UNIVERSITI TEKNOLOGI MARA

# HYBRID DEEP SPATIAL SLICING NETWORK MODELS TO IMPROVE THE CROPS SCENE RECOGNITION AND DETECTION TOWARDS DECARBONIZED PROCESSING EFFICACY

NURHAZIRAH BINTI MOHD RAHIM

MSc

August 2026

# UNIVERSITI TEKNOLOGI MARA

# HYBRID DEEP SPATIAL SLICING NETWORK MODELS TO IMPROVE THE CROPS SCENE RECOGNITION AND DETECTION TOWARDS DECARBONIZED PROCESSING EFFICACY

**NURHAZIRAH BINTI MOHD RAHIM**

Thesis submitted in partial fulfilment
of the requirements for the degree of
**Master of Science**
**(Electrical Engineering)**

**Faculty of Electrical Engineering**

**August 2026**

# CONFIRMATION BY PANEL OF EXAMINERS

I certify that a Panel of Examiners has met on 8th June 2026 to conduct the final examination of Nurhazirah binti Mohd Rahim on her Masters of Science thesis entitled "Hybrid Deep Spatial Slicing Network Models to Improve the Crops Scene Recognition and Detection Towards Decarbonized Processing Efficacy" in accordance with Universiti Teknologi MARA Act 1976 (Akta 173). The Panel of Examiners recommends that the student be awarded the relevant degree. The Panel of Examiners was as follows:

Nur Emileen Abdul Rashid, PhD
Professor
Faculty of Electrical Engineering
Universiti Teknologi MARA
(Chairman)

Mohamad Farid Misnan, PhD
Associate Professor
Faculty of Electrical Engineering
Universiti Teknologi MARA
(Internal Examiner)

Suzaimah Ramli
Associate Professor
Faculty of Defence Science and Technology
Universiti Pertahanan Nasional Malaysia
(External Examiner)

**PROFESSOR DR HJH ZURAEDA IBRAHIM**
Dean
Institute of Postgraduates Studies
Universiti Teknologi MARA

Date: 14 August 2026

# AUTHOR'S DECLARATION

I declare that the work in this thesis was carried out in accordance with the regulations of Universiti Teknologi MARA. It is original and is the result of my own work, unless otherwise indicated or acknowledged as referenced work. This thesis has not been submitted to any other academic institution or non-academic institution for any degree or qualification.

I hereby acknowledge that I have been supplied with the Academic Rules and Regulations for Postgraduate, Universiti Teknologi MARA, regulating the conduct of my study and research.

| | | |
|---|---|---|
| Name of Student | : | Nurhazirah binti Mohd Rahim |
| Student ID. No. | : | 2024581393 |
| Programme | : | Master of Science (Electrical Engineering) – CEEE750 |
| Faculty | : | Electrical Engineering |
| Thesis Title | : | Hybrid Deep Spatial Slicing Network Models to Improve the Crop Scene Recognition and Detection towards Decarbonized Processing Efficacy |
| Signature of Student | : | |
| Date | : | August 2026 |

# ABSTRACT

Recent advances in deep learning-based fruit crop detection have improved recognition accuracy in complex and unstructured agricultural environments. However, these methods often require high computational resources, which may increase processing time, energy consumption, and carbon footprint, particularly when applied to high-resolution unmanned aerial vehicle (UAV) imagery. This study presents a performance analysis of a Deep Spatial Slicing Network (DSSN) framework for improving pineapple crop scene recognition and detection while enhancing decarbonized processing efficacy. The proposed DSSN framework applies an adaptive spatial slicing mechanism to partition high-resolution UAV images into optimized tile sizes according to drone flight altitude, which preserves small pineapple features, reduces spatial uncertainty caused by scale variation, and prevents important crop information from being lost during image down-sampling. The optimized image tiles are processed using three deep convolutional object detection models, namely Faster R-CNN, SSD, and YOLOv8. For the two-stage Faster R-CNN detector, several pretrained backbone networks, including ResNet-101, ResNet-50, InceptionResNet-v2, and VGG-19, are comprehensively evaluated to identify the most effective feature extractor for complex field backgrounds. The experimental evaluation is conducted using pineapple images captured at multiple UAV flight heights, where test images at 4 m, 6 m, 8 m, and 10 m are used to assess object detection performance based on precision, recall, F1-score, and mean average precision. To estimate suitable tile sizes for unseen operational conditions, validation images at 3 m, 5 m, 7 m, and 9 m are modeled using various mathematical approaches, including piecewise linear interpolation, polynomial interpolation, cubic spline interpolation, linear regression, and exponential regression. Among these methods, cubic spline interpolation provides the most reliable and accurate estimation for adaptive tile sizing. The experimental results demonstrate that YOLOv8 achieves the best overall detection performance and reliability in complex pineapple crop scenes, whereas SSD performs less effectively and is less suitable when combined with spatial slicing. Furthermore, the decarbonized efficacy assessment demonstrates that reducing unnecessary computational workload through adaptive tile optimization significantly minimizes redundant pixel processing, which contributes to lower processing time, lower energy consumption, and reduced carbon release during model testing. Therefore, the proposed DSSN framework successfully supports both accurate pineapple crop detection and environmentally conscious deep learning deployment in precision agriculture.

