1) proposing entirely new model architectures, or (2) applying an existing, single model (e.g., YOLOv8l or Faster R-CNN) to a specific dataset. While valuable, these studies often fail to address the critical, practical question of scalability and deployability. A significant research gap exists in the empirical, comparative analysis of a

family of models, where the architecture is consistent, but the computational cost and capacity are systematically varied.

This thesis is positioned to fill this specific gap. Instead of just proving that "deep learning works" for Indonesian roads, this study provides a direct, "apples-to-apples" comparison of the entire YOLOv12 model family. It moves the conversation from "which model architecture is best?" to "within a state-of-the-art architecture, what is the optimal performance-per-cost trade-off?" This positioning is crucial for providing practical, actionable guidance to stakeholders who must balance accuracy needs with real-world hardware and budget limitations, a context often overlooked in purely academic performance benchmarks.

1.3 Scope of the Work

This research is focused on a specific set of models, tasks, and conditions to ensure a rigorous and controlled comparative analysis. The scope of this work is defined as follows:

a. Model Focus: The research is focused exclusively on the comparative analysis of five pre-defined variants of the YOLOv12 architecture: YOLOv12n, YOLOv12s, YOLOv12m, YOLOv12l, and YOLOv12x. The study does not involve the design of a new architecture or comparison with other model families like SSD or Faster R-CNN.
b. Task Focus: The study concentrates on the object detection task of identifying and localizing four specific classes of road surface damage: Pothole, Longitudinal Crack, Lateral Crack, and Alligator Crack, as defined in the dataset. The research does not extend to damage severity classification or semantic segmentation.
c. Dataset Focus: The models are trained and evaluated using a proprietary Indonesian road damage dataset, which contains 5,915 labeled images captured from various provinces across Indonesia. The findings are specific to the characteristics and potential biases of this dataset.
d. Evaluation Focus: The comparison is based on a set of standard quantitative object detection metrics, including Precision, Recall, and mean Average Precision at an IoU threshold of 0.50 (mAP50), as well as computational performance metrics such as inference speed (milliseconds per image) and parameters (millions). The evaluation is supplemented with a qualitative analysis of visual detection results.
e. Environmental Focus: The research utilizes a consistent training environment on a high-performance computing (HPC) node equipped with NVIDIA Tesla V100 GPUs and a separate, defined testing environment on a workstation with an NVIDIA RTX 3080 GPU.

This ensures a fair comparison between models and simulates a practical deployment scenario.

1.4 Research Objectives and Contributions

Based on the defined problems and scope, this research is guided by a clear objective and aims to provide several tangible contributions to both academia and industry practice.

a. Research Objective:

To develop, train, and conduct a rigorous comparative analysis of five YOLOv12 model variants (n, s, m, l, x) to quantify the trade-offs between detection accuracy and computational efficiency for the task of road damage detection on a diverse Indonesian dataset.

b. Research Contributions:

1. Establishes Novel Benchmarks: This study will provide the first comprehensive performance benchmarks for the state-of-the-art YOLOv12 architecture on a large-scale, real-world dataset specific to Indonesian road conditions. This contributes new and relevant empirical knowledge to the field of intelligent transportation systems and computer vision.
2. Provides a Practical Deployment Guide: The research will deliver a quantitative, data-driven analysis of the performance-per-cost trade-offs for each model size. This serves as a practical guide for Indonesian road authorities, engineering consultants, and other stakeholders, enabling them to make informed, evidence-based decisions on model selection that align with their specific operational requirements, hardware capabilities, and budget constraints.
3. Investigates Model Scaling on Real-World Data: The findings will offer valuable insights into the complex interplay between model capacity, computational cost, and detection performance when faced with the inherent challenges of a real-world, imbalanced dataset. This contributes to the broader academic understanding of deploying advanced deep learning models in practical scenarios that exist beyond idealized public benchmarks.

1.5 General Research Methodology

To achieve the stated objectives, this research will be conducted following a systematic and structured methodology, which can be summarized in five distinct phases. This workflow

is designed to ensure a rigorous and reproducible investigation from initial problem formulation to the final analysis of results.

a. Phase 1: Literature Review and Problem Formulation: The research commences with a comprehensive review of existing literature. This includes an examination of academic papers, government reports, and technical documentation related to road infrastructure management in Indonesia, the evolution of automated damage detection techniques, and the architectural principles of the YOLO family of models. This foundational work is crucial for identifying the research gap and formally defining the problem statements, objectives, and scope of the thesis.
b. Phase 2: Dataset Preparation and Preprocessing: This phase focuses on the acquisition, cleaning, and preparation of the Indonesian road damage dataset for model training. Key activities include analyzing the dataset's geographical and class distribution, normalizing class labels, converting annotations from their original format to the YOLO-compatible format, and partitioning the 5,560 labeled images into a training set (80%) and a validation set (20%). To enhance model robustness and mitigate the effects of data imbalance, various data augmentation techniques will be systematically applied to the training data.
c. Phase 3: Model Training: In this core phase, the five YOLOv12 model variants (n, s, m, l, x) will be trained on the prepared dataset. Training will be executed in a controlled high-performance computing (HPC) environment. To ensure a fair and direct comparison, all models will be trained for a fixed number of epochs (300) using a consistent set of hyperparameters, including batch size, optimizer, and learning rate schedule.
d. Phase 4: Model Evaluation and Testing: Following the training phase, the performance of each of the five trained models will be systematically evaluated. This evaluation will be conducted on the held-out validation set using a dedicated testing hardware setup. Model performance will be measured using a suite of quantitative metrics, including Precision, Recall, and mean Average Precision (mAP50). Computational efficiency will be assessed by measuring inference speed. This quantitative assessment will be supplemented by a qualitative analysis of visual detection results on new, unseen images to assess practical performance.
e. Phase 5: Comparative Analysis and Report Writing: The final phase involves a thorough analysis and synthesis of the results collected in the previous phase. This includes a direct comparison of the five models across all metrics to articulate the performance-efficiency

trade-offs. The findings, discussion of their implications, and overall conclusions will be compiled and documented in the final thesis report.

This overall research flow provides a clear and logical path to answering the research question and achieving the project's objectives.

1.6 Report Structure

This thesis is organized into five chapters, each building upon the last to present a complete and coherent research narrative.

a. Chapter 1: Introduction: This chapter presents the background and motivation for the research, establishing the socio-economic context and the technological problem. It formally defines the problem statements, research questions, objectives, scope, and contributions, and concludes with an overview of the general research methodology and the structure of this report.
b. Chapter 2: Literature Review: This chapter provides the detailed theoretical foundation for the study. It delves into the problem of road management in Indonesia, offers a comprehensive review and comparison of automated road damage detection methodologies, and provides an in-depth technical examination of the YOLOv12 architecture and the fundamental principle of model scaling.
c. Chapter 3: Research Methodology: This chapter details the specific, technical steps undertaken to execute the research. It provides a thorough description of the Indonesian road damage dataset, the data preprocessing pipeline, the specific configurations used for model training, and the comprehensive framework designed for testing and evaluating the models.
d. Chapter 4: Results and Discussion: This chapter presents the empirical findings of the research. It includes a direct quantitative and qualitative comparison of the five trained YOLOv12 model variants. The results are analyzed in the context of the dataset's unique characteristics, and the practical implications of the observed performance-efficiency trade-offs are discussed in detail.
e. Chapter 5: Conclusion and Future Work: This final chapter summarizes the key findings and contributions of the thesis, providing a direct answer to the research question. It acknowledges any limitations of the study and proposes concrete and actionable directions for future work that could build upon the results and further advance the practical application of this technology in Indonesia.

CHAPTER II
LITERATURE REVIEW AND BACKGROUND THEORY

2.1 Problem Contextualization: The Imperative for Automated Road Damage Detection in Indonesia

The condition of a nation's road infrastructure is a direct indicator of its economic vitality and social cohesion. For Indonesia, an archipelagic state with a vast and geographically dispersed network, the maintenance of roads is not merely a matter of public works but a critical enabler of national development. However, the country faces profound challenges in this domain, stemming from a combination of socio-economic pressures, unique geographical constraints, and the limitations of traditional infrastructure management practices. The persistent issue of road damage necessitates a paradigm shift towards more efficient, objective, and scalable monitoring solutions.

The economic consequences of inadequate road infrastructure are severe and multifaceted. Poor road quality directly impedes economic activity, disrupts supply chains, and increases transportation costs, which ultimately affects consumer prices and reduces industrial competitiveness. Research indicates a strong positive correlation between road quality and local economic welfare; a 10% improvement in road conditions can lead to a 1.6% increase in household welfare, primarily by facilitating the transition of workers from informal sectors to higher-wage formal employment (Rothenberg et al., 2022). Conversely, damaged roads stifle local economies, particularly in rural areas, by hindering access to markets, education, and healthcare services, which can lead to the isolation of communities and exacerbate regional inequality (Rachman, Rifai, Rijaluddin, & Prasetijo, 2025). The financial burden is immense, with Indonesia spending an estimated USD 300-500 million annually on disaster recovery, a significant portion of which is allocated to repairing damaged roads and bridges. Furthermore, the economic losses attributable to road traffic crashes, many of which are exacerbated by poor road conditions, are estimated to be between 2.8% and 3.7% of the nation's Gross Domestic Product (GDP) annually (Kurohman, 2025).

These economic pressures are compounded by Indonesia's unique infrastructural and geographical landscape. The country comprises over 17,000 islands, connected by a road network that spans more than 546,000 kilometers. Maintaining this vast and fragmented network is a monumental logistical challenge. According to the Land Transportation Statistics 2023 published by BPS, only 43.98% of roads are in good condition, while the remainder are

classified as moderate (21.86%), damaged (12.84%), or seriously damaged (21.32%). In other words, more than one-third of Indonesia's road network is in a state of disrepair. This problem is aggravated by systemic issues, including a history of underinvestment in maintenance, complex bureaucratic structures involving central, provincial, and district governments, and a funding model that often prioritizes new construction over the upkeep of existing assets (Badan Pusat Statistik, 2024). The result is a cycle of deterioration that is difficult to break with conventional methods.

The current paradigm for road condition assessment relies heavily on manual inspections. This approach is fundamentally ill-suited to the scale and complexity of Indonesia's challenges. Manual surveys are notoriously labor-intensive, time-consuming, and expensive, making them inefficient for monitoring extensive road networks. Furthermore, they are inherently subjective; different inspectors may classify the type and severity of damage differently, leading to inconsistent data and inefficient allocation of limited maintenance resources. This lack of objective, standardized data undermines effective planning and prioritization, which is a critical failure in a system where funding is already constrained and difficult to navigate. Inspectors also face significant safety hazards while working on active roadways, adding a human cost to the process.

