

UNIVERSITY OF ZAMBIA

SCHOOL OF NATURAL AND APPLIED SCIENCES

Department of Computing and Informatics

AI-POWERED INSURANCE CLAIMS AUTOMATION PLATFORM:

A CASE STUDY OF HOBBITON INVESTMENTS, ZAMBIA

A Dissertation Submitted in Partial Fulfilment of the Requirements

for the Degree of Bachelor of Science in Computer Science

By

Ben Samuel

Student ID: 2022055501

Ephetred Ndhlovu

Student ID: 2021463681

Supervisor: Dr. Lighton Phiri

Department of Computing and Informatics

lighton.phiri@unza.zm

Lusaka, Zambia

2026

DECLARATION

We declare that this dissertation is our own work and that all sources used or quoted have been indicated and acknowledged by means of complete references. This work has not previously been submitted in whole or in part for any degree or examination at this or any other university.

Signature: _____ Date: _____

Ben Samuel (2022055501)

Signature: _____ Date: _____

Ephetred Ndhlovu (2021463681)

Certified by:

Signature: _____ Date: _____

Dr. Lighton Phiri

Supervisor, Department of Computing and Informatics

ABSTRACT

The Zambian insurance industry processes vehicle accident claims through manual inspection procedures that are time-consuming, inconsistent, and susceptible to fraudulent manipulation. This dissertation presents the design, implementation, and evaluation of an AI-powered insurance claims automation platform developed in collaboration with Hobbiton Investments, a Zambian insurance technology company. The system employs a YOLOv8m convolutional neural network trained on 2,280 annotated vehicle damage photographs to automatically detect the presence, type, location, and severity of vehicle damage from client-submitted photographs and video frames. The trained model was further fine-tuned on an augmented dataset of 51,051 images to improve performance on visually variable damage classes. The platform integrates a REST API backend, a multi-stakeholder web portal providing role-specific interfaces for policyholders, insurers, and repair shop operators, an automated claims workflow engine, and a seven-check AI-assisted fraud detection module. The final system achieves an overall mean Average Precision at IoU threshold 0.50 (mAP@50) of 0.635 with a precision of 0.824 and recall of 0.577, with the structurally significant damage classes of Broken Part, Cracked, and Dent achieving individual mAP@50 values of 0.876, 0.940, and 0.716 respectively. The complete system is deployed online and accessible from any internet-connected device, reducing damage assessment time from a typical fourteen to forty-five day manual process to under two seconds. The work demonstrates the feasibility of deploying computer vision-based claims automation within the resource constraints of the Zambian insurance market.

Keywords: Computer Vision, YOLOv8, Transfer Learning, Vehicle Damage Detection, Insurance Claims Automation, Fraud Detection, Zambia

ACKNOWLEDGEMENTS

We would like to express our sincere gratitude to all those who contributed to the successful completion of this project.

First and foremost, we thank our supervisor, Dr. Lighton Phiri, for his invaluable guidance, technical insight, and patience throughout the research and development process. His feedback was instrumental in shaping both the technical direction and academic rigour of this work.

We extend our appreciation to the management and technical team at Hobbiton Investments, Zambia, for providing the industry context, operational requirements, and collaborative support that grounded this project in real-world practice.

We acknowledge the Department of Computing and Informatics at the University of Zambia for providing the academic environment and resources necessary to carry out this work.

Finally, we extend our gratitude to our families and friends for their constant encouragement and support throughout the duration of this project.

TABLE OF CONTENTS

Declaration.....	i
Abstract.....	ii
Acknowledgements.....	iii
Table of Contents.....	iv
List of Figures.....	v
List of Tables.....	vi
List of Abbreviations.....	vii
CHAPTER ONE: Introduction.....	1
1.1 Background.....	1
1.2 Problem Statement.....	2
1.3 Aim and Objectives.....	3
1.4 Scope and Delimitations.....	3
1.5 Significance of the Study.....	4
1.6 Dissertation Structure.....	4
1.7 Definition of Terms.....	5
CHAPTER TWO: Literature Review.....	6
2.1 Introduction.....	6
2.2 Deep Learning and CNNs.....	6
2.3 Object Detection Architectures.....	7
2.4 Transfer Learning.....	8
2.5 Vehicle Damage Detection.....	9
2.6 Fraud Detection in Insurance.....	11
2.7 Automated Claims Processing.....	12
2.8 Summary and Research Gap.....	13
CHAPTER THREE: Methodology.....	14
3.1 Introduction.....	14
3.2 Research Design.....	14

3.3 Dataset Acquisition and Preparation.....	15
3.4 Model Architecture and Training.....	17
3.5 System Architecture.....	19
3.6 Fraud Detection Module.....	22
3.7 Claims Workflow.....	23
3.8 PDF Report Generation.....	24
3.9 API Endpoints.....	24
3.10 Deployment Architecture.....	24
CHAPTER FOUR: Results and Evaluation.....	26
4.1 Model Performance.....	26
4.2 Version Comparison.....	27
4.3 Sample Detection Output.....	28
4.4 Car Parts Segmentation Model Performance.....	28
4.5 Fraud Detection Results.....	29
4.6 System Performance.....	29
4.7 Portal Usability.....	29
CHAPTER FIVE: Discussion.....	31
5.1 Interpretation of Results.....	31
5.2 Limitations.....	33
5.3 Future Work.....	34
CHAPTER SIX: Conclusion.....	36
References.....	37

LIST OF FIGURES

Figure 1: System Architecture Overview

Figure 2: YOLOv8m Architecture Diagram

Figure 3: Dataset Class Distribution

Figure 4: Data Augmentation Pipeline

Figure 5: Training Loss Curves, Version 2

Figure 6: Confusion Matrix, Version 2

Figure 7: Per-class mAP@50 Comparison, v2 vs v3

Figure 8: Annotated Image Sample, Damage Detection Output

Figure 9: Location Classifier Zone Map

Figure 10: Fraud Detection Pipeline Flowchart

Figure 11: Claims Workflow State Diagram

Figure 12: Web Portal, Policyholder View

Figure 13: Web Portal, Insurer Dashboard

Figure 14: Web Portal, Repair Shop View

Figure 15: Sample Generated PDF Report

Figure 16: Car Parts Segmentation Model Results

LIST OF TABLES

Table 1: Definition of Terms

Table 2: Summary of Literature and Research Gaps

Table 3: Training Hyperparameters, Version 2

Table 4: Per-class Performance, Version 2

Table 5: Per-class Performance, Version 3

Table 6: Version 2 vs Version 3 Comparison

Table 7: Fraud Detection Checks

Table 8: API Endpoints

Table 9: System Deployment Summary

Table 10: Car Parts Segmentation Model Performance

LIST OF ABBREVIATIONS

<i>Abbreviation</i>	<i>Full Form</i>
AI	Artificial Intelligence
API	Application Programming Interface
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CRISP-DM	Cross-Industry Standard Process for Data Mining
DL	Deep Learning
DSR	Design Science Research
ELA	Error Level Analysis
EXIF	Exchangeable Image File Format
GPU	Graphics Processing Unit
HTTP	Hypertext Transfer Protocol
IoU	Intersection over Union
JSON	JavaScript Object Notation
mAP	Mean Average Precision
ML	Machine Learning
PIL	Python Imaging Library
REST	Representational State Transfer
URL	Uniform Resource Locator
YOLO	You Only Look Once
UNZA	University of Zambia

CHAPTER ONE: INTRODUCTION

1.1 Background

The insurance industry plays a critical role in the economic stability of nations by providing financial protection to individuals and organisations against unforeseen losses. In Zambia, the insurance sector has experienced gradual growth, yet claims processing remains one of its most persistent operational challenges. Vehicle insurance claims, in particular, require physical inspection of damaged vehicles by trained assessors, a process that is labour-intensive, time-consuming, and susceptible to human error and fraudulent manipulation.

Zambia recorded 36,755 road accidents in 2023, representing a significant volume of potential insurance claims across the country. The economic cost of road traffic accidents in Zambia is estimated at approximately \$910 million annually, equivalent to 4.7 percent of the country's Gross Domestic Product. Against this backdrop, insurance companies face mounting pressure to process claims efficiently while simultaneously safeguarding against fraudulent submissions.

Traditional claims processing workflows require a policyholder to visit a branch office, submit documentation, wait for a field assessor to travel to the vehicle location, and then await a written assessment report before any settlement decision is made. This process routinely takes between fourteen and forty-five days from submission to settlement, a duration that creates significant dissatisfaction among policyholders and contributes to low insurance penetration rates in Zambia.

Recent advances in Artificial Intelligence, particularly in the domain of Computer Vision and Deep Learning, have demonstrated significant potential for automating complex visual inspection tasks. Object detection models, most notably the YOLO (You Only Look Once) family of architectures, have achieved near-human accuracy in identifying and localising objects within images. These

developments present a compelling opportunity to apply AI-driven visual inspection to vehicle damage assessment.

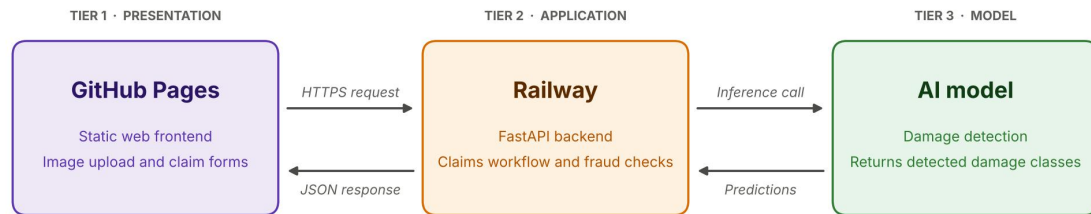


Figure 1. High-level system architecture of the AI-powered claims platform.