# ACKNOWLEDGEMENT

First and foremost, praises and thanks to God, the Almighty, for His showers of blessing throughout my research work to complete the research successfully. Firstly, I would like to express my gratitude to the Ministry of Higher Education (MOHE) Malaysia, Grant No: 600-RMC/FRGS 5/3 (118/2023) for the graduate research assistance (GRA) sponsorship during my study.

I would like to express my deep and sincere gratitude to my research supervisor, Dr. Muhammad Asraf bin Hairuddin, for giving me the opportunity to do research and providing invaluable guidance throughout this research. He has taught me the methodology to carry out the research and to present the research works as clearly as possible. His sincerity, motivation, vision, and patience have deeply inspired me to finish my research. It was a great privilege and honor to work and study under his guidance. My appreciation extends to my co-supervisor, Prof. Dr. Megat Syahirul Amin Megat Ali and Prof. Dr. Nooritawati binti Md Tahir, for their priceless assistance and support.

I am extremely grateful to my parents, Mohd Rahim bin Yahya and
for their love, prayers, caring and sacrifices for educating and preparing me for my future. I am very much thankful to my beloved ones, Mohd Ridha Fahmi for his love, understanding, constant encouragement and genuine support throughout this research work. Also, I express my thanks to my siblings for their support and valuable prayers. Finally, my thanks go to all the people who have supported me in completing the research work directly or indirectly.

# TABLE OF CONTENTS


| | | Page |
|---|---|---|
| **CONFIRMATION BY PANEL OF EXAMINERS** | | **ii** |
| **AUTHOR'S DECLARATION** | | **iii** |
| **ABSTRACT** | | **iv** |
| **ACKNOWLEDGEMENT** | | **v** |
| **TABLE OF CONTENTS** | | **vi** |
| **LIST OF TABLES** | | **ix** |
| **LIST OF FIGURES** | | **xi** |
| **LIST OF ABBREVIATIONS** | | **xiii** |
| **CHAPTER 1 INTRODUCTION** | | **1** |
| 1.1 | Research Background | 1 |
| 1.2 | Problem Statement | 4 |
| 1.3 | Research Questions | 5 |
| 1.4 | Research Objectives | 6 |
| 1.5 | Significance of Study | 6 |
| 1.6 | Thesis Scope | 6 |
| 1.7 | Thesis Outline | 7 |
| **CHAPTER 2 LITERATURE REVIEW** | | **9** |
| 2.1 | Introduction | 9 |
| 2.2 | Precision Agriculture | 9 |
| | 2.2.1 Characteristics of Pineapples | 10 |
| 2.3 | Deep Spatial Slicing Network | 12 |
| 2.4 | Pretrained Deep Neural Networks | 13 |
| | 2.4.1 ResNet-101 and ResNet-50 | 14 |
| | 2.4.2 InceptionResNet-v2 | 14 |
| | 2.4.3 VGG-19 | 14 |

2.5 Convolutional Neural Networks 14
2.5.1 Faster R-CNN 16
2.5.2 Single Shoot Multibox Detector (SSD) 18
2.5.3 YOLOv8 19
2.5.4 Analysis of Deep Convolutional Methods in Object Detection 21
2.5.5 Hyperparameter Settings 28
2.5.6 Performance Model Measurement 29
2.6 Relation of DSSN between CNN Methods 30
2.7 Contribution of the DLM to the Carbon Dioxide 31