The confluence of these factors, the significant economic stakes, vast geographical scale, systemic maintenance challenges, and the inadequacy of traditional methods, creates a clear and urgent need for an automated road damage detection system. However, a viable solution for the Indonesian context cannot be evaluated on technical merit alone. It must be designed to address the specific constraints of the environment. From this contextual analysis, a set of essential criteria for any proposed technical solution emerges. The system must be:

a. Accurate: It must be capable of reliably detecting and classifying various types of road damage (e.g., potholes, alligator cracks, longitudinal cracks, lateral cracks) to provide actionable, high-fidelity data for maintenance planning.
b. Fast (Real-Time): The system must operate at high speeds, measured in Frames Per Second (FPS), to enable on-the-fly analysis from vehicles moving at normal traffic speeds. This real-time capability is essential for timely intervention and efficient, large-scale data collection.
c. Computationally Efficient & Scalable: The solution must offer a favorable balance between performance and computational cost. To be economically viable for nationwide deployment across various government levels, it must be deployable on a spectrum of

hardware, from high-end, dedicated survey vehicles to more accessible, resource-constrained platforms like smartphones or standard vehicle-mounted cameras. This necessitates a careful consideration of model size, parameter count, and computational load (FLOPs).

d. Objective and Consistent: By automating the detection process, the system must provide standardized, data-driven assessments. This removes the subjectivity inherent in manual inspections, enabling more transparent, evidence-based decision-making for maintenance prioritization and budget allocation.

Ultimately, the challenge is not simply to detect road damage, but to develop a system that can deliver objective, actionable data at a scale and cost appropriate for Indonesia's unique socio-economic and infrastructural realities. This understanding frames the subsequent review of technical solutions, placing a strong emphasis on the trade-offs between accuracy, speed, and computational efficiency.

2.2 A Comparative Analysis of Automated Road Damage Detection Methodologies

The advent of deep learning has revolutionized the field of computer vision, offering powerful tools for automated object detection. Over the last decade, several distinct architectural families have emerged, each with a unique approach to balancing accuracy, speed, and computational complexity. This section provides a comparative analysis of these dominant methodologies: two-stage detectors, single-stage detectors, and transformer-based models, evaluating their principles and performance in the context of road damage detection and their suitability against the criteria established in the preceding section.

2.2.1 Foundational Two-Stage Approaches: The R-CNN Family

Two-stage detectors, pioneered by the Region-based Convolutional Neural Network (R-CNN) and its successors, approach object detection with a deliberate, sequential methodology. The core principle involves two distinct steps: first, a dedicated network module, such as a Region Proposal Network (RPN), scans the image to generate a set of potential object locations, known as "regions of interest." Second, a separate classification network analyzes each of these proposed regions to determine if an object is present and, if so, to classify it (Silva, Leithardt, Batista, Villarrubia González, & De Paz Santana, 2023). This bifurcated process allows the model to focus its analytical power on a smaller number of promising image patches, which is a key reason for its characteristically high accuracy.

The most prominent member of this family is Faster R-CNN, which integrates the region proposal mechanism into a single, end-to-end trainable network. In the context of road damage detection, studies have shown that Faster R-CNN can achieve impressive accuracy, demonstrating a strong capability for identifying fine-grained details like cracks, even when trained on relatively small datasets. Some research has reported classification accuracies as high as 98.88% for specific damage categories using this approach.

However, the primary limitation of the two-stage architecture is its inherent computational overhead. The sequential nature of proposing and then classifying regions results in significantly slower inference speeds compared to single-stage alternatives. This latency makes two-stage detectors largely unsuitable for real-time applications, such as on-the-fly detection from a moving vehicle, which is a critical requirement for an efficient, large-scale road monitoring system. While their accuracy is high, their lack of speed presents a fundamental barrier to practical deployment in the Indonesian context.

2.2.2 The Rise of Real-Time Single-Stage Detectors: SSD and the YOLO Lineage

In response to the speed limitations of two-stage detectors, single-stage architectures were developed. These models streamline the detection process into a single pass through the neural network, simultaneously predicting bounding box locations and class probabilities for all objects in an image. This unified approach eliminates the need for a separate region proposal stage, enabling dramatic gains in inference speed (Liu et al., 2016) .The two most influential families in this category are the Single Shot MultiBox Detector (SSD) and You Only Look Once (YOLO).

The Single Shot MultiBox Detector (SSD) architecture typically uses a standard classification network (e.g., VGG, MobileNet) as a "backbone" for feature extraction and then adds a series of auxiliary convolutional layers at the end. These layers allow the model to make predictions at multiple feature map scales, which helps in detecting objects of various sizes (Liu et al., 2016). When paired with a lightweight backbone like MobileNet, SSD models can achieve an excellent balance of speed and efficiency, making them well-suited for deployment on mobile or embedded systems with limited computational resources (Bouazizi, Azroumahli, El Mourabit, & Oussouaddi, 2024). Performance on road damage tasks can be variable; some studies report modest mean Average Precision (mAP) scores around 13% on challenging datasets, particularly struggling with thin features like transverse cracks, while others have achieved over 65% mAP at a real-time speed of 71 FPS on general road-related objects.

The You Only Look Once (YOLO) framework represents a further conceptual simplification, framing object detection as a single regression problem. The input image is divided into a grid, and each grid cell is responsible for predicting the bounding boxes and class probabilities for any object whose center falls within it (Buhl, n.d.). The YOLO family has undergone a remarkably rapid and influential evolution, with each new version systematically improving the trade-off between speed and accuracy. While the earliest versions were known for their blistering speed at the cost of lower accuracy compared to two-stage detectors, subsequent iterations have progressively closed this accuracy gap. Starting with YOLOv4, significant architectural innovations such as advanced backbones, sophisticated feature fusion necks (e.g., Path Aggregation Network, PANet), and optimized loss functions have allowed YOLO models to achieve state-of-the-art accuracy while maintaining, and in some cases even improving, their real-time performance (Jegham, Koh, Abdelatti, & Hendawi, 2025). Recent studies utilizing models like YOLOv7 and YOLOv8 for road damage detection consistently report high performance, with mAP scores reaching 73.2% and 83.5%, respectively, demonstrating their strong suitability for this task (Dupont, 2024).

2.2.3 The Emergence of Transformer-Based Architectures: The DETR Family

A more recent paradigm shift in object detection has been driven by the introduction of transformer architectures, first popularized in the field of natural language processing. The Detection Transformer (DETR) re-imagines object detection by eliminating the need for many of the hand-crafted components that defined previous architectures, such as anchor boxes and non-maximum suppression (NMS) (Carion et al., 2020). A DETR model typically consists of a CNN backbone to extract image features, which are then fed into a transformer encoder-decoder architecture. The transformer uses a self-attention mechanism to model the global relationships between all features and directly outputs a final set of predictions for all objects in the image (Carion et al., 2020).

Initial DETR models, while conceptually elegant and highly accurate, were notoriously slow to train and had high inference latency (Zhu et al., 2021), making them impractical for real-time use. However, the field is evolving rapidly, and newer variants like RT-DETR (Real-Time DETR) have made substantial progress in optimizing the architecture for efficiency (Fairuz Mat Radzi, Amiruddin Abd Rahman, & Luqman Arif Bin Mohamad, 2025). One such model, Pavement-DETR, which is based on RT-DETR, has demonstrated state-of-the-art results on a public road damage dataset, achieving a mAP@0.5 of 87.1% (Zuo, Huang, Yuan,

& Li, 2025). This performance, which surpasses many YOLO variants, is attributed to its enhanced ability to handle small and overlapping defects by leveraging the global context provided by the attention mechanism.

Despite these impressive advancements, transformer-based detectors can still be more computationally demanding than the most efficient YOLO models. For instance, Pavement-DETR has a higher computational load (67.5 GFLOPs) than a comparable YOLOv10s model (24.5 GFLOPs) (Han et al., 2025). As a newer and more complex architecture, the ecosystem of tools, pre-trained models, and deployment pipelines is also less mature than that of the well-established YOLO family.

2.2.4 Categorized Comparison of Detection Architectures

To synthesize the preceding analysis and inform the selection of the most appropriate methodology for this research, the key characteristics, advantages, and disadvantages of each architectural family are summarized in Table 2.1. The evaluation is framed by their suitability for addressing the specific challenges of road damage detection in the Indonesian context.

Table 2.1 A Comparative Analysis of Deep Learning Architectures for Road Damage Detection

Architectural Family	Key Characteristics	Advantages	Disadvantages	Suitability for Indonesian Context
Two-Stage (e.g., Faster R-CNN)	Region Proposal + Classification. Anchor-based.	High accuracy, especially for complex/small objects. Performs well with limited data.	Computationally expensive. Slow inference speed, generally not real-time.	**Low.** The lack of real-time capability is a critical failure for large-scale, vehicle-based monitoring required in Indonesia. High computational cost makes it economically unscalable.
Single-Stage (e.g., SSD)	Single-pass detection. Anchor-based. Multi-scale feature maps.	Fast, real-time performance. Good for deployment on mobile/edge devices (Bouazizi et al., 2024).	Generally lower accuracy than two-stage methods. Can struggle with very small or overlapping objects.	**Medium.** Offers the required real-time speed and efficiency, but potential accuracy limitations might compromise the reliability of the data for maintenance decisions.
Single-Stage (e.g., YOLO Family)	Single-pass regression problem. Grid-based. Rapid evolution in architecture.	Excellent balance of high speed (real-time) and high accuracy. Scalable variants (n, s, m, l, x) available. Strong performance on road damage tasks (Silva et al., 2023).	Early versions struggled with small objects, though largely addressed in recent iterations. Performance can vary significantly between model sizes.	**High.** Represents the optimal trade-off between speed, accuracy, and scalability. The availability of different model sizes directly addresses the need for a flexible solution deployable across various hardware and budget constraints in Indonesia.
Transformer-based (e.g., DETR)	End-to-End detection without anchors/NKS. Attention Mechanism	Potentially state-of-the-art accuracy. Excellent at modeling object relationships. Newer variants (RT-DETR) are real-time capable	Can be computationally demanding and complex. Slower training convergence. The field is less mature than YOLO.	**Medium to High**. A very promising direction, but the YOLO family is more established, has a wider range of lightweight variants, and a simpler deployment pipeline, making it a more pragmatic and robust choice for this specific undergraduate research project.

The evolution of these architectures reveals a clear trend of convergence. While the field was once defined by a stark trade-off between the speed of single-stage detectors and the accuracy of two-stage detectors, modern models are blurring these lines. The YOLO series has aggressively incorporated sophisticated architectural components to rival the accuracy of two-stage methods. Simultaneously, the newest YOLO models, such as YOLOv12, are beginning to integrate attention-like mechanisms to further boost modeling capability without sacrificing speed, borrowing concepts from the transformer world (Tian et al., 2025). This dynamic landscape, where different architectural families learn from one another, reinforces the decision to select a state-of-the-art model that represents the current pinnacle of this converged approach.