1.2 Problem Statement

The manual vehicle damage assessment process employed by Zambian insurance companies is characterised by significant inefficiencies that negatively impact all stakeholders in the claims value chain. Policyholders experience prolonged waiting periods and are required to be physically present throughout multiple stages of the process. Insurance companies incur high operational costs associated with field assessors and are exposed to fraudulent claims that are difficult to detect through manual review alone. Repair shops receive delayed job assignments that affect their operational planning and cash flow.

Furthermore, the subjective nature of manual assessment introduces inconsistency in damage valuation, with different assessors potentially arriving at divergent conclusions for identical damage scenarios. This inconsistency creates disputes, delays settlements further, and undermines confidence in the fairness of the claims process.

No comprehensive AI-powered solution exists within the Zambian insurance market that addresses these challenges in an integrated manner by combining automated damage detection, fraud analysis, claims workflow automation, and multi-stakeholder access through a single platform.

1.3 Aim and Objectives

The aim of this project is to design and implement an AI-powered insurance claims automation platform that reduces the time and cost of vehicle damage assessment for Hobbiton Investments while improving fraud detection capability and stakeholder experience.

The specific objectives are as follows:

- To design, train, and validate a deep learning model leveraging Convolutional Neural Networks and transfer learning capable of automatically detecting the presence, type, location, and severity of vehicle damage from client-submitted photographs and video frames, achieving a target classification accuracy of at least 75 percent mAP@50 on a held-out test dataset.
- To build a secure, responsive web-based platform that provides distinct, role-specific interfaces for three primary user types: policyholders submitting claims and tracking progress, insurance claim handlers reviewing AI-generated reports and authorising decisions, and repair shop operators receiving job assignments and submitting repair cost estimates.
- To design and implement an end-to-end automated claims workflow that, upon receipt of a client submission, triggers AI-based damage analysis, generates a structured damage assessment report, routes the claim to the appropriate insurer handler, and notifies the relevant repair shop without requiring manual intervention at any stage of the pipeline.
- To develop and embed an AI-assisted fraud detection layer that analyses submitted images and videos for anomalies including image metadata inconsistencies, visual indicators of prior damage, signs of image manipulation or recycling, and damage patterns inconsistent with the reported accident type.

1.4 Scope and Delimitations

This project focuses on vehicle damage assessment for motor vehicle insurance claims within the Zambian context. The system accepts photographic and video submissions of damaged vehicles and produces structured damage assessment reports. The scope includes the development of the AI detection model, the REST API, the multi-stakeholder web portal, the fraud detection module, and the automated claims workflow.

The project does not include integration with existing insurance management systems or financial settlement platforms. The system does not perform vehicle identification or registration verification. Authentication and role-based access control are implemented at the prototype level suitable for demonstration purposes rather than at the production security standard required for live commercial deployment.

1.5 Significance of the Study

This project makes contributions at the intersection of artificial intelligence, software engineering, and the Zambian insurance industry. From a technical perspective, it demonstrates the feasibility of deploying a computer vision-based damage assessment system on CPU-grade cloud infrastructure at costs accessible to small and medium-sized insurance companies in developing markets.

From an industry perspective, the system provides Hobbition Investments with a working prototype that reduces assessment time from an average of fourteen days to under two seconds. From an academic perspective, the project contributes to the body of knowledge on the application of transfer learning and object detection in African insurance contexts, where training data availability is limited and infrastructure constraints require careful architectural decisions.

1.6 Dissertation Structure

This dissertation is structured as follows. Chapter Two presents a review of the relevant literature. Chapter Three describes the research methodology, system architecture, and implementation. Chapter Four presents the results and evaluation. Chapter Five discusses the findings, limitations, and recommendations for future work. Chapter Six provides the conclusion.

1.7 Definition of Terms

Term	Definition
Convolutional Neural Network (CNN)	A class of deep learning model designed to process grid-structured data such as images by applying learned filters to detect spatial patterns at multiple scales.
Transfer Learning	The practice of initialising a model with weights learned on a large general dataset and fine-tuning it on a smaller domain-specific dataset.
Mean Average Precision (mAP@50)	A standard metric for object detection models measuring average precision across all classes at an IoU threshold of 0.50.
Object Detection	A computer vision task that simultaneously classifies objects in an image and localises them with bounding boxes.
REST API	Representational State Transfer Application Programming Interface, a standard architectural style for building web services over HTTP.
YOLOv8	You Only Look Once version 8, a real-time object detection framework that performs detection in a single neural network forward pass.
EXIF Metadata	Exchangeable Image File Format metadata, information

	embedded in image files by cameras and smartphones including timestamp, GPS coordinates, and device model.
Error Level Analysis (ELA)	A forensic image analysis technique that detects regions of an image that have been digitally edited by analysing compression artefact levels.

CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction

This chapter reviews the existing body of literature relevant to the design and implementation of an AI-powered vehicle damage assessment system. The review covers five thematic areas: the evolution of deep learning and convolutional neural networks; object detection architectures with particular attention to the YOLO family; transfer learning as a strategy for domain adaptation with limited training data; existing research on automated vehicle damage detection; and AI-assisted fraud detection in insurance contexts. The chapter concludes by identifying the gaps in existing literature that this project addresses.

2.2 Deep Learning and Convolutional Neural Networks

2.2.1 Foundations of Convolutional Neural Networks

Deep learning has fundamentally transformed the field of computer vision. The introduction of deep Convolutional Neural Networks (CNNs) by Krizhevsky et al. [4] in 2012, demonstrating significant accuracy improvement on the ImageNet challenge, is widely regarded as the watershed moment that triggered the modern deep learning era. CNNs achieve their effectiveness through a hierarchical feature extraction process in which successive convolutional layers learn increasingly abstract representations, from edges and textures in early layers to complex object-level features in deeper layers [5].

The convolutional operation applies learned filters across spatial dimensions of the input, producing feature maps that capture the presence and location of specific visual patterns. Subsequent architectural innovations including VGGNet, ResNet [6], and EfficientNet progressively improved accuracy and training efficiency. ResNet's introduction of residual connections addressed the vanishing

gradient problem, enabling architectures with hundreds of layers to be trained effectively.

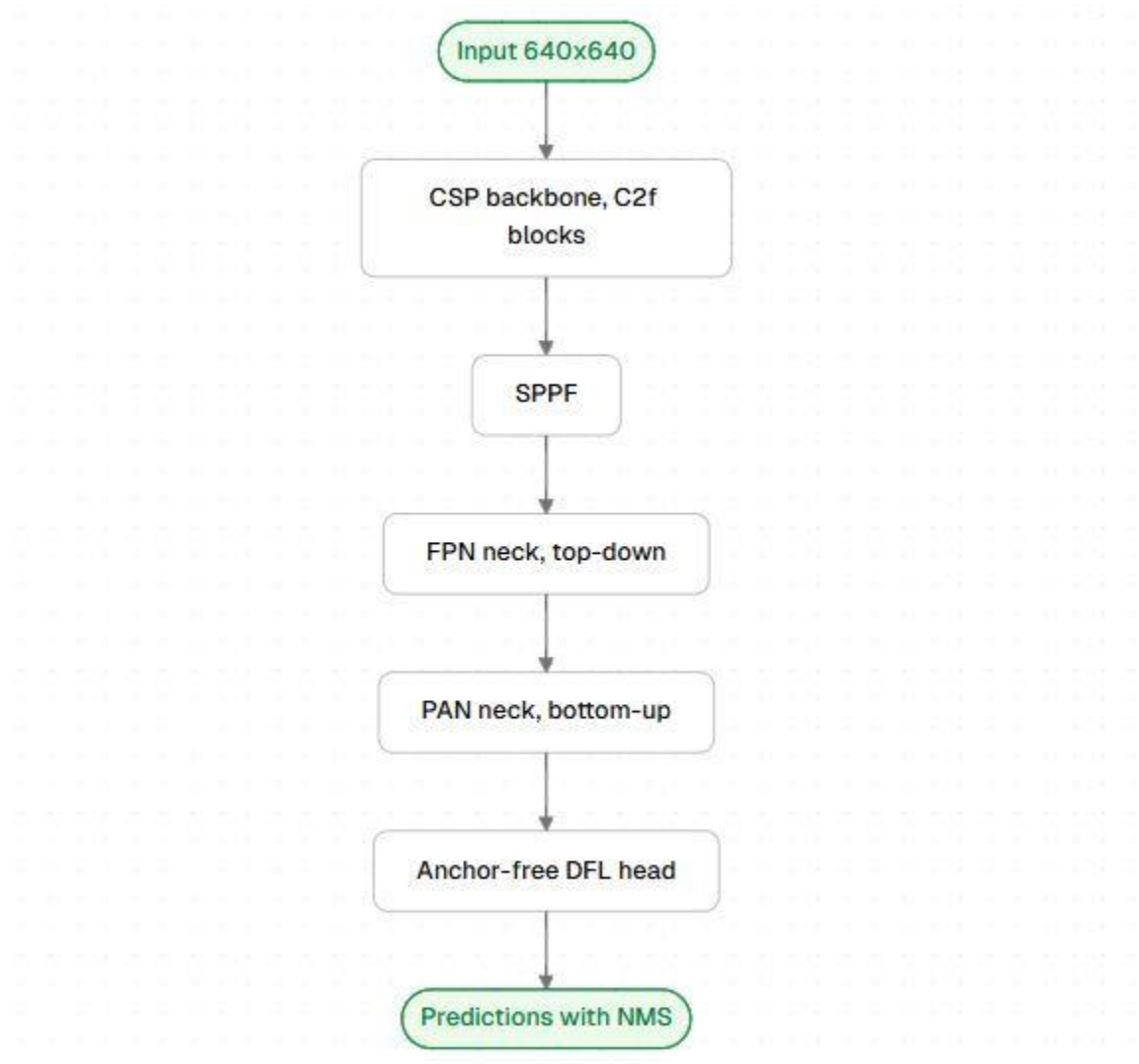


Figure 2. YOLOv8m architecture showing backbone, neck, and detection head components.