**CHAPTER 3 RESEARCH METHODOLOGY 33**
3.1 Introduction 33
3.2 Data Collection 35
3.3 Data Preparation 36
3.3.1 Train Data 36
3.3.2 Test Data 37
3.3.3 Labeling Data 38
3.4 Generate Tile Size 39
3.4.1 Apply Tile Size 40
3.5 Validation from Test Data 42
3.6 Model Classifier 46
3.6.1 Pretrained Deep Neural Network 46
3.6.2 Deep convolutional models 46
3.7 Performance Measures 49
3.7.1 Classifier Matrix 50
3.7.2 Carbon Footprint Effect 51

**CHAPTER 4 RESULTS AND DISCUSSIONS 54**
4.1 Introduction 54
4.2 Data Collection 54
4.3 Hyperparameter Value 55
4.4 Performance Evaluation 57
4.4.1 Pretrained Deep Neural Network 57

4.4.2 Deep Convolutional Method 66
4.4.3 Results of Interpolation and Regression Lines 73
4.5 Decarbonization Effect 82
4.5.1 Computational Time 82
4.5.2 Carbon Release 85

**CHAPTER 5 CONCLUSION** **88**
5.1 Conclusion 88
5.2 Future Study 89

**REFERENCES** **91**

**APPENDICES** **101**

**AUTHOR'S PROFILE** **105**

# LIST OF TABLES


| Tables | Title | Page |
|---|---|---|
| Table 2.1 | Deep Learning based on Previous Work for Object Detection Using Deep Learning Models | 22 |
| Table 2.2 | Summary of Hyperparameter Settings for DLM | 28 |
| Table 3.1 | Parameters and Derivation Methods for Reference and Validation Datasets Across Different Altitude Levels | 44 |
| Table 4.1 | Dataset for Training and Testing | 54 |
| Table 4.2 | Hyperparameter Values Settings | 55 |
| Table 4.3 | Performance Metrics Using ResNet-101 | 58 |
| Table 4.4 | Performance Metrics Using ResNet-50 | 60 |
| Table 4.5 | Performance Metrics Using ResNet-50 | 61 |
| Table 4.6 | Performance Metrics Using VGG-19 | 63 |
| Table 4.7 | Average Performance Comparison Across Multiple Metrics | 65 |
| Table 4.8 | Performance Metrics Using Faster R-CNN | 66 |
| Table 4.9 | Performance Metrics Using SSD | 67 |
| Table 4.10 | Performance Metrics Using YOLOv8 | 69 |
| Table 4.11 | Tabulate Tabulate Data of Average Performance Evaluation Deep Convolutional Methods | 72 |
| Table 4.12 | Quantitative Comparison of the Predicted Tile Size Obtained through Various Interpolation and Regression Methods Based on the Given Reference Height | 74 |
| Table 4.13 | Evaluation of DLM Performance in Predicting Tile Size Based on Height Measurements Using Piecewise Linear Interpolation | 76 |
| Table 4.14 | Evaluation of DLM Performance in Predicting Tile Size Based on Height Measurements Using Polynomial Linear | 77 |
| Table 4.15 | Evaluation of DLM Performance in Predicting Tile Size Based on Height Measurements Using Cubic Spline Interpolation | 78 |

| | | |
|---|---|---|
| Table 4.16 | Evaluation of DLM Performance in Predicting Tile Size Based on Height Measurement Using Linear Regression | 79 |
| Table 4.17 | Evaluation of DLM Performance in Predicting Tile Size Based on Height Measurement Using Exponential Regression | 80 |
| Table 4.18 | Computational Time of DLM during Testing Time | 82 |
| Table 4.19 | Computational Speed Analysis for DLM | 84 |
| Table 4.20 | Carbon Release of DLM | 86 |