2.3 Justification for the Selected Solution: The YOLOv12 Framework

The comparative analysis of deep learning architectures reveals that the You Only Look Once (YOLO) family provides the most compelling and pragmatic solution for the challenges of automated road damage detection in Indonesia. This framework uniquely satisfies the critical requirements of real-time speed, high accuracy, and scalability, making it the most suitable choice for this research. The selection of YOLOv12, the latest iteration of this lineage, is further justified by its state-of-the-art architecture and its inherent design philosophy, which directly enables the core experimental investigation of this thesis.

Based on the evaluation in Table 2.1, the YOLO architecture stands out as the optimal choice. Two-stage detectors like Faster R-CNN, despite their high accuracy, are disqualified by their slow inference speeds, which are incompatible with the need for real-time, vehicle-based monitoring. The SSD family, while fast, can present a compromise on accuracy that may affect the reliability of the collected data. Transformer-based models like DETR are a highly promising frontier, with recent variants achieving excellent performance; however, the YOLO family remains more mature, offers a broader ecosystem of tools, and provides a simpler and more established deployment pathway, making it a more robust and lower-risk choice for an undergraduate thesis project. YOLO, therefore, occupies the "sweet spot," offering an exceptional balance of the speed necessary for large-scale data acquisition and the accuracy required for generating meaningful maintenance insights.

The decision to use YOLOv12 specifically is driven by its position at the cutting edge of object detection research. YOLOv12 is not merely an incremental update; it is presented as an "attention-centric" framework that seeks to harness the powerful modeling capabilities of

attention mechanisms, which are a hallmark of transformers, but without incurring their typical computational cost and speed penalties (Tian et al., 2025).It has demonstrated superior performance over its immediate predecessors (YOLOv10 and YOLOv11) and has been shown to be competitive with, or even faster than, optimized transformer-based detectors like RT-DETR.[33] Selecting YOLOv12 allows this research to build upon the most current architectural advancements, ensuring its findings are relevant and contribute to the contemporary understanding of real-time object detection.

Most importantly, the selection of the YOLOv12 framework is fundamentally justified by the central research question of this Skripsi: to conduct a comprehensive performance comparison between different sizes of the model. The YOLO framework, since the introduction of YOLOv5, has been explicitly designed and released as a family of scalable models, typically ranging from nano (n) to small (s), medium (m), large (l), and extra-large (x). YOLOv12 continues this critical feature, providing a spectrum of pre-defined model sizes (Tian et al., 2025).

The availability of multiple, officially scaled variants is the key enabler for the entire experimental design of this thesis. It facilitates a systematic and controlled investigation into the fundamental trade-off between model complexity (measured by parameters and FLOPs) and performance (measured by mAP, FPS, and training time). This investigation directly addresses the practical deployment challenges faced by Indonesian road authorities. The findings will not produce a single, one-size-fits-all recommendation but rather a data-driven "performance-per-cost" curve. This allows stakeholders to make informed decisions, selecting, for example, a lightweight nano model for a low-cost smartphone-based monitoring initiative or a high-performance xlarge model for a dedicated, high-precision survey vehicle. The choice of YOLOv12 is therefore a strategic one; its architecture is not only state-of-the-art, but its design philosophy of providing a scalable family of models is what empowers this research to deliver a uniquely practical and contextually relevant contribution.

2.4 An In-Depth Examination of the YOLOv12 Architecture

2.4.1 The Evolution of the YOLO Family

The YOLO series has consistently pushed the boundaries of what is possible in real-time object detection. Each major release introduced key concepts that built upon the last, creating a powerful foundation for subsequent innovations.

a. YOLOv1 (2015): This was the revolutionary first step. It reframed object detection as a single regression problem, enabling unprecedented real-time speeds by eliminating the complex, multi-stage pipelines of its predecessors.
b. YOLOv2 (2016) & YOLOv3 (2018): These versions introduced critical refinements to improve accuracy. YOLOv2 incorporated anchor boxes to better predict objects of varying aspect ratios and used a new, more powerful backbone network called Darknet-19. YOLOv3 further enhanced this with the deeper Darknet-53 backbone and the introduction of predictions at three different scales, significantly improving the detection of small objects.[14]
c. YOLOv4 (2020) & YOLOv5 (2020): This era marked a focus on holistic optimization. YOLOv4 introduced a "bag of freebies" and "bag of specials". A collection of training techniques and architectural modules that boosted accuracy without increasing inference cost. YOLOv5, developed by Ultralytics, made a pivotal shift from the custom Darknet framework to the more accessible PyTorch framework. This move democratized the use of YOLO and formally introduced the concept of scalable model variants (n, s, m, l, x), which became a defining feature of the family.
d. YOLOv6 through YOLOv11 (2021-2024): This period saw continued refinement from various research groups, each proposing new backbones, feature fusion necks, and other optimizations to further improve the delicate balance between speed and accuracy, setting the stage for the next major architectural leap.

2.4.2 Core Architectural Innovations in YOLOv12

YOLOv12 represents a significant architectural evolution, shifting toward an attention-centric design that incorporates transformer-inspired concepts to enhance performance while preserving the real-time efficiency that defines the YOLO lineage (Tian et al., 2025). Its core innovations are concentrated in the backbone and neck of the network. At the backbone level, YOLOv12 introduces the Residual Efficient Layer Aggregation Network (R-ELAN), which improves gradient flow during training and helps prevent the vanishing gradient issue that often occurs in deep networks. By aggregating features across layers, R-ELAN generates richer and more robust feature maps, which are essential for detecting complex or small-scale objects such as fine cracks and varied textures in road damage (Tian et al., 2025). In addition to this, the model incorporates a novel attention mechanism called Area Attention (A2). Traditional self-attention, as used in transformers, is computationally expensive because its complexity

increases quadratically with the number of pixels. YOLOv12 addresses this by dividing the feature map into structurally adjacent regions and applying attention within and across these areas. This strategy allows the model to capture global context and concentrate on the most relevant parts of the image, such as damaged pavement, while significantly reducing computational load. The efficiency of this mechanism is further improved through FlashAttention, which optimizes memory access on the GPU (Tian et al., 2025). To encode spatial information, YOLOv12 uses a 7×7 separable convolution block known as the Position Perceiver. This block separates standard convolution into depthwise and pointwise operations, which reduces the number of parameters and computations. The large kernel size enables the model to maintain a wide receptive field, preserving spatial context and location information in an efficient manner (Tian et al., 2025).

To provide a more granular analysis, the architecture of YOLOv12 can be deconstructed into its three primary components: the Backbone, the Neck, and the Head.

A. Backbone: Residual Efficient Layer Aggregation Network (R-ELAN)

The backbone is the foundation of the detector, responsible for extracting rich, hierarchical feature maps from the input image. YOLOv12 introduces a novel backbone structure called the Residual Efficient Layer Aggregation Network (R-ELAN). This is a critical advancement over the ELAN and CSPNet structures used in prior yolo versions, designed to specifically address the optimization challenges that arise when intergrating attention mechanisms into very deep networks. As illustrated in the comparison from the source paper (Figure 2.X), the standard ELAN architecture (b) aggregates features across multiple paths to improve gradient flow. However, R-ELAN (d) introduces two key innovations. First, it incorporates a block-level residual connection with a scaling factor, which allows gradients to flow more directly through the block. This shortcut is crucial for stabilizing the training of high-capacity models (like the L and X variants) and preventing the "vanishing gradient" problem. Second, it redesigns the feature aggregation path into a more efficient bottleneck structure. This combination allows the backbone to learn more robust representations of complex visual patterns, such as varied road damage, in a stable and computationally efficient manner.

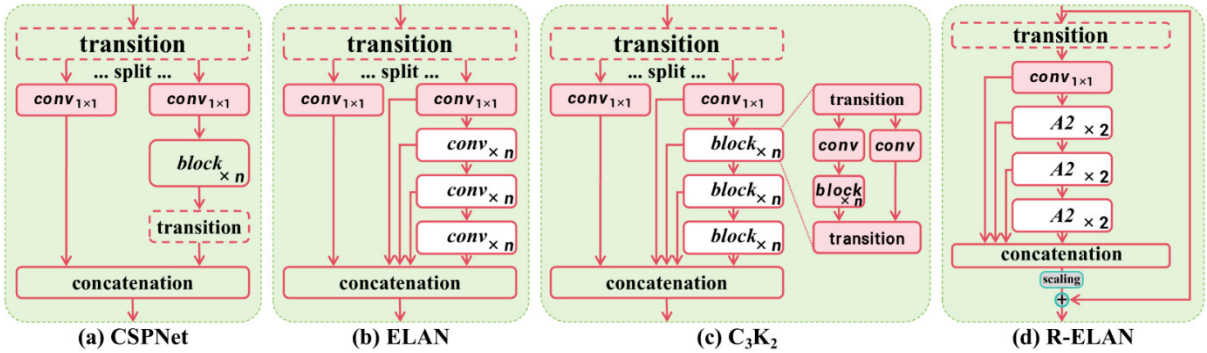


Figure 2.1 Architectural comparison of feature aggregation modules, highlighting the introduction of the residual connection in R-ELAN (d) used in YOLOv12

Source: (Tian et al., 2025)

B. Neck: Area Attention and Feature Fusion

The neck of the network fuses feature maps from different stages of the backbone to create representations that are rich in both semantic (what is it?) and spatial (where is it?) information. YOLOv12's primary innovation here is its Area Attention (A2) mechanism.

Traditional attention is computationally expensive. Area Attention offers an efficient alternative by partitioning the feature map into structurally adjacent regions (e.g., four horizontal stripes) and computing attention within these larger areas. As conceptualized in Figure 2.2 This allows the model to capture global context and focus on salient regions of an image (like a crack on a road) while dramatically reducing the computational load. The efficiency is further improved by using FlashAttention, an optimized implementation that minimizes slow memory access on the GPU. To help the attention mechanism perceive positional information without slow positional encodings, YOLOv12 also uses a Position Perceiver, which is a large (7x7) separable convolution block that efficiently preserves spatial context.



Figure 2.2 Comparison of the representative local attention mechanisms with YOLOv12's area attention.

Source: (Tian et al., 2025)

C. Detection Head and Model Scaling Principles

The detection head is the final component, taking the fused feature maps from the neck to predict bounding boxes and class probabilities. YOLOv12 uses an efficient, anchor-free detection head common in modern real-time detectors.