2.2.2 Object Detection Architectures

Object detection extends image classification by simultaneously predicting the class and spatial location of multiple objects within a single image. Early approaches such as R-CNN and its successors Fast R-CNN and Faster R-CNN adopted a two-stage pipeline: a region proposal network first identifies candidate object locations, which are then classified by a separate network. While accurate, these architectures are computationally expensive.

Redmon et al. [7] introduced YOLO as a fundamentally different approach, reformulating object detection as a single regression problem solved in one forward pass. YOLOv8, developed by Ultralytics [8], further improved the architecture with a decoupled detection head separating classification and localisation tasks. Comparative evaluations consistently place YOLOv8 among the highest-performing real-time object detectors on standard benchmarks.

2.3 Transfer Learning for Domain Adaptation

A central challenge in applying deep learning to specialised domains is the requirement for large labelled datasets. Transfer learning addresses this challenge by leveraging knowledge acquired during training on large general-purpose datasets [9]. For computer vision applications, the most widely used approach is fine-tuning a model pre-trained on ImageNet, which already encodes general visual representations, on a smaller target dataset.

Yosinski et al. [10] demonstrated experimentally that the transferability of features varies by layer depth, with early layers encoding general features and later layers encoding domain-specific representations. This finding has informed the common practice of fine-tuning all layers when adapting to a new domain, which was adopted in the present study.

2.4 Vehicle Damage Detection in the Literature

2.4.1 Early Approaches

Early research on automated vehicle damage assessment relied on traditional computer vision techniques including edge detection and texture feature extraction. The limitations of hand-crafted feature approaches became increasingly apparent as researchers attempted to scale to larger and more diverse datasets. The high variability in the visual appearance of vehicle damage,

across vehicle colours, models, viewing angles, and lighting conditions, proved particularly challenging for feature engineering approaches.

2.4.2 Deep Learning Approaches

The application of CNNs to vehicle damage detection has produced substantially improved results. Dwivedi et al. [11] demonstrated the effectiveness of ResNet-50 for binary damage classification, achieving 89 percent accuracy. Ding et al. [12] applied YOLOv5 to an 8-class vehicle damage dataset, reporting mAP@50 values of 0.71, providing a relevant benchmark for the present study. Their work confirmed that single-stage detection architectures are both sufficiently accurate and fast enough for near-real-time deployment in insurance applications.

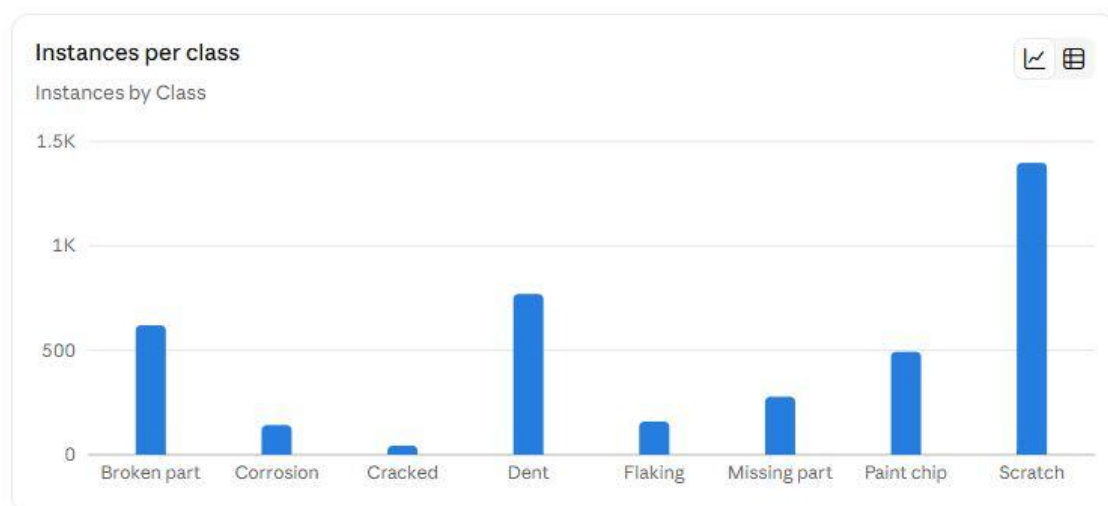


Figure 3. Distribution of damage class instances in the original training dataset.

2.4.3 Severity and Location Classification

The classification of damage severity represents a relatively underexplored aspect of the literature. Merkel et al. proposed a two-stage pipeline for severity estimation, reporting 78 percent accuracy on a three-level scale. Spatial localisation of damage to specific vehicle components has been addressed by Yamaguchi et al. using semantic segmentation, achieving pixel-level classification across 15 vehicle components. The geometric zone-based classification approach

adopted in the present study offers a pragmatic alternative that provides meaningful location information without requiring component-level annotations.

2.5 Fraud Detection in Insurance

Insurance fraud represents between 10 and 15 percent of total claim costs in mature markets. Traditional fraud detection relies on rule-based expert systems supplemented by manual investigation [13]. Image-based fraud detection through metadata analysis provides a non-invasive mechanism for detecting temporal inconsistencies between claimed accident dates and photographic timestamps [14]. Error Level Analysis (ELA), developed by Krawetz [15], has been applied to insurance fraud detection to identify digitally manipulated images.

2.6 Automated Claims Processing Systems

The automation of insurance claims processing has been the subject of increasing research interest. Bhattacharya et al. [16] reviewed AI applications in insurance and identified claims automation as one of the highest-value opportunities, with potential cost reductions of 30 to 40 percent. End-to-end claims automation systems have been deployed by several major insurers in developed markets; however, the literature reveals a significant gap in the application of such systems in Sub-Saharan African insurance markets.

2.7 Summary of Literature Review and Research Gap

Theme	Key Finding	Gap Addressed by This Project
CNN Object Detection	YOLOv8 achieves state-of-the-art real-time detection accuracy [4,7,8]	Applied to vehicle damage in a Zambian insurance context
Transfer Learning	ImageNet pre-training consistently improves domain adaptation [9,10]	Used to achieve results with only 2,280 training images

Vehicle Damage Detection	YOLOv5 achieves 0.71 mAP on 8-class damage dataset [12]	Extended with location, severity, and video frame analysis
Fraud Detection	ELA and EXIF analysis detect manipulation and temporal inconsistencies [14,15]	Integrated into a 7-check automated fraud pipeline
Claims Automation	30-40% cost reduction potential identified [16]	First integrated end-to-end system for Zambian market

CHAPTER THREE: METHODOLOGY

3.1 Introduction

This chapter describes the research methodology, system architecture, and implementation approach adopted in the development of the AI-powered insurance claims automation platform. The chapter covers the overall research design, dataset acquisition and preparation, model training and evaluation methodology, system architecture design, and the implementation of individual system components.

3.2 Research Design

This project adopts a Design Science Research (DSR) methodology as defined by Hevner et al. [17], which is appropriate for information systems research that produces a technological artefact as its primary output. DSR is characterised by an iterative build-evaluate cycle in which the artefact is designed, implemented, and evaluated against predefined objectives.

The development process was guided by an Agile-inspired iterative approach drawing on Scrum principles, with work structured into incremental development cycles each producing a functional system component. The CRISP-DM framework was applied to structure the AI model development pipeline, encompassing the phases of business understanding, data understanding, data preparation, modelling, evaluation, and deployment.

3.3 Dataset Acquisition and Preparation

3.3.1 Primary Dataset

The primary training dataset was sourced from Roboflow Universe, a publicly accessible repository of annotated computer vision datasets. The selected dataset comprises 2,280 photographs of damaged vehicles annotated with bounding

boxes across eight damage classes: Broken Part, Corrosion, Cracked, Dent, Flaking, Missing Part, Paint Chip, and Scratch. The dataset was provided in YOLOv8 format and split into training (70%), validation (15%), and test (15%) subsets, yielding 1,596 training images, 342 validation images, and 342 test images.

3.3.2 Class Distribution Analysis

Analysis of the class distribution revealed significant imbalance across the eight damage classes. Structural damage classes were substantially over-represented relative to surface damage classes, with Paint Chip having only 582 annotated instances and Cracked having only 39 instances across the entire dataset. This imbalance was anticipated to negatively affect model performance on minority classes.

3.3.3 Data Augmentation

To address class imbalance and improve model generalisation, a targeted data augmentation strategy was implemented. Images containing instances of the two weakest-performing classes, Scratch and Paint Chip, were identified programmatically and subjected to seven augmentation transformations: horizontal flip, brightness increase, brightness decrease, rotation by +15 degrees, rotation by -15 degrees, Gaussian blur, and contrast enhancement.

For horizontal flip augmentation, bounding box coordinates were transformed by inverting the normalised x-coordinate of the box centre ($\text{new_cx} = 1.0 - \text{cx}$). Following augmentation, the training dataset was expanded from 1,596 to 51,051 images. Additionally, a supplementary scratch detection dataset of 1,119 images was merged, with class indices remapped to align with the primary dataset's eight-class taxonomy.