# LIST OF FIGURES


| Figures | Title | Page |
|---|---|---|
| Figure 2.1 | Schematic Architecture of the Faster R-CNN Model | 17 |
| Figure 2.2 | The Model Structure of an SSD Consists of Training and Prediction Networks | 18 |
| Figure 2.3 | The YOLOv8 Architecture Showcases Its Innovative Design for Object Detection and Classification | 20 |
| Figure 3.1 | General Methodology of Research | 33 |
| Figure 3.2 | The Main Flow Model Diagram | 34 |
| Figure 3.3 | Cropped Pineapple for Training Images | 36 |
| Figure 3.4 | Sample Set of Test Images | 37 |
| Figure 3.5 | Labeling Process of Pineapple Images by Using the makesense.ai Website | 38 |
| Figure 3.6 | Method of Spatial Slicing Images | 40 |
| Figure 3.7 | Stages of Applying Tile Size on Testing Image: Original Image of Pineapple Crops (a), Standard Tiled Image Without Overlap (b), and the Resultant Overlapping Tiles for Enhanced Detection (c) | 42 |
| Figure 3.8 | Overall Flowchart for Validation from Test Images | 43 |
| Figure 3.9 | Sample Set of Dataset Images for Validation from Test Data | 44 |
| Figure 3.10 | Schematic Architecture of the Faster R-CNN Model | 47 |
| Figure 3.11 | Model Structure of SSD | 48 |
| Figure 3.12 | The Architectures of YOLOv8 | 49 |
| Figure 3.13 | Overview of the Performance Measures Used in This Study | 50 |
| Figure 3.14 | Confusion Matrix | 50 |
| Figure 4.1 | Demonstration of Pineapple Detection from the Image by Using ResNet-101 | 59 |
| Figure 4.2 | Demonstration of Pineapple Detection from the Image by Using ResNet-50 | 61 |

| | | |
|---|---|---|
| Figure 4.3 | Demonstration of Pineapple Detection from the Image by Using InceptionResNet-v2 | 62 |
| Figure 4.4 | Demonstration of Pineapple Detection from the Image by Using VGG-19 | 64 |
| Figure 4.5 | Average Performance for the Pretrained Network | 65 |
| Figure 4.6 | Average Performance Evaluation Deep Convolutional Methods | 72 |
| Figure 4.7 | Comparing the Predicted Tile Size Obtained through Various Interpolation and Regression Methods Based on the Given Reference Height | 74 |
| Figure 4.8 | Utility and Efficiency Value for DLM | 84 |

# LIST OF ABBREVIATIONS

**Abbreviations**

| | |
|---|---|
| ACC | Accuracy |
| ARB | Adaptive Red and Blue Chromatic Map |
| CF | Carbon Footprint |
| CI | Carbon Intensity |
| CNN | Convolutional Neural Network |
| COD | Cascade Object Detector |
| DEA | Decarbonized Efficacy Algorithm |
| DL | Deep Learning |
| DLM | Deep Learning Method |
| DSSN | Deep Spatial Slicing Network |
| EFF | Energy Efficiency |
| FPR | False Positive Rate |
| FPS | Frame per Second |
| GPU | Graphics Processing Unit |
| HVS | Hue Value Segmentation |
| mAP | Mean Average Precision |
| NDI | Normalized Difference Index |
| PA | Precision Agriculture |
| PRE | Precision |

| | |
|---|---|
| REC | Recall |
| ROC | Receiver Operating Characteristic |
| SSD | Single Shot-Multibox Detector |
| TPR | True Positive Rate |
| TPU | Tensor Processing Unit |
| UAV | Unmanned Aerial Vehicle |
| YOLO | You Only Look Once |

# CHAPTER 1
# INTRODUCTION

## 1.1 Research Background

The pineapple, scientifically known as Ananas Comosus, farming represents a significant economic sector with high export values of the global pineapple industry in tropical countries such as Malaysia, the Philippines, Thailand, Indonesia, and Costa Rica. These nations are among the largest pineapple hubs for commercializing fresh or canned fruits for domestic supply and export (Bandara et al., 2022). In 2024, Malaysia produced 520,000 tons of pineapples, a 17% increase from 2023. According to the Food and Agriculture Organization (FAO), Malaysia ranked 20th globally in pineapple production in 2023, contributing approximately 1.34% of the world's output. Next, according to official data, Johor is the largest producer of pineapples and has the largest area of pineapple plantations on peat soil in Malaysia. Johor has long been the top pineapple producer, contributing approximately 68% of Malaysia's total pineapple acreage.