A defining feature of the YOLOv12 framework, and central to this thesis, is its inherent scalability. The architecture is released as a family of five officially scaled models: YOLOv12n (Nano), YOLOv12s (Small), YOLOv12m (Medium), YOLOv12l (Large), and YOLOv12x (Extra-Large). These variants are created from a single base architecture by systematically adjusting its depth (number of R-ELAN blocks) and width (number of channels in each layer). This principle creates a clear spectrum of models, from the lightweight 'n' variant designed for edge devices to the high-capacity 'x' variant designed for maximum accuracy, enabling the direct investigation of the performance-versus-cost trade-offs that form the core of this research. The head is the final component, taking the fused feature maps from the neck to predict bounding boxes and class probabilities.

2.4.3 Model Scaling in YOLOv12:

The principle of model scaling is fundamental to the YOLO family's practical utility and is central to the experimental design of this research. Scaling enables the creation of a spectrum of models from a single base architecture, each tailored to a different point on the performance-versus-cost curve. This scaling is primarily achieved by adjusting two compound coefficients for the network: depth and width. The depth of a network refers to the number of layers it contains, while the width refers to the number of channels (or filters) in each layer (Dlužnevskij, Stefanovič, & Ramanauskaitė, 2021). A larger model, such as YOLOv12-x, has greater depth and width than a smaller model, like YOLOv12-n, and this directly impacts the

model's capacity and computational requirements. Larger models, including YOLOv12-l and YOLOv12-x, contain a high number of parameters and a large computational footprint (FLOPs). Their increased depth and width give them greater learning capacity, allowing them to capture more complex and subtle features from the training data, which generally translates to higher detection accuracy (mAP). However, this comes at a cost: they process images more slowly (lower FPS), require more GPU memory (VRAM) for training and inference, take longer to train, and produce larger model weight files. These characteristics make them suitable for applications where accuracy is paramount and powerful hardware is available. In contrast, smaller models such as YOLOv12-n and YOLOv12-s reduce both depth and width, resulting in far fewer parameters and a much lower computational load. They are substantially faster (higher FPS), have smaller memory footprints, and generate compact model files, which makes them ideal for deployment on resource-constrained edge devices such as mobile phones or embedded systems in vehicles, where real-time speed and low power consumption are critical. The trade-off is that their reduced capacity may limit their ability to learn the most intricate patterns, which can result in lower overall accuracy compared to their larger counterparts. Importantly, the scaling strategy adopted in YOLOv12 is consistent with that of YOLOv11, reinforcing the continuity of design principles across the YOLO family (Tian et al., 2025). The existence of this scalable family of models is what makes the present research both possible and practical. By training and evaluating each of the YOLOv12 variants, from nano to extra-large, on the Indonesian road damage dataset, this study empirically maps out the trade-offs between accuracy and efficiency. The results, presented in the following chapters, provide a quantitative guide that directly addresses the practical question of which model size offers the best balance of performance and efficiency for a given set of deployment constraints, thereby contributing actionable insights to the field of intelligent transportation systems in Indonesia.

2.5 Core Evaluation Metrics in Object Detection

2.5.1 Intersection over Union (IoU)

Intersection over Union (IoU) is the fundamental metric used to measure the accuracy of a single predicted bounding box against its corresponding ground-truth bounding box. It quantifies the degree of overlap by calculating the ratio of the area of their intersection to the area of their union. The value ranges from 0 (no overlap) to 1 (perfect overlap). In object

detection, a prediction is typically considered a True Positive (TP) if its IoU with a ground-truth box is above a certain threshold (e.g., 0.5). If the IoU is below the threshold, it is a False Positive (FP). A ground-truth object that is not detected by any prediction is a False Negative (FN).

2.5.2 Precision and Recall

Precision and Recall are two core metrics that evaluate the quality of a model's predictions. Precision measures the reliability of the predictions, answering the question: "Of all the objects the model detected, what fraction was actually correct?" It is calculated as the ratio of True Positives to the total number of detections (TP + FP). Recall measures the model's sensitivity, answering the question: "Of all the actual objects present, what fraction did the model find?" It is calculated as the ratio of True Positives to the total number of ground-truth objects (TP + FN).

2.5.3 Average Precision (AP) and mAP50

In object detection, predictions are evaluated not only by class correctness but also by how well the predicted bounding box overlaps with the ground truth, measured by the Intersection over Union (IoU). The concepts of Average Precision (AP) and mean Average Precision (mAP) were popularized by computer vision challenges, most notably the PASCAL VOC challenge (Everingham, Van Gool, Williams, Winn, & Zisserman, 2010)

The Precision–Recall (P–R) curve shown in Figure 2.3 is generated by varying the confidence threshold of detections. At high thresholds, YOLO produces fewer detections, leading to high precision but low recall. As the threshold decreases, recall improves, but precision drops due to false positives.

The Average Precision (AP) for a class is defined as the area under this curve, often computed using interpolated precision values at fixed recall levels. The mean Average Precision (mAP) is then the mean of AP values across all classes, providing a single measure of overall detection performance.

When the IoU threshold is fixed at 0.50, the metric is denoted mAP50 (or mAP@0.5). This specific metric became a primary evaluation standard in earlier object detection benchmarks (Everingham et al., 2010). While stricter metrics like mAP@[.5:.95] (averaging AP over 10 IoU thresholds from 0.5 to 0.95), popularized by the COCO dataset (Lin et al., 2014), are now common, mAP50 remains a widely used and robust benchmark for YOLO models and is

appropriate for this study. As illustrated in Figure 2.3, detections with IoU ≥ 0.50 are counted as True Positives, while those below this threshold are False Positives. This makes mAP50 a relatively lenient but widely used benchmark in YOLO implementations, though stricter measures such as mAP@[.5:.95] are often used in COCO-style evaluations.



Figure 2.3 Precision-Recall curves for two object classes, illustrating how Average Precision (AP) is computed.

CHAPTER III
RESEARCH METHODOLOGY

3.1 Methodological Framework

This chapter presents the systematic methodology employed for the comparative analysis of YOLOv12 model variants for road damage detection. The research was centered on implementing five scaled versions of the YOLOv12 architecture, ranging from nano to extra-large, to empirically determine the trade-offs between model complexity, detection accuracy, and computational efficiency when applied to a challenging, real-world dataset of Indonesian road imagery. The methodology followed a structured, five-phase process as visually summarized in Figure 3.1, which details the workflow from problem formulation and data preparation to the final comparative analysis. The core of this research involved training the models on a specialized dataset provided by the Indonesian Ministry of Public Works and evaluating them using a rigorous set of quantitative and qualitative testing scenarios. This approach was designed to provide a robust and replicable foundation for assessing the practical viability of different model sizes for deployment in intelligent transportation systems.

The evolution of this research from its initial proposal to final execution involved a significant data curation and problem-scoping phase. An initial, large collection of over 30,000 images with sparse labeling was refined into a smaller, high-quality dataset of 5,915 fully labeled images, which became the cornerstone of this study. This deliberate refinement was essential for effective supervised learning and ensured that the subsequent model training and evaluation were based on reliable and consistent ground-truth data. The entire workflow was designed to be systematic, ensuring that each model variant was subjected to identical conditions, thereby isolating the architectural differences as the primary variable under investigation.

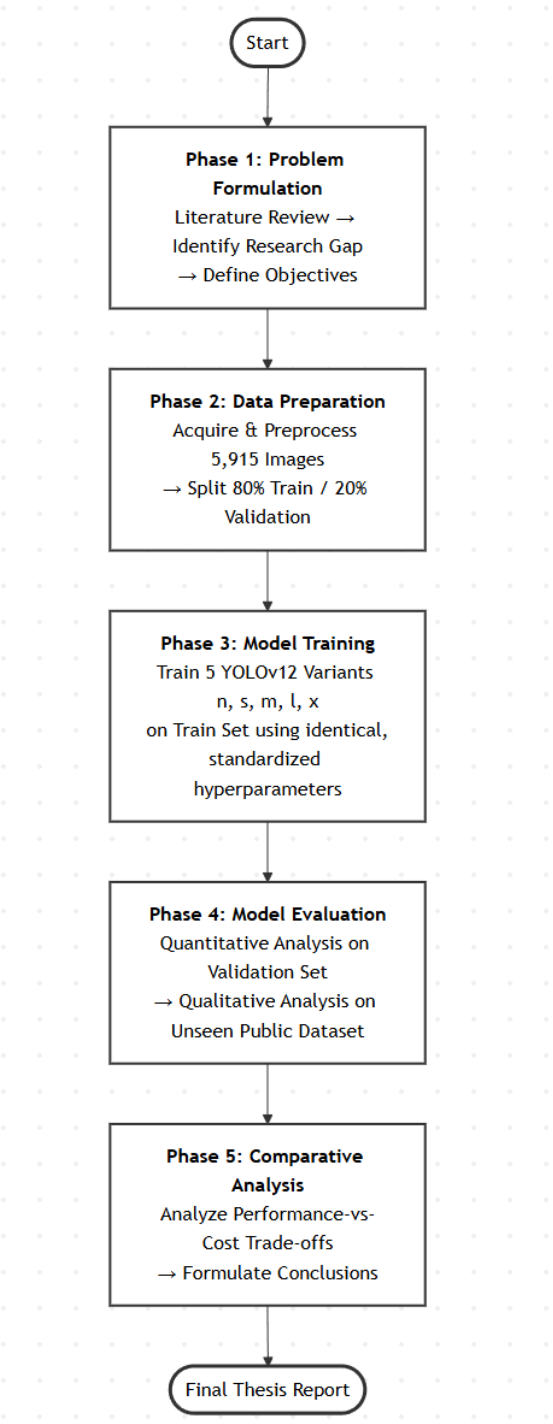


Figure 3.1 Systematic workflow diagram illustrating the research methodology

3.2 Defining Indicators, Parameters, and Variables

3.2.1 Research Indicators

The performance of the YOLOv12 model variants was evaluated using two primary indicators designed to capture both predictive quality and practical deployability. Detection accuracy reflected the model's ability to correctly identify and classify the four specified types of road damage within an image, thereby serving as a direct measure of the correctness of its predictions. Equally important, computational efficiency assessed the resources and time required for inference, providing insight into the feasibility of deploying the model in real-world, resource-constrained environments such as on-vehicle systems or edge devices. Together, these indicators offered a balanced evaluation framework, ensuring that improvements in accuracy were not achieved at the expense of efficiency, and vice versa.