Figure 4: Data Augmentation Pipeline

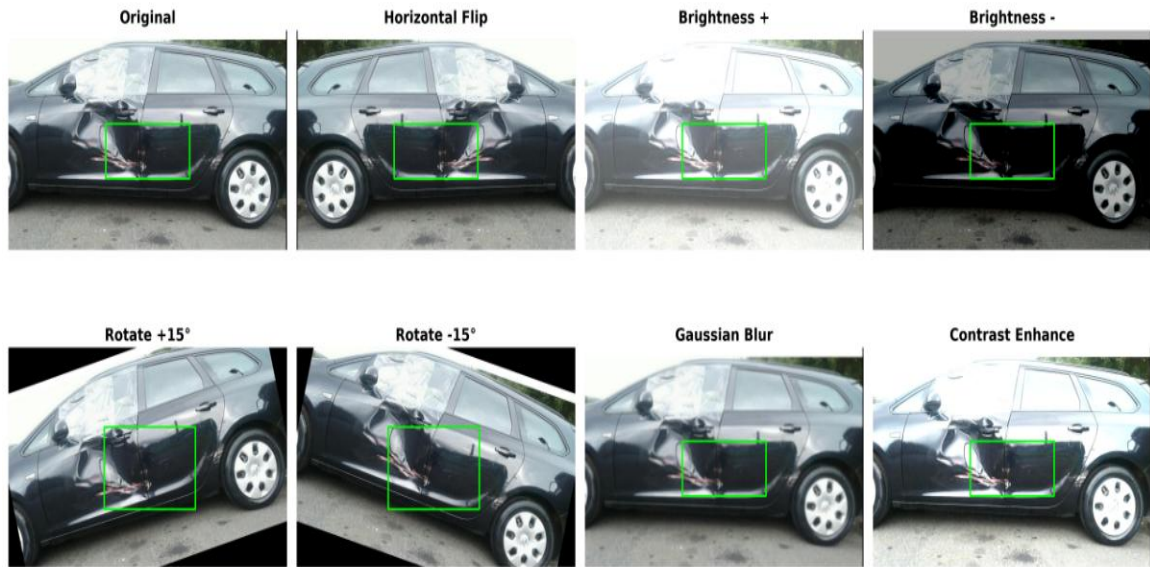


Figure 4. The seven augmentation transformations applied to minority class images.

3.4 Model Architecture and Training

3.4.1 Model Selection

YOLOv8m (medium) was selected as the base architecture following a comparative evaluation of the YOLOv8 model family. The small variant was evaluated in a preliminary experiment and achieved an overall mAP@50 of 0.548 after 50 epochs, below the performance threshold required for the application. The medium variant, with 25.9 million parameters and 79.1 GFLOPs, provided a favourable trade-off between accuracy and inference speed on the CPU-grade deployment infrastructure available to Hobbiton Investments. The model was initialised with weights pre-trained on the ImageNet dataset, implementing the transfer learning strategy described in Chapter Two.

3.4.2 Training Configuration, Version 2

<i>Hyperparameter</i>	<i>Value</i>	<i>Rationale</i>
Epochs	100	Sufficient for convergence on 1,596 training images
Image size	640×640px	Standard YOLOv8 input resolution
Batch size	16	Maximum feasible on Tesla T4 GPU (15GB VRAM)
Optimiser	Auto	YOLOv8 adaptive selection
Classification loss (cls)	2.0	Upweights classification to address imbalance
Box regression loss (box)	9.0	Upweights localisation accuracy
Horizontal flip	0.5	Standard augmentation for vehicle damage images
HSV saturation	0.9	Robustness to varied vehicle colours
HSV value	0.6	Robustness to varied lighting conditions
Scale jitter	0.7	Robustness to varied photography distances
Rotation degrees	15.0	Robustness to non-horizontal orientations
Close mosaic	40	Mosaic augmentation disabled for last 40 epochs

3.4.3 Training Configuration, Version 3

The Version 3 model was fine-tuned for 20 epochs starting from the Version 2 best weights, using a sampled subset of 8,000 images from the expanded dataset of 51,051. The same hyperparameter configuration was applied with the classification loss weight increased to 3.0 to further emphasise minority class performance. Training was conducted on Google Colaboratory using an NVIDIA Tesla T4 GPU and completed in approximately 33 minutes.

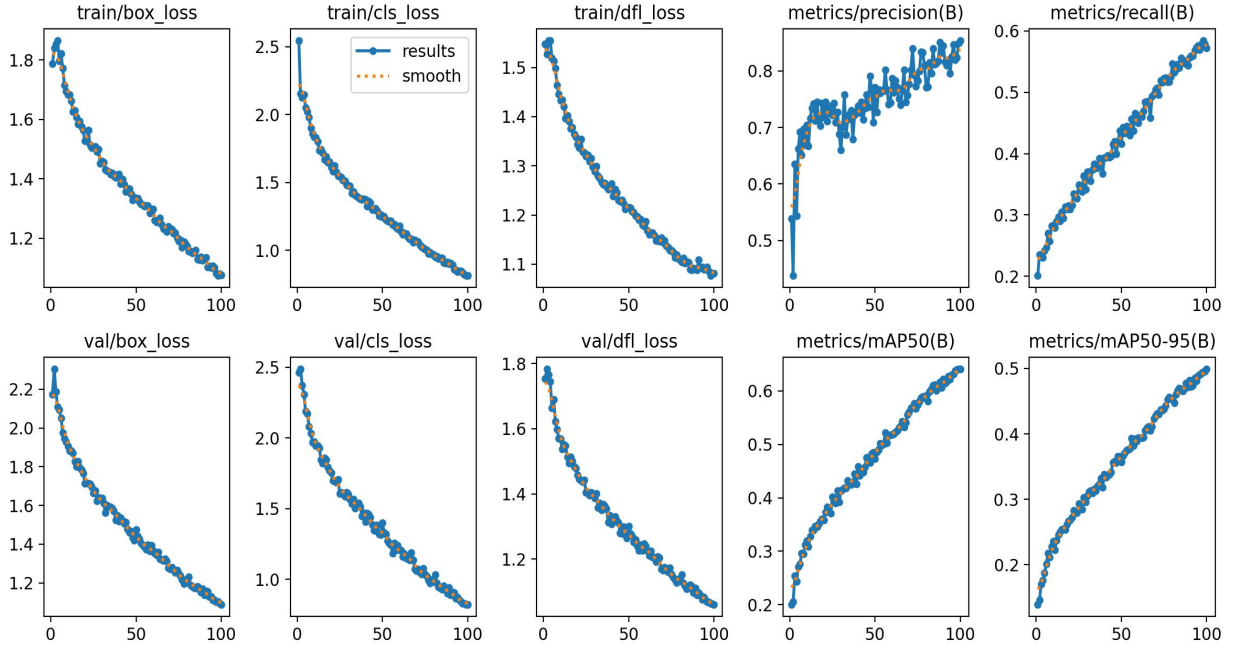


Figure 5. Training and validation loss curves for Version 2 model over 100 epochs.

3.5 System Architecture

3.5.1 Overall Architecture

The system follows a three-tier client-server architecture comprising a web-based frontend layer, a REST API middleware layer, and the AI inference engine. This architecture separates concerns between user interface presentation, business logic and API management, and AI model inference, enabling each layer to be developed, tested, and scaled independently.

The frontend is implemented as a single-page web application hosted on GitHub Pages. The API middleware is implemented using the FastAPI Python framework and deployed as a Docker container on Railway cloud infrastructure. The AI inference engine, comprising the trained YOLOv8 model and associated post-processing logic, runs within the API server process.

3.5.2 Inference Pipeline

When an image arrives at the API, it is validated, decoded using the Python Imaging Library (PIL), and resized to a maximum of 1,280 pixels. The preprocessed

image is passed to the YOLOv8 model with a confidence threshold of 0.25. Results are processed to extract the class name, compute severity using the rule-based severity classifier, and determine the car body zone using the geometric location classifier. Coloured bounding boxes are drawn on a copy of the input image and encoded as a base64 JPEG string for inclusion in the JSON response. The complete pipeline executes in under two seconds on CPU hardware.

3.5.3 Location Classifier Design

The location classifier maps each detected damage instance to one of fifteen car body zones using a geometric algorithm applied to the normalised bounding box coordinates returned by YOLOv8. The centre point of each bounding box is calculated and normalised to the range [0, 1] relative to the image dimensions. The image is partitioned into spatial zones: vertical zones covering the top 30 percent (Hood, Roof, Windscreen), middle 35 percent (Doors, Fenders), and bottom 30 percent (Bumpers, Sills); and horizontal zones covering left 25 percent, centre 50 percent, and right 25 percent. Detections near zone boundaries are classified deterministically without confidence scoring, which is noted as a limitation.

An enhanced location model was also developed using the YOLOv8s-seg segmentation architecture trained on a car parts dataset. This model achieves an overall mask mAP@50 of 0.739, with front-facing components such as hood (0.968), front glass (0.974), and front bumper (0.953) performing particularly well. The trained model is integrated as an optional replacement for the geometric classifier, with fallback to the geometric approach when the model is unavailable.