In various pineapple farming stages, activities such as crop counting, monitoring, yield estimation, fruit maturity classification, and disease detection have been observed by fully utilizing advanced automated crop detection and drone technologies (Y. Li et al., 2024; Liu & Wang, 2024; Meng et al., 2023; Rahimi et al., 2020). The growing interest in using aerial vehicles, or drones, in the plantation sector has proven cost-effective and time-saving, especially for both small- and large-scale plantations. In precision fruit farming, drone technology has proven to be effective for capturing image data of fruit trees on farms (Imaging & Learning, 2024). The use of drones in agricultural crop settings has seamlessly blended advanced imaging and artificial intelligence for automating crop detection, which significantly contributes to Agriculture 5.0 by enabling more precise, efficient, sustainable, and data-driven decisions integrated into current farming practices (Agarwal et al., 2018)(Al-shareeda et al., 2022).

The growing focus on pineapple as a high-value crop is evident in tropical countries, as it serves as a viable source of income, especially for smallholder farmers. Sustainable pineapple production can be achieved in part through sound crop

management practices, including yield estimation using precision agricultural technology to make more informed decisions about crop management (Alaieri, 2024). The conventional process involves the manual reporting of crop yields by agricultural officers before harvesting, to estimate crop output. This requires improvement, considering the increase in pineapple farming efficiency. To date, several approaches are available to automate current farming practices, namely, remote sensing (Gonzalez-Hernandez et al., 2023; Pham et al., 2024; Sakuma & Yamano, 2020), computer vision (Ali et al., 2023; Alqethami et al., 2022; Meng et al., 2023), and machine learning algorithms (Kundur & Mallikarjuna, 2022; Mohd Ali et al., 2022; Qiu et al., 2023).

The accuracy and efficiency of the pineapple crop classification process have been improved by incorporating automated image analysis. In (Rahimi et al., 2020), W. N. S. Rahimi et al. proposed an improved segmentation approach comprising image thresholding and HSV color space transformation for crop counting in pineapple plantations. This aids in effectively identifying features in low-contrast images, which may result from varying heights and lighting conditions in the captured color image. In (Cuong et al., 2022), N. H. H. Cuang et al. developed a recognition algorithm using the YOLO-v4 model to improve accuracy and reduce training time in predicting fruit maturity before harvesting. However, the proposed approach requires a large dataset to obtain a robust model. Thus, a combined method, which is composed of a Cascade Object Detector (COD), Hue Value Segmentation (HVS), Adaptive Red and Blue Chromatic Map (ARB), Normalized Difference Index (NDI), and Convolutional Neural Network (CNN), results in a high-quality image for accurate maturity identification and segmentation of pineapples from the background. The authors in (Hobbs et al., 2021) proposed the automatic counting and localization of flowering pineapple plants by analyzing the density of flowers in the images using the U-net backbone model, which resulted in accurate counting performance with low error, thereby enhancing the efficiency and effectiveness of pineapple harvesting operations.

In addition, environmental complexity in the agricultural sector can be characterized by several substantial challenges that must be addressed for robust field implementation. However, ineffective management and large-scale production affect both the production volume and the company's global market competitiveness. The need for automatic identification through technology-oriented approaches, such as image processing and deep learning (DL), has advanced in addressing challenges that have been reported for various types of crops, such as computationally intensive processing,

object detection difficulties due to color similarity, lighting variabilities, occlusion in natural outdoor environments, and overcounting of fruits (Wang et al., 2025).

Thus, the Deep Spatial Slice Networks (DSSN) model offers an innovative method for crop scene recognition and detection using DL. This is especially useful for tasks involving large volumes of image data that must be processed efficiently and intelligently. This model segments the input image of a crop scene into smaller slices or patches, enabling more granular pixel-level analysis. The segmentation process was greatly enhanced by this slicing technique, allowing the model to precisely recognize and separate various scene components, such as pineapple and background features. The effectiveness of DSSNs and similar models highlights their role in the broader transition to the PA.

Faster R-CNN, SSD, and YOLO are now the dominant deep learning architectures for fruit detection and recognition, with YOLO and SSD favored for real-time applications and Faster R-CNN often achieving superior localization in dense scenes. Faster R-CNN can accurately detect multiple fruit types, such as mangoes, apples, and almonds, in orchard environments and performs reliably under complex lighting and occlusion conditions. The SSD, commonly used to improve single-shot detectors, has been successfully applied to fruit maturity recognition and multi-class ripeness detection, confirming that SSD-based architectures can achieve accurate object localization and classification in large agricultural images. Meanwhile, YOLOv8 variants for agricultural tasks, including real-time apple detection, melon maturity detection, and fruit-tree-trunk localization, consistently achieve high mAP and FPS with fewer parameters and lower computational cost, underscoring the suitability of YOLOv8 for resource-constrained field deployments.