3.2.2 Research Parameters

To quantitatively evaluate the research indicators, a specific set of performance parameters was selected, drawing on standard practices in the field of object detection to ensure a comprehensive basis for comparison. Detection accuracy was assessed using three key metrics. Precision (P) measures the reliability of the model's positive predictions by calculating the ratio of correctly identified damage instances (True Positives, TP) to the total number of predicted instances (TP + False Positives, FP), essentially answering the question of what fraction of detected damages were actually correct. Recall (R), on the other hand, evaluates the model's sensitivity by calculating the ratio of correctly identified damage instances (TP) to the total number of actual ground-truth instances (TP + False Negatives, FN), thereby indicating what fraction of existing damages were successfully detected. To provide a more comprehensive evaluation, mean Average Precision at an Intersection over Union threshold of 0.50 (mAP@0.5) was employed, representing the area under the precision–recall curve averaged across all four damage classes. A detection is considered a True Positive under this metric if the predicted bounding box overlaps with the ground-truth bounding box by at least 50% of their combined area. This measure was chosen for its robustness and suitability in road damage detection, where irregular shapes such as cracks make precise bounding box alignment particularly challenging. In addition to accuracy, computational efficiency was evaluated using two parameters. Giga Floating Point Operations (GFLOPs) quantify the theoretical computational complexity of a model's forward pass by measuring the total number of floating-

point operations required, offering a hardware-independent indicator of model complexity. Complementing this, inference speed, measured in milliseconds, reflects the wall-clock time required for a model to process a single image on the designated testing hardware, thereby providing a practical measure of real-world speed and throughput.

3.3 Data Collection and Preprocessing

3.3.1 Data Source and Characteristics

The dataset utilized for training and validation in this research is a private collection compiled by the Indonesian Ministry of Public Works (PU). As an internal dataset, it does not have a public citation. It contains 5,915 images captured across 20 distinct provinces and regions of Indonesia. While specific metadata on the capture devices is unavailable, the varied image resolutions—primarily 1920x1088 (3,839 images), 1920x1080 (1,416 images), and 1344x768 (305 images)—and the consistent ground-level perspective suggest the data was collected using a range of equipment, likely including vehicle-mounted dashcams and handheld digital cameras or smartphones

A critical characteristic of this dataset was its inherent imbalance, both geographically and in terms of class distribution. As shown in Table 3.1, the data collection was not uniform across regions, with some provinces like Papua Barat and Jawa Timur being heavily overrepresented.

Table 3.1 Regional Distribution of labeled images in the dataset

Region	Labeled Images
Papua - Barat	1,379
Jawa Timur	689
Bali	667
Banten	542
Sulawesi Utara	320
D.I Yogyakarta	271
Jawa Tengah	271

Papua – Merauke – Lingkar Brawijaya - PGT	247
Lampung	231
Jambi	225
Nusa Tenggara Timur	222
Papua – Merauke – Lingkar Brawijaya – SPRI	201
Bengkulu	192
Papua – Jayapura	177
Jawa Barat	88
Sumatera Selatan	69
Sumatera Utara	59
Maluku	37
Nusa Tenggara Barat	29
Bangka Belitung	26
Total Images	**5,915**

Furthermore, the dataset exhibited a severe class imbalance among the four annotated damage types: pothole, longitudinal_crack, lateral_crack, and alligator_crack. As detailed in Table 3.2, instances of pothole were more than twice as common as the next most frequent class, while alligator_crack and lateral_crack were significantly underrepresented. These imbalances posed a major methodological challenge, as models trained on such data can develop a bias towards the majority class, leading to poor performance on minority classes.

Table 3.2 Distribution of damage instances by class in the dataset

Damage Class	Instance Count
pothole	8,033
longitudinal_crack	3,299
lateral_crack	1,539
alligator_crack	1,410

3.3.2 Data Cleaning and Normalization

To ensure data quality and compatibility with the YOLOv12 framework, a structured multi-step data preparation process was carried out. First, all images without corresponding annotations were excluded from the initial dataset to maintain a purely supervised learning environment and avoid introducing noise during training. Next, annotation labels were programmatically standardized to a consistent format; for example, variations such as “Alligator crack” and “alligator crack” were normalized to alligator_crack in order to eliminate class ambiguity. The annotations were then converted into the YOLO text format, which required transforming bounding box coordinates from the original representation (top-left x, top-left y, width, height) into a normalized format (center-x, center-y, width, height), with all values expressed relative to the image dimensions.

3.3.3 Image Transformation and Augmentation

A standardized pipeline for image transformation was applied to all images before they were fed into the models. To address the challenges posed by the limited size and imbalanced nature of the dataset, a comprehensive suite of online data augmentation techniques was also employed during the training process. This strategy was not an arbitrary choice but a critical methodological step designed to artificially increase the diversity and volume of the training data, thereby improving model robustness and mitigating the risk of overfitting (Dunphy, Fekri, Grolinger, & Sadhu, 2022). The specific techniques and their parameters are detailed in Table 3.3.

Table 3.3 Data augmentation techniques applied during model training

Category	Technique	Description & Parameters
Geometric Transformations	Rotation	Images were randomly rotated within a range of degrees to simulate different camera angles and orientations.
	Horizontal Flipping	Images were randomly flipped horizontally with a 50% probability to teach the model orientation invariance.
	Translation	Images were randomly shifted horizontally and vertically by up to 10% of their dimensions (translate = 0.1).
	Scaling	Images were randomly scaled up or down by up to 50% (scale = 0.5) to improve detection of damages at various sizes.
Photometric Transformations	Brightness Adjustment	The brightness of images was randomly varied by up to 40% (hsv_v = 0.4) to simulate different lighting conditions (e.g., day, dusk, shadows).
	Contrast & Saturation	The contrast and saturation were randomly adjusted (hsv_s = 0.7, hsv_h = 0.015) to simulate different weather conditions and camera settings.
Advanced Augmentations	Mosaic Augmentation	Four training images were combined into a single 640x640 image, forcing the model to learn to identify objects in a wider variety of contexts and partial views (mosaic = 1.0).
	Mixup Augmentation	Two images and their corresponding labels were linearly interpolated and combined, a regularization technique that helps prevent the model from making overconfident predictions (mixup = 0.1).

3.3.4 Dataset Partitioning

The final curated dataset of 5,915 images was partitioned into training and validation sets. An 80/20 split was used, resulting in a training set of 4,732 images and a validation set of

1,183 images. To ensure that the severe class imbalance documented in Table 3.2 did not lead to a biased evaluation, stratified sampling was employed. This technique ensures that the proportional representation of each damage class was maintained across both the training and validation splits, providing a foundation for unbiased model training and a reliable assessment of generalization performance.

The dataset was intentionally partitioned into only training and validation sets, with no separate "test" set, for two primary reasons. First, given the modest size of the final curated dataset (5,915 images), creating a third split would have further reduced the amount of data available for the training process, which was already challenged by significant class imbalance. Second, the "validation" set in this study functions as the final test set for the comparative analysis. The models were trained on the training data, and their final performance was measured on the unseen validation data. No hyperparameter tuning or model selection was performed based on the validation set results; its sole purpose was to serve as a consistent benchmark to evaluate the final performance of the five pre-defined model variants. This approach ensures that the entire 20% of the data was held-out for a fair and final evaluation. To test the models' ability to generalize to data from a completely different source, a final evaluation was conducted on a separate, public dataset, as detailed in Sections 4.2 and 4.3. This serves as an additional, unbiased test of real-world performance.

3.4 Model Training

3.4.1 Training Environment and Configuration

To ensure a fair and direct comparison of the architectural differences between the model variants, all training was conducted in a consistent and controlled environment using a standardized set of hyperparameters.

a. Hardware Environment: All models were trained on the High-Performance Computing (HPC) Hanabi server at Universitas Islam Indonesia. The training process utilized two NVIDIA Tesla V100 GPUs, each equipped with 32GB of VRAM.
b. Training Hyperparameters: All models were trained using the standardized baseline configuration detailed in Table 3.4. These hyperparameters were chosen based on established best practices and empirical considerations, while the core optimizer settings were determined through a preliminary experimental phase, as detailed in the following section.

All models were trained for a maximum of 300 epochs with a batch size of 32 and an input image resolution of 640x640 pixels. These hyperparameters were chosen based on established best practices and empirical considerations. The 640x640 resolution is a common standard for YOLO models, providing a good balance between capturing sufficient image detail for detecting small damages like lateral cracks and managing computational load. The batch size of 32 was selected to maximize the utilization of the available GPU memory, leading to more stable gradient estimates and efficient training. A maximum training duration of 300 epochs was set as a practical limit for the experiment, to allow all model variants sufficient time to learn from the underrepresented damage classes, and also ensuring that even the largest model variant had sufficient time to converge. This was paired with an early stopping mechanism (patience of 50 epochs), which was the primary determinant for concluding training. This strategy ensures that each model was trained to its optimal point, preventing both under-training and excessive training.

Table 3.4 Final Training Hyperparameters

Hyperparameter	**Value**
Image Resolution	640x640 pixels
Batch Size	32
Optimizer	Stochastic Gradient Descent (SGD)
Initial Learning Rate	0.01
Momentum	0.937
Maximum Epochs	300
Early Stopping Patience	50 epochs

3.4.2 Hyperparameter Selection

The core training settings, specifically the optimizer and learning rate, were determined through a focused preliminary experimental phase. This was done to establish a robust and fair baseline for the main comparative analysis.

c. **Process:** A preliminary experimental phase was conducted to determine the optimal training configuration for the Indonesian road damage dataset. Various optimizers (e.g., Adam, SGD) and learning rates were evaluated on the medium-sized (YOLOv12m) model for a limited number of epochs.
d. **Outcome:** The Stochastic Gradient Descent (SGD) optimizer, with a high momentum of 0.937 and an initial learning rate of 0.01, was found to yield the most stable convergence and the lowest validation loss in these initial tests.
e. **Standardized Baseline:** This empirically determined configuration (SGD, lr=0.01, momentum=0.937) was then adopted as the standard baseline. It was applied consistently across all five YOLOv12 model variants to ensure that the final comparative analysis was a fair and direct comparison of their architectural differences, rather than a comparison of differently tuned models.

3.5 Model Testing and Evaluation

3.5.1 Inference Environment

To simulate a practical deployment environment that would be more representative of real-world applications than the high-end HPC training server, all inference testing was performed on a separate, dedicated hardware platform. This setup consisted of an NVIDIA RTX 3080 GPU with 10GB of VRAM, a Ryzen 7 5700X3D CPU, and 32GB of DDR4 RAM. All tests were run using Python 3.9 and PyTorch 1.12 with CUDA 12.8.1. Using this distinct hardware for testing provides a more realistic assessment of the models' computational efficiency and speed.