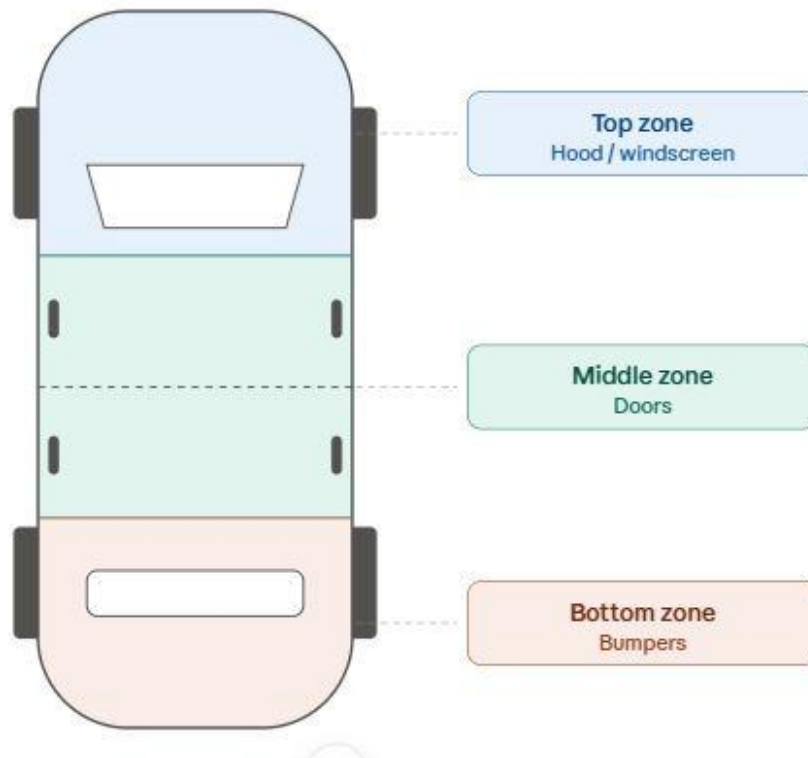


Figure 9. Geometric zone map used by the location classifier showing the 15 car body zones.

3.5.4 Severity Classifier Design

The severity classifier is a deterministic rule-based mapping that assigns a severity level to each detected damage class based on domain knowledge provided by Hobbition Investments. Broken Part, Missing Part, and Cracked are assigned Severe, reflecting their potential to compromise vehicle structural integrity. Corrosion, Dent, and Flaking are assigned Moderate. Scratch and Paint Chip are assigned Minor. The mapping was validated with Hobbition Investments' claims handling team prior to implementation.

3.6 Fraud Detection Module

The fraud detection module implements seven automated checks executed sequentially on every submitted image:

Check	Method	Flag Condition
-------	--------	----------------

1. Duplicate detection	MD5 cryptographic hash of image bytes	Hash matches a previously submitted image
2. Resolution check	PIL image dimensions	Width or height below 300 pixels
3. EXIF metadata presence	PIL EXIF extraction	No EXIF metadata present in image
4. Timestamp analysis	DateTime parsing of EXIF DateTimeOriginal	Image older than 30 days or dated in future
5. GPS validation	EXIF GPSInfo DMS-to-decimal conversion	GPS coordinates equal to (0.0, 0.0)
6. ELA manipulation detection	JPEG re-compression and pixel difference	Mean brightness of enhanced difference > 20
7. Damage pattern analysis	Set intersection of detected classes	All detected classes are pre-existing types

The fraud risk level is determined by the count of flags raised: zero flags produces Low risk; one or two flags produces Medium risk; three or more flags produces High risk. All fraud flags are advisory, no claim is automatically rejected. This design decision preserves human accountability and protects against false positive rejection of legitimate claims.

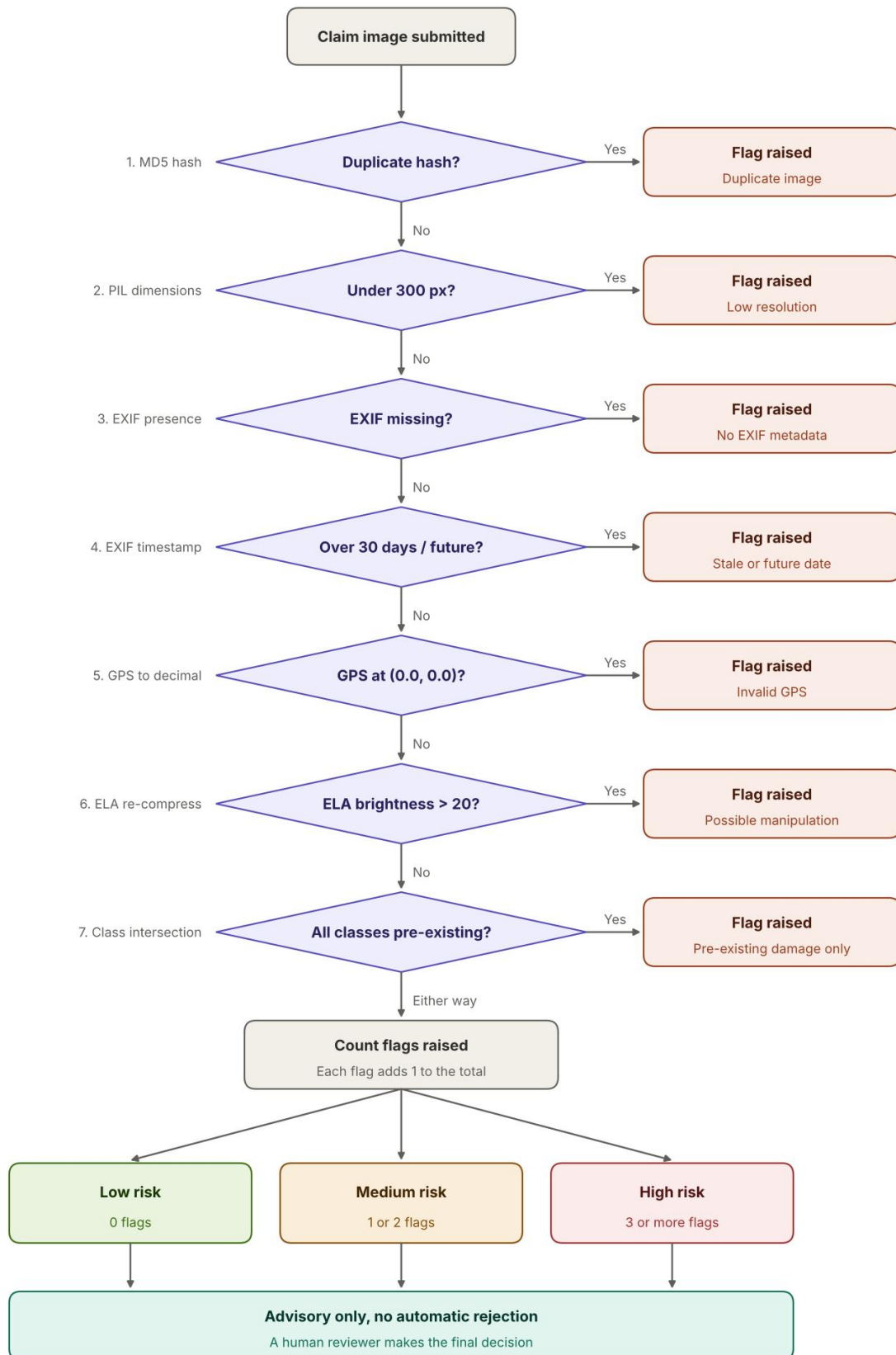


Figure 10. Fraud detection pipeline showing all seven automated checks.

3.7 Claims Workflow Implementation

The automated claims workflow is implemented as a stateful pipeline within the FastAPI application. Claims are stored in an in-memory Python dictionary indexed by unique claim identifiers. The claim lifecycle follows seven discrete states: SUBMITTED, PENDING_REVIEW, APPROVED, REJECTED, REPAIR_ASSIGNED, REPAIR_QUOTED, and SETTLED. State transitions are triggered by specific API endpoint calls, with each transition appending an entry to the claim's history log.

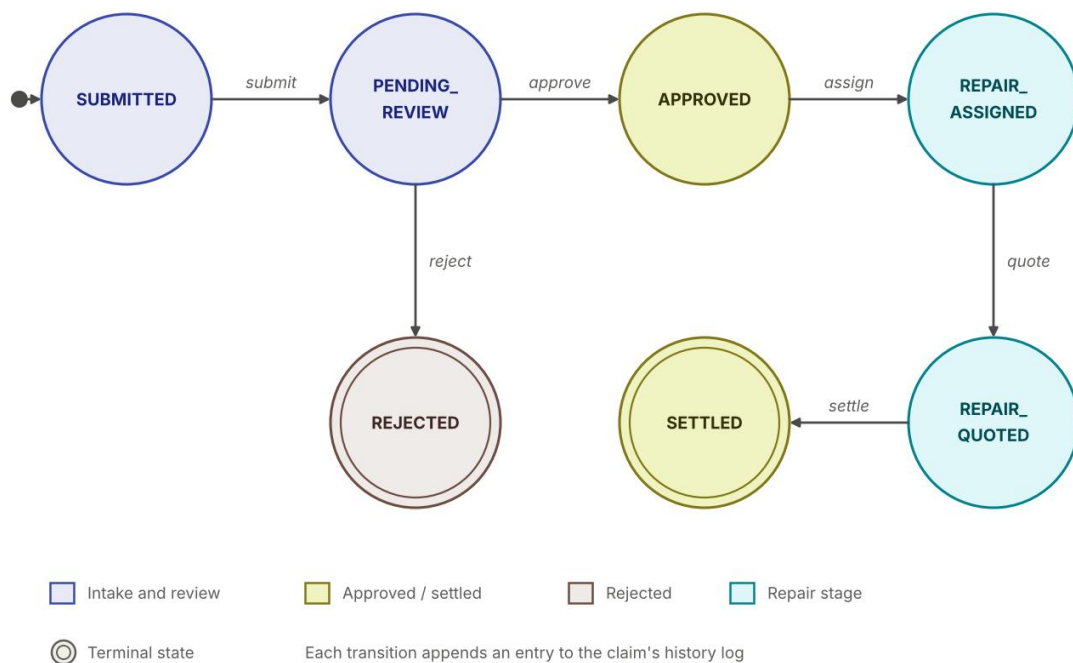


Figure 11. State machine diagram of the automated claims workflow lifecycle.