Based on these findings, the DSSN can be viewed as a spatial pre-processing and scene-structuring layer that slices very large crop scenes into optimally scaled tiles before feeding them into the Faster R-CNN, SSD, or YOLOv8. By balancing object-to-background ratios, stabilizing spatial context within each tile, and mitigating resolution loss due to downsampling, the DSSN is expected to improve region proposal accuracy in Faster R-CNN, strengthen multi-scale feature learning in SSD, and sharpen grid-based predictions in YOLOv8 by reducing intra-tile noise. Thus, although DSSN is a development model, its design is strongly grounded in and justified by prior evidence on patch-based CNNs and high-performance fruit detection using Faster R-CNN, SSD, and YOLOv8 in complex agricultural environments.

This study intends to develop a combined model, deep spatial slices with deep convolutional networks, that can effectively separate raw pineapple images from their complex backgrounds and detect the crops. Ultimately, the proposed model was validated by comparing its performance with that of a deep convolutional method (DCM) baseline using metrics such as the receiver operating characteristic (ROC), true positive rate (TPR), false positive rate (FPR), accuracy (ACC), precision, recall, and F1-score. Next, a decarbonized efficacy algorithm (DEA) was developed to accurately calculate the carbon footprint of crops recorded by UAV during runtime processing. The anticipated research results will provide a pioneering model that can enhance crop recognition through a unified network structure and runtime processing capability. This will lay the foundation for improving the network structure and the correlation between runtime processing and decarbonization elements to support carbon-neutral policies, thereby positively affecting the agricultural sector.

## 1.2 Problem Statement

Video and high-resolution aerial imagery have altered content complexity, driven by sophisticated agricultural datasets and recordings of complex field case studies. UAVs can simultaneously capture images and videos of large plantation scales. Still, difficulties with semantic segmentation and recognition arise from factors such as color similarity, uneven illumination, dense crop sections, and variations in object scale. Consequently, it has proven difficult to segment, detect, and classify crops using a single standard model. In this study, "high uncertainty" refers specifically to the spatial ambiguity, low feature visibility, and unstable object-boundary representation that occur when pineapple fruits appear small, partially overlapped, or visually similar to surrounding leaves and soil in UAV images. Traditional machine learning techniques, such as Support Vector Machines (SVM) and Random Forest (RF) (Talaviya et al., 2020), are task-specific and poorly equipped to handle such highly variable field conditions. Therefore, advanced deep learning models that can preserve fine-grained spatial information and resolve this high uncertainty are required to recognize pineapples within complicated backgrounds (Tang et al., 2020).

Standard baseline deep convolutional models, such as Faster R-CNN, SSD, and YOLO, typically require large training datasets and complex network architectures to achieve high detection accuracy. However, their performance becomes less reliable

when high-resolution UAV images are processed directly as full images because resizing and down-sampling can reduce the visibility of small features. For example, in a standard YOLOv8 model without spatial slicing, pineapple regions occupy only a small portion of the image, leading the detector to learn more background-dominant features and produce less stable bounding-box predictions. To address this issue, the Deep Spatial Slicing Network (DSSN) is proposed to provide more precise structuring at the local tile level by dividing images based on flight altitude and object-size characteristics. However, its capability to handle complex agricultural scenes from multi-height drone images remains to be systematically evaluated. Therefore, it is necessary to evaluate and compare the proposed model against these baseline models using specific metrics (Prado et al., 2021).

Both baseline deep learning models and DSSN architectures require considerable computational power, repeated model inference, and high GPU utilization, which further exacerbates the carbon footprint problem in digital farming. Existing pineapple detection studies commonly emphasize accuracy but provide limited evaluation of how model design and runtime processing contribute to this computational carbon footprint. Consequently, this increases the need to develop energy-efficient techniques by optimizing runtime and selecting lighter network architectures to minimize active GPU energy consumption (Schwartz et al., 2017). To address this sustainability challenge, this study introduces a Decarbonized Efficacy Algorithm (DEA) that estimates the carbon footprint based on runtime processing, GPU power consumption, and carbon intensity. By integrating this uncertainty to balance learning and environmental objectives, the framework can attain high performance without substantially compromising environmental sustainability.