3.5.2 Testing Scenarios

A two-pronged evaluation was conducted to assess both the quantitative performance and the qualitative capabilities of the trained models. This dual approach provides a more holistic understanding than relying on metrics alone.

a. Quantitative Performance Assessment: The primary evaluation involved systematically running each of the five trained YOLOv12 variants on the entire validation set, which comprised 1,183 images. This process was executed under controlled conditions to ensure comparability:

1. Procedure: For each model, predictions (bounding boxes, class labels, and confidence scores) were generated for every image in the validation set. These predictions were then compared against the ground-truth annotations to determine the number of True Positives (TP), False Positives (FP), and False Negatives (FN) for each of the four damage classes.
2. Metrics Calculation: Based on the TP, FP, and FN counts, the primary performance parameters, Precision, Recall, and mAP@0.5, were calculated for each model. Concurrently, the average inference speed in milliseconds per image was measured to evaluate computational efficiency.
3. Analysis Focus: This scenario provided the core empirical data for the comparative analysis. The objective was to directly compare the accuracy-efficiency trade-offs across the five model variants and identify which architecture offered the optimal balance for this specific road damage detection task.

b. Qualitative Inference Analysis: To move beyond internal metrics and visually assess the models' real-world detection capabilities, a secondary evaluation was performed. This involved using each model to run inference on a set of new, unseen images.
    1. Procedure: Representative test images were selected from the public "Road Damage Indonesia" dataset available on Roboflow (Kantor, 2023). This dataset was entirely separate from the private training and validation data, providing a true test of generalization. Each of the five models was used to generate detections on these images.
    2. Analysis Focus: The resulting detections were visually inspected and compared. This qualitative analysis served to validate and contextualize the quantitative findings. For example, it allowed for the observation of specific failure modes (e.g., missed detections in complex scenes) or strengths (e.g., superior localization of small cracks). This step was crucial for understanding how and why the models performed as they did, providing insights that numerical metrics alone cannot capture. The inclusion of this test on a public dataset also enhances the transparency and generalizability of the research findings.

CHAPTER IV
RESULTS AND DISCUSSIONS

4.1 Quantitative Performance Analysis

4.1.1 Overall Model Comparison

The initial analysis begins with the primary performance indicators that map the fundamental trade-off between computational cost and detection accuracy. Table 4.1 presents this trade-off in detail, including the precise computational and training metrics from the final experiments. The 'Best Epoch (Actual)' column reflects the epoch at which the highest mAP50 was recorded, as determined by the early stopping mechanism (patience=50). The 'Total Training Time' column shows the full duration of each training run, executed on two NVIDIA Tesla V100 GPUs.

An interesting and non-linear trend emerged in the training times. The total training duration is a factor of both the model's computational load (GFLOPs) and the number of epochs it ran before early stopping was triggered. The YOLOv12x, being the largest model, expectedly had the longest training time (5.085 hours). Conversely, the YOLOv12m model, which began overfitting the earliest (best epoch at 92), completed its run in the shortest time (2.616 hours). This premature overfitting also resulted in a lower final mAP50 (0.327) than the smaller YOLOv12s model (0.334), making it a less optimal choice in the model lineup.

Reading from top to bottom, a clear and expected trend emerges: as model size increases from the lightweight YOLOv12n (2.6M parameters) to the high-capacity YOLOv12x (59.0M parameters), the computational requirements increase proportionally. The GFLOPs surge from 6.3 to 198.5, and the practical Inference Speed slows from a remarkable 1.0 ms to 6.9 ms.

However, the most critical story this table tells lies in the mAP50 column, which does not follow this linear progression. Accuracy improves from the 'n' to the 's' variant and then peaks with the YOLOv12l model at a mAP50 of 0.343. Counter-intuitively, the largest and most computationally expensive model, YOLOv12x, sees a performance decrease, achieving a lower mAP50 of 0.336. This non-linear result is the first key finding, immediately suggesting

that for this specific dataset, simply increasing model capacity has diminishing returns and, beyond an optimal point, becomes actively detrimental to performance.

Table 4.1 Performance and Efficiency of YOLOv12 Variants

Model Variant	Parameters (M)	GFLOPs	Inference Speed (ms)	Best Epoch (Actual)	Training Time (hours)	Precision	Recall	mAP50
YOLOv12n	2.6	6.3	1.0	117	2.785	0.417	0.343	0.317
YOLOv12s	9.2	21.2	1.2	107	2.783	0.450	0.346	0.334
YOLOv12m	20.1	67.1	2.5	92	2.616	0.457	0.325	0.327
YOLOv12l	26.3	88.6	4.6	80	3.466	0.431	0.356	**0.343**
YOLOv12x	59.0	198.5	6.9	144	5.085	0.424	0.366	0.336

4.1.2 Per-Class Detection Accuracy

To understand why the overall mAP50 scores behaved as they did, a more granular analysis of each model's performance on the four distinct damage classes is necessary. Table 4.2 breaks down this performance. A consistent pattern is immediately visible across all five models.

The alligator_crack class, for instance, stands out as the most reliably detected, achieving the highest mAP50 scores across the board and peaking at 0.418 with the YOLOv12l model. This is a particularly insightful result because, as established in Chapter 3, alligator_crack is one of the least frequent classes in the dataset. Its high performance suggests that its visual features—a distinctive web of interconnected cracks—are so unique that the models can learn to identify it effectively even with fewer examples.

Conversely, the story for lateral_crack is one of consistent struggle. It achieved the lowest mAP50 scores for every model variant, peaking at only 0.273. This poor performance is a direct reflection of the severe class imbalance in the training data, where lateral_crack instances were significantly underrepresented. Unlike the visually distinct alligator cracks, thin lateral cracks likely require more examples for the model to learn to differentiate them from non-damage pavement markings or shadows. This table, therefore, tells a crucial story about the interplay between a class's visual distinctiveness and its representation in the dataset.

Model Variant	mAP50 (all)	pothole	lateral_crack	longitudinal_Crack	alligator_crack
YOLOv12n	0.317	0.323	0.238	0.329	0.377
YOLOv12s	0.334	0.342	0.269	0.333	0.394
YOLOv12m	0.327	0.338	0.247	0.326	0.398
YOLOv12l	**0.343**	**0.343**	0.265	**0.348**	**0.418**
YOLOv12x	0.336	0.344	**0.273**	0.327	0.400

Table 4.2 Per-Class Detection Performance (mAP50) Across YOLOv12 Variants

4.1.3 Analysis of Training Process and Overfitting

The non-linear performance observed in the YOLOv12x model necessitates an analysis of the training process itself Figure 4.1 displays the training and validation loss curves for all five model variants. Overfitting is a common problem in machine learning where a model learns the training data too well, including its specific noise and quirks, rather than the general underlying patterns. This results in a model that performs exceptionally well on data it has already seen (training loss decreases) but poorly on new, unseen data (validation loss stagnates or increases).

The evidence for this is the visible divergence between the training loss (blue line) and the validation loss (red line) in the graphs. For all five models, the training loss consistently decreases, showing the models are learning the training data. However, the validation loss (red line) eventually flattens out and begins to rise, indicating that the model has stopped generalizing and is starting to memorize the training set.

This divergence is a classic indicator of overfitting, which is visible to some degree in all five model variants. However, the impact of this overfitting is most clearly demonstrated by the largest models. The YOLOv12x model, for example, trained for the most epochs (best epoch at 144), yet its final mAP50 score (0.336) was lower than that of the smaller YOLOv12l model (0.343). This strongly suggests that while all models overfitted, the massive capacity of the YOLOv12x (59.0M parameters) made it the most impacted by overfitting. Its high capacity, when applied to a dataset of this size, was detrimental to generalization, causing it to "memorize" the training data rather than learn the underlying patterns, which ultimately resulted in lower performance on the unseen validation data.

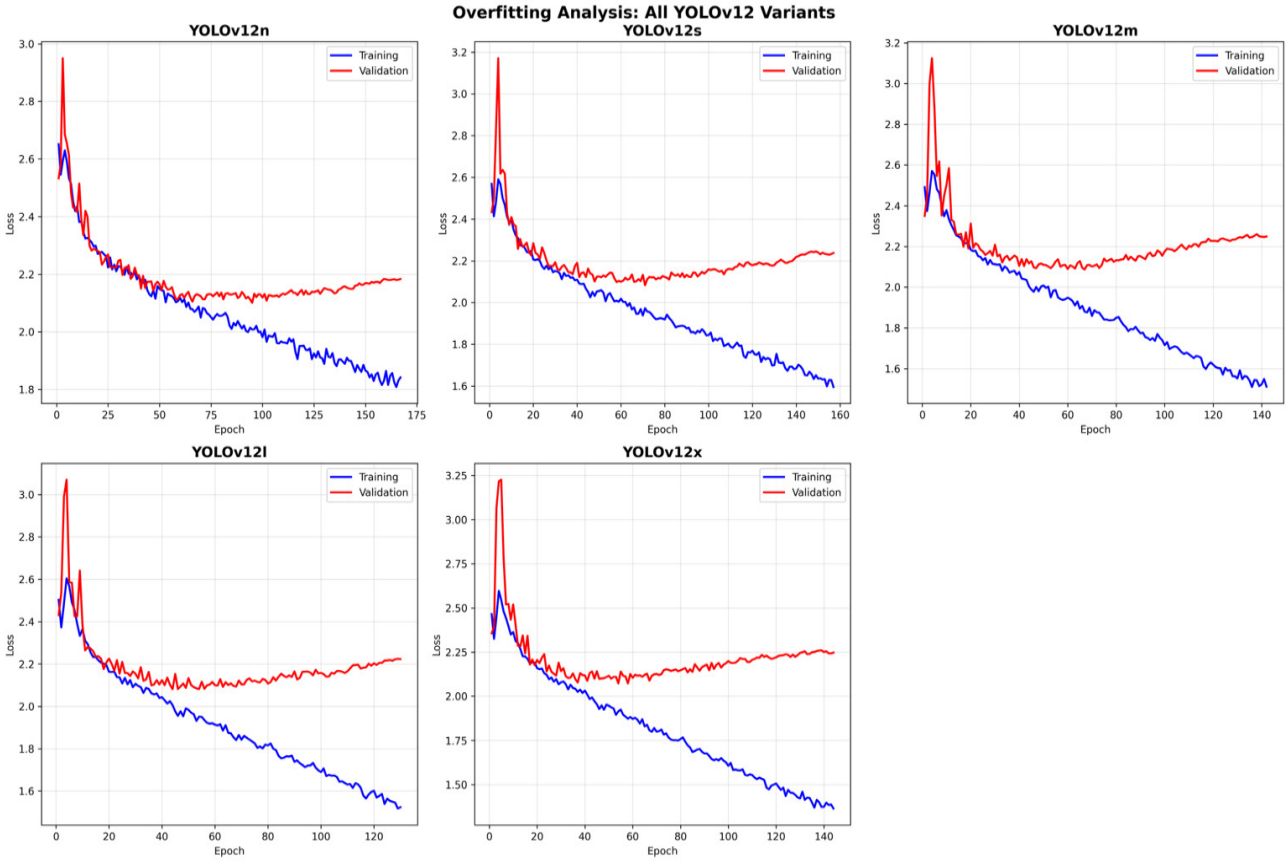


Figure 4.1 Training and Validation loss curves for all five YOLOv12 Variants.