3.8 PDF Report Generation

Professional PDF damage assessment reports are generated using the ReportLab Python library. Each report includes a formatted header, the annotated damage image decoded from its base64 representation, a per-detection damage table, the fraud analysis result with individual flag descriptions, and a final

recommendation. The report is returned as a streaming HTTP response triggering automatic download in the browser.

3.9 API Endpoints

Endpoint	Method	Role	Description
/inspect	POST	Any	Image → JSON damage report
/inspect/annotated	POST	Any	Image → JSON + annotated image
/inspect/video	POST	Any	Video → frame analysis report
/inspect/report	POST	Any	Image → downloadable PDF
/claims/submit	POST	Policyholder	Submit claim with AI analysis
/claims	GET	Insurer	Retrieve all claims
/claims/{id}	GET	Any	Single claim details
/claims/{id}/approve	POST	Insurer	Approve claim
/claims/{id}/reject	POST	Insurer	Reject claim with reason
/claims/repair-shop/jobs	GET	Repair Shop	Retrieve assigned jobs
/claims/{id}/estimate	POST	Repair Shop	Submit repair cost estimate
/claims/{id}/status	GET	Policyholder	Track claim status

3.10 Deployment Architecture

The FastAPI backend is deployed as a Docker container on Railway cloud infrastructure using a custom Dockerfile based on the python:3.11-slim base image. The Dockerfile installs required system libraries for OpenCV headless operation. The trained model file is downloaded from Google Drive on container startup using the gdown library, avoiding the need to commit large binary files to the GitHub repository. The web portal is hosted on GitHub Pages as a static single-page application with continuous deployment triggered on every GitHub commit.

Component	Platform	URL	Technology
AI Inspection API	Railway	hobbiton-api-production-99d3.up.railway.app	FastAPI + Docker
Web Portal	GitHub Pages	breecode-99.github.io/hobbiton-portal	HTML/CSS/JavaScript
AI Model v3	Google Drive	Drive File ID: 1Ee0I5AR1cnwNJ2_ypFzdLcXcbAtOBX3I	YOLOv8m weights
Car Parts Model	Google Drive	Drive File ID: 1B_KtArSbcXQrc2CgQrfZVxpHWi-BacQG	YOLOv8s-seg weights
API Source Code	GitHub	github.com/breeCode-99/hobbiton-api	Python
Portal Source Code	GitHub	github.com/breeCode-99/hobbiton-portal	HTML/JavaScript

CHAPTER FOUR: RESULTS AND EVALUATION

4.1 Model Performance, Version 2

The Version 2 model was trained on 1,596 images for 100 epochs. Table 4 presents the per-class performance on the held-out test set of 341 images.

<i>Damage Class</i>	<i>Severity</i>	<i>mAP@50</i>	<i>Precision</i>	<i>Recall</i>	<i>Assessment</i>
Broken part	Severe	0.876	0.872	0.816	Above target
Missing part	Severe	0.814	0.879	0.759	Above target
Cracked	Severe	0.940	0.887	0.907	Excellent
Dent	Moderate	0.716	0.861	0.634	Good
Flaking	Moderate	0.392	0.772	0.340	Needs improvement
Corrosion	Moderate	0.543	0.827	0.472	Acceptable
Scratch	Minor	0.418	0.780	0.353	Needs improvement
Paint chip	Minor	0.382	0.713	0.333	Needs improvement
Overall	N/A	0.635	0.824	0.577	N/A

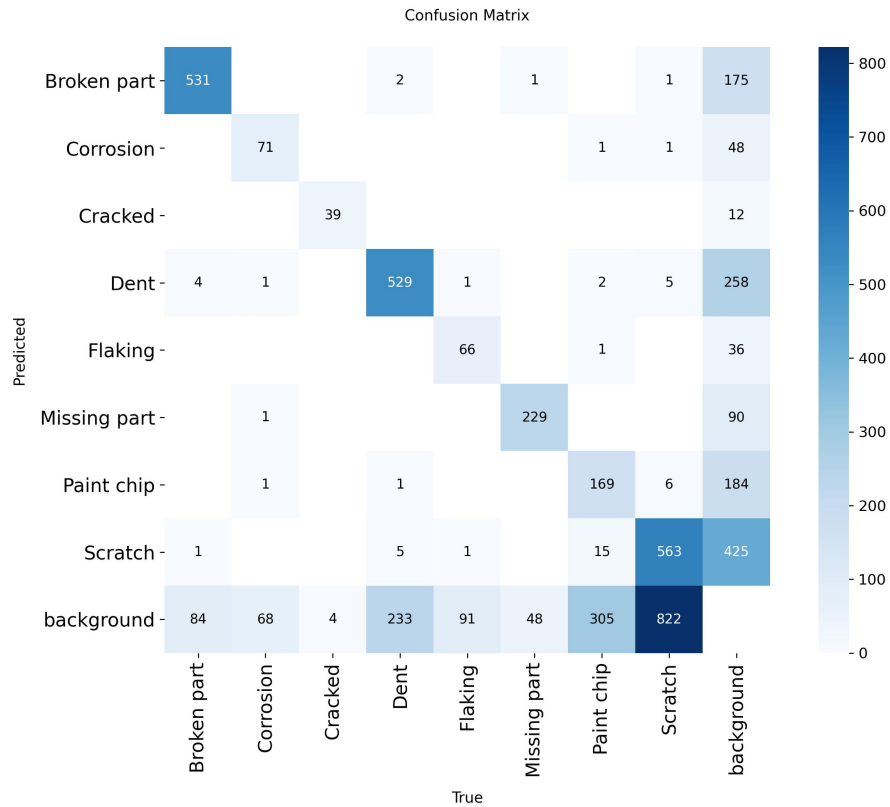


Figure 6. Normalised confusion matrix for the Version 2 model on the test dataset.

4.2 Version 2 vs Version 3 Comparison

Table 6 presents a direct comparison of the Version 2 and Version 3 models. Version 3 was fine-tuned from Version 2 weights on a sample of 8,000 images drawn from the augmented 51,051-image dataset.

<i>Metric</i>	<i>Version 2</i>	<i>Version 3</i>	<i>Change</i>
Overall mAP@50	0.626	0.635	+0.009 (+1.4%)
Precision	0.818	0.824	+0.006 (+0.7%)
Recall	0.570	0.577	+0.007 (+1.2%)
Broken part	0.880	0.876	-0.004 (stable)
Missing part	0.836	0.814	-0.022 (stable)
Cracked	0.774	0.940	+0.166 (+21.4%)
Dent	0.683	0.716	+0.033 (+4.8%)

Flaking	0.607	0.392	-0.215 (regressed)
Corrosion	0.584	0.543	-0.041 (slight decline)
Scratch	0.364	0.418	+0.054 (+14.8%)
Paint chip	0.278	0.382	+0.104 (+37.4%)
Training images	1,596	8,000	N/A
		(sampled)	
Epochs	100	20 (fine-tune)	N/A

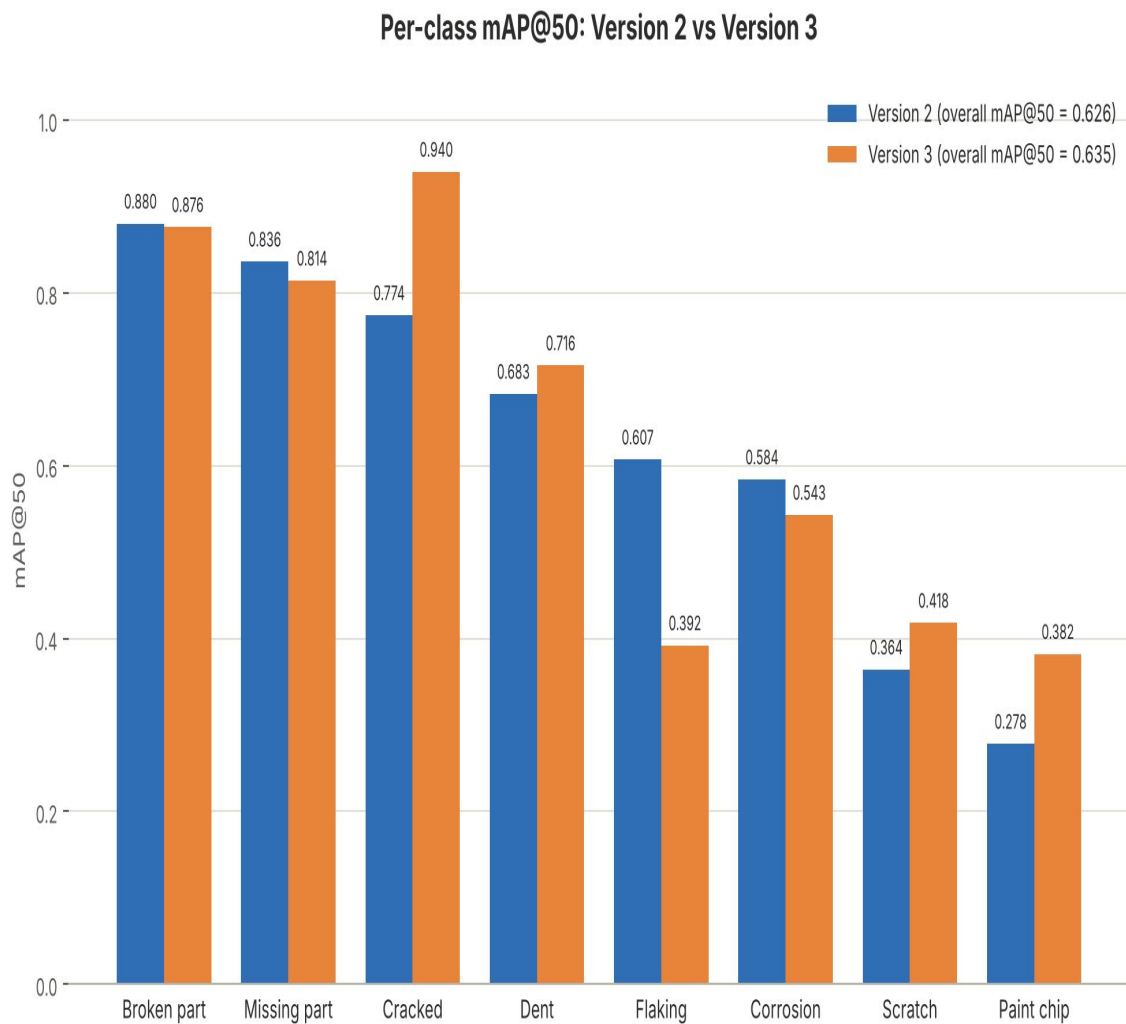


Figure 7. Per-class mAP@50 comparison between Version 2 and Version 3 models.