4.2 Analysis of Generalization on Public Dataset

A core part of this research is to evaluate how well the models, trained exclusively on the private Ministry of Public Works (PU) dataset, can generalize to data from a completely different and unseen source. For this, the "Road Damage Indonesia" dataset (Kantor, 2023) was used as an unbiased public benchmark. As the public dataset used different class names and indices, a remapping process was performed to ensure the class indices of the test data matched the indices the models were trained on, enabling a fair comparison.

4.2.1 Quantitative Generalization (Domain Gap)

To rigorously measure generalization, a full quantitative validation was performed for all five trained models on the remapped public dataset. The results are presented in Table 4.3 and compared directly against the performance on the internal validation set.

Table 4.3 Generalization Performance: Internal Validation vs. Public Dataset (mAP50)

Model Variant	Internal Validation (mAP50)	Public Dataset (mAP50)	Performance Drop (mAP50)	Relative Drop (%)
YOLOv12n	0.317	0.245	-0.072	-22.7%
YOLOv12s	0.334	0.257	-0.077	-23.1%
YOLOv12m	0.327	0.251	-0.076	-23.2%
YOLOv12l	**0.343**	**0.270**	-0.063	**-21.3%**
YOLOv12x	0.336	0.270	**-0.066**	-19.6%

As shown in the table, a significant performance drop of 20-23% was observed for all model variants when tested on the public dataset. This phenomenon is a classic example of "domain gap" or "domain shift." It indicates that the models were highly optimized for the specific visual characteristics of the private PU training data (e.t, camera types, time of day, pavement textures, and annotation styles).

Further analysis of the metrics reveals the cause: the performance drop was almost entirely driven by a catastrophic collapse in recall. While the internal validation recall was ≈0.32-0.37 (Table 4.1), the recall on the public dataset was only ≈0.15-0.18. In contrast, precision remained relatively high (≈0.35-0.43). This means the models were not sensitive enough to find the damages in the new dataset, even though the damages they did find were often correct.

This finding does not invalidate the models; rather, it highlights a critical real-world challenge. It demonstrates that a model trained in one region or with one set of equipment may not perform at the same level when deployed in a new environment without re-training or fine-tuning. Interestingly, the YOLOv12x model saw the smallest relative performance drop, suggesting its high capacity may have captured more generalized features that made it slightly more robust to the new domain.

4.3 Qualitative Performance Analysis

To visually validate the quantitative findings, a qualitative analysis was performed by running inference on sample images from the public dataset. These case studies help to illustrate the practical detection capabilities and failure modes of the models when faced with new data.

4.3.1 Case study 1: Pothole Detection

The first test image, shown in Figure 4.2, features a complex road scene with three distinct potholes of varying sizes and levels of obscurity. The comparative detection results from all five models are displayed in Figure 4.3.



Figure 4.2 Original Test Image for Pothole Detection

Source: (Kantor, 2023)

The visual results clearly demonstrate the impact of model size on detection sensitivity. The smaller n and s models consistently failed to detect the third, less prominent pothole. In contrast, the m, l, and xl models successfully identified all three potholes, showcasing their superior ability to detect objects in cluttered environments.



(a) YOLOv12n Output



(b) YOLOv12s Output



(c) YOLOv12m Output



(d) YOLOv12l Output



(e) YOLOv12x Output

Figure 4.3 Visual Output for All Five Models on the Test Imager for Pothole Detection

4.3.2 Case study 2: Complex Crack Detection

The second case study (Figure 4.4) presents a more challenging scenario with multiple, intersecting longitudinal and lateral cracks. As shown in Figure 4.5, this scene proved difficult for the smaller models.



Figure 4.4 Original Test Image for Complex Crack Detection

Source: (Kantor, 2023)

The n and s models failed to identify all the cracks present in the scene, missing at least one of each type. The larger m, l, and xl models, however, demonstrated their enhanced feature extraction capabilities by successfully identifying all five distinct cracks. This highlights the advantage of higher model capacity when dealing with fine-grained and overlapping objects.



(a) YOLOv12n Output



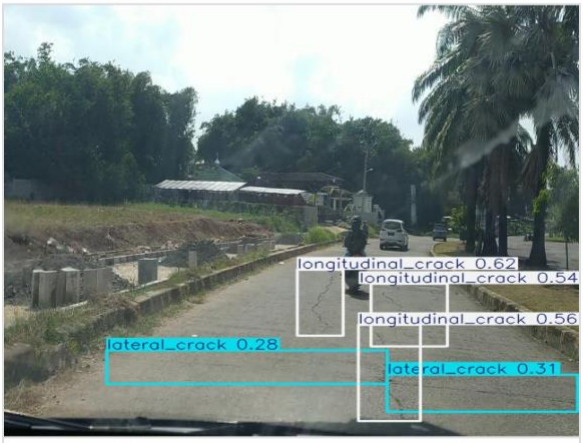
(b) YOLOv12s Output



(c) YOLOv12m Output



(d) YOLOv12l Output



(e) YOLOv12x Output

Figure 4.5 visual output for all five models on the complex crack test image

4.3.3 Case study 3: Alligator Crack Detection

The final test case (Figure 4.6) features a clear instance of an alligator crack. The detection results in Figure 4.7 are particularly revealing, as they provide visual evidence of the overfitting discussed in the quantitative analysis.



Figure 4.6 Original Image for Alligator Crack Detection

Source: (Kantor, 2023)

While all models successfully identified the primary alligator crack with high confidence, the smallest variant, YOLOv12n, produced a low-confidence false positive by misclassifying a non-existent feature as a 'pothole.' This likely stems from the model's limited representational capacity and reduced spatial resolution, which can cause it to misinterpret surface textures or

shadows as meaningful objects. Such behavior is common in highly compressed architectures that prioritize speed and efficiency over contextual nuance.



(a) YOLOv12n Output



(b) YOLOv12s Output



(c) YOLOv12m Output



(d) YOLOv12l Output



(e) YOLOv12x Output

Figure 4.7 Visual output for all five models on the alligator crack test image

4.4 Discussion

4.4.1 The Performance-Efficiency Trade-off

The central finding of this research is the non-linear relationship between model size and detection accuracy for this specific task. The results clearly demonstrate that while larger models are more computationally expensive, they do not always yield better results. The YOLOv12l variant emerged as the top performer, striking an optimal balance between model capacity and generalization.

However, the performance difference between the best model (YOLOv12l, mAP50 0.343) and the smallest, most efficient model (YOLOv12n, mAP50 0.317) is relatively small (a difference of ~8%). This has profound practical implications. Given that the YOLOv12n is over four times faster and requires significantly fewer computational resources, it presents a highly compelling option for real-world deployment on resource-constrained edge devices, such as mobile phones or in-vehicle systems. For this specific 4-class detection problem, the marginal gain in accuracy from the larger models may not justify the substantial increase in computational cost and latency.

4.4.2 Impact of Dataset Characteristics

The results cannot be interpreted in a vacuum; they are intrinsically linked to the characteristics of the training data detailed in Chapter 3. The severe class imbalance, particularly the underrepresentation of lateral_crack, is the most likely cause for its consistently poor detection performance across all models. Furthermore, the overfitting observed in the YOLOv12x model is a direct consequence of its high learning capacity being applied to a dataset of limited size and diversity. A model with nearly 60 million parameters requires a vast and varied dataset to learn generalizable features effectively; when the data is insufficient, the model is prone to memorizing training examples, as evidenced by its lower mAP50 and the false positive in the qualitative analysis

4.4.3 Practical Implications for Deployment

The findings of this research provide a data-driven framework for stakeholders, such as the Indonesian Ministry of Public Works, to make informed decisions about model selection based on their specific operational needs and constraints.

a. For High-Speed, Large-Scale Monitoring: For applications where speed and efficiency are paramount, such as a citizen-sourcing mobile app or a low-cost, vehicle-mounted system for continuous monitoring, the YOLOv12n model is the clear recommendation. It offers respectable accuracy at a fraction of the computational cost.
b. For High-Accuracy Surveys: For applications where achieving the highest possible accuracy is the primary goal, such as in a dedicated survey vehicle used for detailed engineering assessments or for generating ground-truth data, the YOLOv12l model is the optimal choice. It provides the best detection performance without the negative effects of overfitting seen in the larger model.

The results definitively show that a "one-size-fits-all" approach is inappropriate. The optimal model choice is contingent upon the specific deployment scenario and the balance required between accuracy and computational resources.

CHAPTER V
CONCLUSION AND FUTURE WORK

5.1 Conclusion

This research was motivated by the critical need for an efficient, objective, and scalable solution for road infrastructure monitoring in Indonesia to overcome the limitations of traditional manual surveys. The primary objective was to conduct a rigorous comparative analysis of five scaled variants of the state-of-the-art YOLOv12 object detection architecture to quantify the trade-offs between detection accuracy and computational efficiency on a real-world Indonesian road damage dataset.

The study began by identifying the core problems facing Indonesian road maintenance: the inefficiency and subjectivity of manual data collection, which hinders the effective allocation of national funds, and the absence of a practical, data-driven framework for selecting an appropriate technological solution. This research directly confronted these issues by systematically evaluating the YOLOv12 family, and the findings provide a clear resolution. The study successfully demonstrated that the relationship between model size and performance is non-linear. The YOLOv12l (Large) variant achieved the peak detection accuracy with a mean Average Precision (mAP50) of 0.343. However, a clear case of diminishing returns was observed, as the largest model, YOLOv12x, yielded a lower score due to overfitting. This proves that for this task, simply increasing model size does not guarantee better results. Furthermore, the study highlighted the high efficiency of the YOLOv12n (Nano) variant, which offered a compelling balance of performance and efficiency, making it an excellent candidate for real-world, resource-constrained applications.

In conclusion, this research successfully fulfilled its objective by providing the data-driven framework that was previously missing. It maps the performance-per-cost curve for a state-of-the-art architecture and concludes that the optimal model choice is not absolute but is contingent upon the specific deployment scenario. It offers clear, evidence-based recommendations for stakeholders, directly answering the need for a more strategic approach to adopting automated road monitoring technology.

5.2 Limitations

To ensure a fair interpretation of the results, it is important to acknowledge the limitations that defined the scope and context of this research. The findings are first and foremost intrinsically linked to the characteristics of the private dataset used, which, as detailed in Chapter 3, contains significant geographical and class imbalances, as well as potential annotation inconsistencies or other flaws in the dataset that undoubtedly influenced model performance. The study's scope was also intentionally focused on a deep comparative analysis of the YOLOv12 architecture, and therefore did not include a comparative benchmark against other prominent object detection families like SSD or Transformer-based models. Finally, the dataset may not comprehensively represent all possible environmental conditions encountered on Indonesian roads, meaning the models' performance under adverse weather or lighting conditions was not explicitly evaluated. The study's results were also impacted by model overfitting, as discussed in Chapter 4, which suggests that the training configuration and regularization techniques could be further optimized.

5.3 Future Work

Based on the findings and limitations identified in this study, several directions are proposed for future research. The most critical next step is a data-centric enhancement, involving a comprehensive data auditing and cleaning campaign to correct potential annotation errors, followed by a targeted data collection strategy to rectify the identified imbalances and the application of advanced data augmentation to improve robustness. Simultaneously, an expanded architectural benchmarking study should be conducted to situate the performance of YOLOv12 in the broader landscape by comparing it against other leading architectures.

Beyond these technical refinements, the ultimate goal of practical application should be pursued through a Pilot Program and Policy Integration. This next phase would involve a direct collaboration with stakeholders such as the Ministry of Public Works to deploy the most promising models (YOLOv12l for high-accuracy surveys and YOLOv12n for rapid mobile assessment) in a controlled, real-world pilot on a designated national road segment. Such a program would not only validate the technology in an operational environment but also provide invaluable data on implementation costs, personnel training requirements, and the workflow modifications needed to integrate this automated data stream into the existing Integrated Road Management System (IRMS). The findings from this pilot could then form the basis of a formal

policy recommendation, creating a scalable and sustainable pathway for the nationwide adoption of this technology.

5.3.1 Mitigating Overfitting

The overfitting observed in this study, particularly with the larger YOLOv12x model, points to a clear area for improvement. Future work should prioritize mitigating this issue through several key strategies:

a. **More Aggressive Data Augmentation:** While standard augmentations were used, more advanced techniques could be employed. This includes methods like *Copy-Paste* (to add more instances of rare classes like 'lateral_crack' onto diverse backgrounds) and *Test-Time Augmentation (TTA)*, where predictions are made on multiple augmented versions of an image and then averaged.

b. **Regularization Techniques:** Stronger regularization could be applied during training. This might involve increasing the *dropout* rate within the model's layers or applying a higher *weight decay*, which penalizes large weights in the network and discourages the model from "memorizing" the training data.

c. **Cross-Validation:** Instead of a single 80/20 split, a *k-fold cross-validation* (e.g., k=5) could be implemented. This would involve training and validating the model 5 times on different subsets of the data, providing a more robust estimate of its true performance and ensuring that the results are not an anomaly of one specific data split.

REFERENCE

Ali, M. L., & Zhang, Z. (2024). The YOLO Framework: A Comprehensive Review of Evolution, Applications, and Benchmarks in Object Detection. *Computers*, *13*(12), 336. https://doi.org/10.3390/computers13120336

Alif, M. A. R., & Hussain, M. (2025). *YOLOv12: A Breakdown of the Key Architectural Features* (Version 1). Version 1. arXiv. https://doi.org/10.48550/ARXIV.2502.14740

antaranews.com. (2024, February 3). Government readies Rp15 trillion for regional road repair in 2024. Retrieved October 6, 2025, from Antara News website: https://en.antaranews.com/news/304878/government-readies-rp15-trillion-for-regional-road-repair-in-2024

Asian Transport Observatory. (2025). Indonesia Road Safety Profile 2025. Retrieved October 6, 2025, from https://asiantransportobservatory.org/analytical-outputs/roadsafetyprofiles/indonesia-road-safety-profile-2025/

Badan Pusat Statistik. (2024). *Land Transportation Statistics 2023*. Retrieved from https://www.bps.go.id/en/publication/2024/11/25/cdcf9b5e74dd2e9bb3458ee4/land-transportation-statistics-2023.html

Bouazizi, O., Azroumahli, C., El Mourabit, A., & Oussouaddi, M. (2024). Road Object Detection using SSD-MobileNet Algorithm: Case Study for Real-Time ADAS Applications. *Journal of Robotics and Control (JRC)*, *5*(2), 551–560. https://doi.org/10.18196/jrc.v5i2.21145

Buhl, N. (n.d.). YOLO Object Detection Explained: A Beginner's Guide | Encord. Retrieved October 7, 2025, from Encord website: https://encord.com/blog/yolo-object-detection-guide/

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020, May 28). *End-to-End Object Detection with Transformers*. arXiv. https://doi.org/10.48550/arXiv.2005.12872

Dlužnevskij, D., Stefanovič, P., & Ramanauskaitė, S. (2021). Investigation of YOLOv5 Efficiency in iPhone Supported Systems. *Baltic Journal of Modern Computing*, *9*(3). https://doi.org/10.22364/bjmc.2021.9.3.07

Dunphy, K., Fekri, M. N., Grolinger, K., & Sadhu, A. (2022). Data Augmentation for Deep-Learning-Based Multiclass Structural Damage Detection Using Limited Information. *Sensors*, *22*(16), 6193. https://doi.org/10.3390/s22166193

Dupont, M. (2024). History of YOLO: From YOLOv1 to YOLOv10. Retrieved October 7, 2025, from Labelvisor website: https://www.labelvisor.com/history-of-yolo-from-yolov1-to-yolov10/

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, *88*(2), 303–338. https://doi.org/10.1007/s11263-009-0275-4

Fairuz Mat Radzi, S., Amiruddin Abd Rahman, M., & Luqman Arif Bin Mohamad, M. (2025). RT-DETR-Pothole: Lightweight Real-Time Detection Transformers for Improved Road Pothole Detection. *Journal of Physics: Conference Series*, *3022*(1), 012003. https://doi.org/10.1088/1742-6596/3022/1/012003

Han, T., Hou, S., Gao, C., Xu, S., Pang, J., Gu, H., & Huang, Y. (2025). EF-RT-DETR: A efficient focused real-time DETR model for pavement distress detection. *Journal of Real-Time Image Processing*, *22*(2), 63. https://doi.org/10.1007/s11554-025-01641-x

Humanitarian OpenStreetMap Team (HOT). (2024). *Indonesia Roads (OpenStreetMap Export)* [Data set]. Retrieved from https://data.humdata.org/dataset/hotosm_idn_roads

Investor Daily. (2024). Next year, national road length to increase by 104 km. Retrieved October 6, 2025, from PwC website: https://www.pwc.com/id/en/media-centre/infrastructure-news/september-2024/next-year--national-road-length-to-increase-by-104-km.html

Jegham, N., Koh, C. Y., Abdelatti, M., & Hendawi, A. (2025, March 17). *YOLO Evolution: A Comprehensive Benchmark and Architectural Review of YOLOv12, YOLO11, and Their Previous Versions*. arXiv. https://doi.org/10.48550/arXiv.2411.00201

Kantor. (2023). Road Damage Indonesia Computer Vision Model. Retrieved October 8, 2025, from Roboflow website: https://universe.roboflow.com/kantor-uskvs/road-damage-indonesia

Kementerian PPN/Bappenas. (2020). *The National Medium-Term Development Plan For 2020-2024* (p. 320). Retrieved from https://perpustakaan.bappenas.go.id/e-library/file_upload/koleksi/migrasi-data-publikasi/file/RP_RKP/Narasi-RPJMN-2020-2024-versi-Bahasa-Inggris.pdf

Kurohman, N. (2025, April 25). Indonesia's Road Safety Crisis: Issues and Policy Responses. Retrieved October 7, 2025, from ResearchGate website: https://www.researchgate.net/publication/391059076_Indonesia's_Road_Safety_Crisis_Issues_and_Policy_Responses

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., … Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Computer Vision – ECCV 2014* (pp. 740–755). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-10602-1_48

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). *SSD: Single Shot MultiBox Detector*. https://doi.org/10.1007/978-3-319-46448-0_2

Rachman, M. N. F. A., Rifai, A. I., Rijaluddin, A., & Prasetijo, J. (2025). The Analysis of Rural Road Distress with Indonesian Standard: A Case of Majalengka-West Java. *The 8th Mechanical Engineering, Science and Technology International Conference*, 58. MDPI. https://doi.org/10.3390/engproc2025084058

Ray, D., & Ing, L. Y. (2016). Addressing Indonesia's Infrastructure Deficit. *Bulletin of Indonesian Economic Studies*, *52*(1), 1–25. https://doi.org/10.1080/00074918.2016.1162266

Rothenberg, A., Gonzalez-Navarro, M., & Getler, P. J. (2022, November 24). The benefits of road maintenance: Lessons from Indonesia. Retrieved October 7, 2025, from CEPR website: https://cepr.org/voxeu/columns/benefits-road-maintenance-lessons-indonesia

Rubix Data Science. (2024). *RUBIX COUNTRY INSIGHTS*. Retrieved from https://rubixds.com/assets/uploads/attachment/a91d8-rubix-country-insights-indonesia-december-2024.pdf

Silva, L. A., Leithardt, V. R. Q., Batista, V. F. L., Villarrubia González, G., & De Paz Santana, J. F. (2023). Automated Road Damage Detection Using UAV Images and Deep Learning Techniques. *IEEE Access*, *11*, 62918–62931. https://doi.org/10.1109/ACCESS.2023.3287770

Tian, Y., Ye, Q., & Doermann, D. (2025, February 18). *YOLOv12: Attention-Centric Real-Time Object Detectors*. arXiv. https://doi.org/10.48550/arXiv.2502.12524

Viswanatha, v., Chandana, R. K., & Ramachandra, A. (2025). Real Time Object Detection System with YOLO and CNN Models: A Review. *ResearchGate*. https://doi.org/10.37896/JXAT14.07/315415

World Bank. (2012). Investing in Indonesia's Roads: Improving Efficiency and Closing the - World Bank Documents and Reports. *Road Sector Public Expenditure 2012*. Retrieved from

https://documents1.worldbank.org/curated/en/366991468269399430/pdf/733030WP0Indon00disclosed0100180120.pdf

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., & Dai, J. (2021, March 18). *Deformable DETR: Deformable Transformers for End-to-End Object Detection*. arXiv. https://doi.org/10.48550/arXiv.2010.04159

Zuo, C., Huang, N., Yuan, C., & Li, Y. (2025). Pavement-DETR: A High-Precision Real-Time Detection Transformer for Pavement Defect Detection. *Sensors*, *25*(8), 2426. https://doi.org/10.3390/s25082426