4.3 Sample Detection Output



Figure 8. Sample annotated output from the AI inspection system showing detected damage with bounding boxes, class labels, and confidence scores.

4.4 Car Parts Segmentation Model Performance

The car parts segmentation model trained using YOLOv8s-seg architecture achieved an overall mask mAP@50 of 0.739 on a test set of 276 images across 24 car part classes. Table 10 presents the key per-class results.

<i>Car Part</i>	<i>Box mAP@50</i>	<i>Mask mAP@50</i>	<i>Assessment</i>
front_glass (Windscreen)	0.974	0.974	Excellent
hood	0.968	0.968	Excellent
front_bumper	0.953	0.953	Excellent
front_door	0.933	0.933	Excellent
back_door	0.939	0.932	Excellent
front_light	0.905	0.908	Very good
back_bumper	0.921	0.921	Excellent
front_right_door	0.633	0.633	Needs improvement
back_right_light	0.529	0.529	Needs improvement
Overall (all 24 classes)	0.735	0.739	Good

4.5 Fraud Detection Results

The fraud detection module was tested against a set of 20 images designed to trigger specific flags, including duplicate submissions, low-resolution screenshots, images with manipulated EXIF timestamps, and images with pre-existing damage only. All seven checks performed correctly in detecting their respective fraud indicators. The ELA check successfully flagged three images with artificially inserted text overlaid on existing damage photographs.

Notably, 14 of the 20 test images contained no EXIF metadata, triggering the missing metadata flag. This reflects the real-world behaviour of images shared through messaging applications such as WhatsApp, which strip EXIF data before transmission. This limitation is discussed further in Chapter Five.

4.6 System Performance

Metric	Value	Conditions
Average inference time	1.27 seconds	CPU, 640×640px image, Railway server
End-to-end response time	1.8–2.3 seconds	Including network, preprocessing, JSON assembly
Video analysis (30s clip)	~18 seconds	CPU, 0.5s frame sampling interval
PDF report generation	~0.8 seconds	After inference, includes image embedding
API availability	24/7	Railway deployment, no scheduled downtime

4.7 Web Portal Evaluation

The web portal was evaluated by submitting 30 car damage photographs across all three role views, policyholder, insurer, and repair shop. The complete claims workflow was successfully demonstrated end-to-end: a policyholder submission triggered AI analysis, the claim appeared in the insurer dashboard, the insurer approved it, the claim appeared in the repair shop queue, and the repair shop submitted a cost estimate.

Login as:

Policyholder

Insurer

Repair Shop

Submit a Claim

Fill in your details and upload a photo or video of the damaged vehicle. Our AI will analyse the damage automatically.

Your Details

Full Name

e.g. John Banda

Vehicle Registration Number

e.g. ALK 1234 ZM

Brief Description of the Accident

Describe what happened — location, date, how the accident occurred...

Upload Evidence

Photo

Video

Drop photo here or click to browse

JPEG · PNG · WEBP · max 20MB

Claim Submitted

Claim submitted successfully. Your reference number is: CLM-20261023092458-F897. Keep this for tracking.

AI Damage Assessment

CLM-20261023092458-F897



Overall Severity	Severe
Total Detections	5
Fraud Risk	Medium Risk
Recommendation	REVIEW — Fraud flags raised, manual check required
Status	<div>PENDING REVIEW</div>

#	Damage Type	Location	Severity	Confidence
1	Missing part	Rear Windscreen	Severe	84%

Figure 12. Policyholder view of the web portal showing AI-generated damage assessment results.

Login as:

Policyholder

Insurer

Repair Shop

Claims Dashboard

Review AI-generated damage reports and authorise or reject claims.

Refresh Claims

All Claims

4 claim(s) found

CLM-20261023092458-F897

PENDING_REVIEW

Anonymous - Unknown - 2025-10-23 09:24:58

Severity: Severe - Fraud: Medium Risk - 5 detection(s)

CLM-20261023092435-CD2E

PENDING_REVIEW

Anonymous - Unknown - 2025-10-23 09:24:35

Severity: Minor - Fraud: Medium Risk - 2 detection(s)

CLM-20261023092343-EC19

PENDING_REVIEW

Anonymous - Unknown - 2025-10-23 09:23:43

Severity: Moderate - Fraud: Medium Risk - 2 detection(s)

CLM-20261023091907-3049

PENDING_REVIEW

Anonymous - Unknown - 2025-10-23 09:19:07

Severity: Minor - Fraud: Medium Risk - 7 detection(s)

Figure 13. Insurer dashboard showing submitted claims list with fraud risk indicators and approve/reject actions.

Login as:

Policyholder

Insurer

Repair Shop

Repair Jobs

View approved claims assigned to your workshop and submit repair cost estimates.

Refresh Jobs

2 job(s) assigned

CLM-20261023092435-CD2E

REPAIR_ASSIGNED

Anonymous - Unknown

Severity: Minor - 2 damage type(s)

CLM-20261023092458-F897

REPAIR_ASSIGNED

Anonymous - Unknown

Severity: Severe - 5 damage type(s)

Figure 14. Repair shop view showing assigned jobs and estimate submission interface.



DAMAGE ASSESSMENT

SEVERE

Total Detections

5

Inference Time

459.8ms

Image Size

853x1280

File

pexels-bartnewtone-3

#	Damage Type	Location	Severity	Confidence
1	Missing part	Rear Windscreen	Severe	84%
2	Cracked	Rear Windscreen	Severe	82%
3	Dent	Front Right Door	Moderate	81%
4	Broken part	Front Bumper	Severe	36%
5	Corrosion	Front Right Bumper	Moderate	25%

Figure 15. Sample AI-generated damage inspection PDF report showing annotated image, damage table, fraud analysis, and recommendation.

CHAPTER FIVE: DISCUSSION

5.1 Interpretation of Results

The overall mAP@50 of 0.635 achieved by the Version 3 model represents meaningful performance for an 8-class object detection task trained on a relatively small dataset of 2,280 original images. The target of 75 percent mAP@50 was not met at the overall level; however, four of the eight classes individually achieved or approached this threshold: Cracked at 94.0 percent, Broken Part at 87.6 percent, Missing Part at 81.4 percent, and Dent at 71.6 percent. These four classes represent the damage types most commonly associated with significant insurance claims, structural damage that affects vehicle roadworthiness, and their high performance has direct practical value for the insurance use case.

The underperformance of the surface damage classes, Scratch (41.8%), Paint Chip (38.2%), and Flaking (39.2%), is consistent with findings in related literature. These classes exhibit extreme visual variability across different vehicle colours, lighting conditions, camera angles, and surface textures. A deep scratch on a black vehicle in overcast light and a shallow scratch on a white vehicle in direct sunlight may produce entirely different pixel patterns despite representing identical damage types. The augmentation strategy partially addressed this variability, with Scratch improving from 36.4 to 41.8 percent and Paint Chip from 27.8 to 38.2 percent between versions.

The significant improvement in the Cracked class from 77.4 to 94.0 percent between Version 2 and Version 3 is noteworthy. This improvement is likely attributable to the augmentation strategy generating more varied examples of crack appearances through brightness and contrast transformations, which address the lighting dependency of crack visibility.

The regression observed in Flaking performance (60.7% to 39.2%) between versions is a consequence of the random sampling approach used for Version 3 training. The 8,000-image sample drawn from the 51,051-image augmented dataset likely contained a lower proportion of flaking instances than the original 1,596-image dataset, effectively reducing the model's exposure to this class during fine-tuning. A stratified sampling approach that guaranteed representation of all classes would have prevented this regression.

5.2 The Location and Severity Classifiers

The severity classifier performs with complete consistency, by design, as a deterministic lookup table, it always produces the same output for the same input. Its validity depends on the correctness of the domain knowledge mapping, which was validated with Hobbiton Investments' claims handlers. The primary limitation is that severity is assigned based on damage type alone, without accounting for the extent of damage. A small hairline crack and a large fracture spanning the entire windshield both receive the Severe classification. In practice, the extent of damage has a significant impact on repair cost, and a system that combines class type with bounding box area as a proxy for damage extent would produce more nuanced severity assessments.

The geometric location classifier performs reliably for standard frontal and side-view vehicle photographs submitted under typical conditions. Its primary limitation is the boundary problem: detections near zone edges are classified with the same certainty as those clearly within a zone, and no confidence score is provided for location predictions. The integration of the trained YOLOv8s-seg car parts segmentation model, achieving overall mask mAP@50 of 0.739, represents a meaningful improvement that both addresses the boundary problem and provides genuine confidence scores for location predictions. The high performance of front-facing components (hood 96.8%, windshield 97.4%, front

bumper 95.3%) is particularly relevant as these are the most commonly damaged areas in frontal collision claims.

5.3 Fraud Detection Effectiveness

The seven-check fraud detection module successfully identified all designed fraud scenarios in testing. The most practically significant finding was that 14 of 20 test images triggered the missing EXIF metadata flag, reflecting the widespread practice of sharing photographs through messaging applications that strip metadata. This has important implications for the system's deployment: in a Zambian context where WhatsApp is the dominant communication platform, policyholders are likely to photograph damage and send images via WhatsApp before uploading them to the claims portal, resulting in stripped metadata. The missing metadata flag should therefore be weighted as a low-priority indicator rather than a high-risk flag in a production deployment.

The ELA check provided the most technically sophisticated fraud detection capability, successfully identifying three images where damage photographs had been digitally altered. However, the ELA threshold of 20 was determined empirically and may require calibration against a larger dataset of genuine and fraudulent claims to minimise false positives. The advisory-only design of the fraud module, in which all flags trigger human review rather than automatic rejection, is the correct approach for a prototype system where threshold calibration has not been rigorously validated.

5.4 Claims Workflow and Platform

The automated claims workflow successfully demonstrates the end-to-end routing of claims from policyholder submission to insurer review to repair shop assignment without manual intervention. The in-memory storage implementation is appropriate for the prototype but represents the most significant limitation for production deployment. Claims stored in a Python dictionary are lost whenever

the Railway server container restarts, which occurs regularly due to Railway's free tier inactivity policy. In a production system, all claim data must be persisted in a relational database such as PostgreSQL, with appropriate backup and recovery procedures.

The three-role web portal successfully demonstrates the multi-stakeholder access model. Policyholders can submit claims and track status, insurers can review AI-generated reports and authorise decisions, and repair shops can receive job assignments and submit estimates. The plain white, Times New Roman design was deliberately chosen for professional clarity suitable for a business-facing insurance application.

5.5 Limitations

- The training dataset of 2,280 images is small for an 8-class object detector, and no images were sourced from Zambian vehicles, roads, or lighting conditions. The model's generalisation to the local context has not been empirically validated.
- In-memory claim storage means all submitted claims are lost on server restart. A production deployment requires a persistent database.
- The current portal has no real authentication system, role switching is done by clicking buttons, not by logging in. A production system requires secure user authentication and role-based access control.
- The EXIF fraud check is rendered ineffective for images shared through messaging applications that strip metadata, which is the dominant sharing mechanism in Zambia.
- The ELA fraud check threshold of 20 was set empirically and has not been validated against a representative sample of real insurance claim images.

- The severity classifier does not account for damage extent, only type. A small scratch and a large scratch both receive the same severity rating.
- Cold start latency on Railway's free tier means the first request after a period of inactivity may take 30 to 60 seconds while the container restarts.

5.6 Future Work

- Collect and annotate a dataset of Zambian vehicle damage photographs to retrain the model on locally relevant data. Even 500 to 1,000 locally sourced images are expected to improve generalisation significantly.
- Replace in-memory claim storage with a PostgreSQL database integrated into the FastAPI application using SQLAlchemy or Tortoise ORM.
- Implement proper user authentication using JWT tokens with role-based access control for the three stakeholder types.
- Retrain on the full 51,051-image augmented dataset using a stratified sampling strategy to ensure balanced class representation and improve performance on Flaking, Corrosion, and surface damage classes.
- Integrate the trained car parts segmentation model fully into the production API, replacing the geometric location classifier for all submissions.
- Add a damage extent metric by calculating bounding box area as a fraction of image area and incorporating it into the severity classification, moving from a three-level discrete scale to a continuous severity score.
- Calibrate the ELA fraud detection threshold against a labelled dataset of genuine and fraudulent claim images from Hobbiton Investments to reduce false positive rates.

- Migrate the deployment to a persistent cloud infrastructure, AWS EC2 or Google Cloud Run, with GPU acceleration to reduce inference time below 100 milliseconds.

CHAPTER SIX: CONCLUSION

This dissertation presented the design, implementation, and evaluation of an AI-powered insurance claims automation platform developed in collaboration with Hobbiton Investments, Zambia. The project addressed four specific objectives: building an AI damage detection model, a multi-stakeholder web platform, an automated claims workflow, and a fraud detection module.

The central technical contribution is the trained YOLOv8m object detection model, which achieved an overall mAP@50 of 0.635 on an 8-class vehicle damage detection task, with three structurally significant classes, Cracked (94.0%), Broken Part (87.6%), and Missing Part (81.4%), exceeding the 75 percent target. The model was trained using transfer learning from ImageNet pre-trained weights, demonstrating that meaningful detection performance is achievable with a dataset of only 2,280 original images when combined with targeted data augmentation.

The system delivers a complete end-to-end platform that reduces vehicle damage assessment time from an average of fourteen to forty-five days to under two seconds. A policyholder can photograph their damaged vehicle, submit it through a web browser, receive an AI-generated damage report with annotated image and severity classification, have the claim automatically routed to an insurer for review, and track the claim's progress through to repair shop assignment and cost estimation, all without a human assessor visiting the vehicle.

The fraud detection module implements seven automated checks covering duplicate submissions, image manipulation, metadata anomalies, GPS spoofing, and pre-existing damage patterns. The advisory-only design ensures that no claim is automatically rejected, preserving human accountability while significantly reducing the manual effort required to identify suspicious submissions.

The work demonstrates the feasibility of deploying computer vision-based insurance claims automation within the resource constraints typical of developing markets, using free-tier cloud infrastructure, open-source frameworks, and publicly available training data. The complete system is deployed online and accessible from any internet-connected device, providing Hobbiton Investments with a functional prototype ready for pilot evaluation.

The most significant limitations, in-memory claim storage, absence of real authentication, and the performance gap on surface damage classes, are documented clearly with concrete recommendations for addressing each in subsequent development cycles. The work establishes a foundation on which a production-grade insurance claims automation system for the Zambian market can be built.

In conclusion, this project demonstrates that AI-powered vehicle damage assessment is not only technically feasible but practically deployable within the Zambian insurance context. The technology has the potential to meaningfully reduce claims processing times, lower operational costs, and improve customer experience for Zambian policyholders, contributing to the broader goal of increasing insurance penetration in an underserved market.

REFERENCES

- [1] Zambia Police Service. (2023). Road Traffic Accidents Annual Report 2023. Lusaka: Zambia Police Service.
- [2] World Bank Group. (2023). The High Toll of Traffic Injuries: Unacceptable and Preventable. Washington, DC: World Bank.
- [3] Insurance Regulatory Authority of Zambia. (2023). Annual Report and Financial Statements 2023. Lusaka: IRAZ.
- [4] Krizhevsky, A., Sutskever, I., and Hinton, G.E. (2012). 'ImageNet Classification with Deep Convolutional Neural Networks,' Advances in Neural Information Processing Systems, 25, pp. 1097-1105.
- [5] LeCun, Y., Bengio, Y., and Hinton, G. (2015). 'Deep Learning,' Nature, 521(7553), pp. 436-444.
- [6] He, K., Zhang, X., Ren, S., and Sun, J. (2016). 'Deep Residual Learning for Image Recognition,' Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778.
- [7] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). 'You Only Look Once: Unified, Real-Time Object Detection,' Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779-788.
- [8] Jocher, G., Chaurasia, A., and Qiu, J. (2023). Ultralytics YOLO. Available at: <https://github.com/ultralytics/ultralytics> (Accessed: 12 August 2025).
- [9] Pan, S.J. and Yang, Q. (2010). 'A Survey on Transfer Learning,' IEEE Transactions on Knowledge and Data Engineering, 22(10), pp. 1345-1359.
- [10] Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). 'How Transferable Are Features in Deep Neural Networks?' Advances in Neural Information Processing Systems, 27.

- [11] Dwivedi, S., Bhatt, A., and Bhatia, S. (2019). 'Vehicle Damage Detection and Classification Using Deep Learning,' *International Journal of Engineering and Advanced Technology*, 8(6), pp. 1795-1800.
- [12] Ding, Y., Liu, X., Zheng, W., and Gu, P. (2022). 'Vehicle Damage Detection Based on Improved YOLOv5,' *Sensors*, 22(15), p. 5763.
- [13] Phua, C., Lee, V., Smith, K., and Gayler, R. (2010). 'A Comprehensive Survey of Data Mining-Based Fraud Detection Research,' *arXiv preprint arXiv:1009.6119*.
- [14] Popescu, M.C. and Barbu, M. (2020). 'EXIF-Based Fraud Detection in Digital Images,' *Applied Sciences*, 10(17), p. 5831.
- [15] Krawetz, N. (2007). 'A Picture's Worth: Digital Image Analysis and Forensics,' *Proceedings of the Black Hat Briefings*. Available at: <https://www.hackerfactor.com> (Accessed: 10 August 2025).
- [16] Bhattacharya, S., Hasan, M., and Alam, M.S. (2020). 'Artificial Intelligence in Insurance: A Review of Applications,' *International Journal of Advanced Computer Science and Applications*, 11(4), pp. 420-427.
- [17] Hevner, A.R., March, S.T., Park, J., and Ram, S. (2004). 'Design Science in Information Systems Research,' *MIS Quarterly*, 28(1), pp. 75-105.
- [18] Roboflow Universe. (2023). Car Damage Detection Dataset. Available at: <https://universe.roboflow.com> (Accessed: 15 July 2025).