## 1.3 Research Questions

a) How can the proposed model perform detection scene tasks with a single model, while ensuring high-quality detection and efficient processing?
b) How can the targeted crops and complex backgrounds be distinguished and optimized to improve recognition performance?
c) Why must a decarbonization algorithm be developed to evaluate the model's effectiveness and efficacy?

## 1.4 Research Objectives

The objectives of this study were as follows:

i. To develop a deep convolutional model on deep sliced networks to detect between images of pineapples with complicated backgrounds.
ii. To evaluate the proposed model by comparing its performance with the DLM baseline using the measures TPR, FPR, precision-recall performance, and mean average precision.
iii. To determine a decarbonized efficacy algorithm (DEA) that estimates the carbon footprint based on runtime processing.

## 1.5 Significance of Study

In this study, we evaluated the accuracy of detecting and classifying pineapples in complex backgrounds using a combination of a deep slice spatial network (DSSN) and a deep convolutional model (DCM). These findings will contribute to predicting pineapple value by selecting a suitable tile size for the crop image captured by a UAV from any height above the ground before harvesting. Therefore, this study contributes to predicting pineapple production on pineapple farms by mitigating the decarbonization effect, even when using deep convolutional models.

## 1.6 Thesis Scope

- Data Collection Location: The aerial image dataset was captured at a local pineapple plantation located at Simpang Renggam, Kluang, Johor, by using a UAV with a DJI Phantom 3 Advanced quadcopter with a 4 K resolution RGB camera. The study area selected for this research work is at coordinates 1º48'55.9" N and 103º15'33.5" E.
- Weather and Environmental Conditions: To ensure data consistency and minimize lighting variances, all image acquisitions were conducted under clear, sunny weather with optimal daylight conditions. Extreme conditions such as strong winds, cloud shadows, or rainy weather were avoided to prevent UAV instability and blurry images.

- Aerial Image Acquisition: Image datasets of the pineapple fields were collected using an Unmanned Aerial Vehicle (UAV) flown at multiple altitudes ranging from 3 m to 10 m above the ground level.
- Dataset Preparation:
    i. Train Dataset: A total of 300 localized images of pineapples were generated by cropping various positions from the high-resolution aerial photographs captured specifically at a flight height of 10 m.
    ii. Testing Dataset: Acquired at exact heights of 4 m, 6 m, 8 m, and 10 m above the ground.
    iii. Validation Dataset for Test Data: Collected at the remaining intermediate heights (3 m, 5 m, 7 m, and 9 m) to represent unseen operational data.
- Model Integration and Object Detection: The Deep Spatial Slicing Network (DSSN) model was integrated with three deep convolutional models, which are Faster R-CNN, SSD, and YOLOv8, to evaluate pineapple recognition and detection capabilities in dense and complex agricultural backgrounds.
- Tile Sizing: An adaptive tile sizing strategy based on ground-truth bounding box statistical measures was deployed to enhance aerial detection accuracy across varying heights.
- Sustainability Evaluation: The environmental impact was bounded by quantifying the carbon footprint released during the model testing phases. This was calculated using energy-based metrics derived from the model testing time, Graphics Processing Unit (GPU) power consumption, and a constant carbon intensity factor.

## 1.7 Thesis Outline

This thesis comprises five chapters, each exploring a specific dimension of the research to provide a comprehensive model for automated crop monitoring, including the recognition and detection of pineapple in complex scenes. The structural organization of these chapters is outlined below.

Chapter One presents the research background, problem statement, and research questions, followed by the objectives of the study, scope of the study, and its significance. This introductory section establishes the research context, highlights the

challenges of manual counting in modern agriculture, and frames the precise boundaries within which this study operates.

Chapter Two presents a comprehensive literature review to ground the study in existing academic knowledge. This chapter comprises published studies and historical baselines on pineapple plantations, deep slice spatial networks, deep learning methods for detection and classification, performance measures, and carbon release effects.

Chapter Three presents the study's methodology, outlining the systematic approach used to achieve the research objectives. This section thoroughly discusses the processes involved in data acquisition, data preparation, model classifiers, and the specific performance measures used for both the classifier matrix and the carbon footprint effect.

Chapter Four presents the results and discussion of this study, serving as the core empirical contribution of the research. This chapter discusses all three deep learning methods, combined with the deep slice spatial network, to determine the most significant model for detecting and classifying pineapples in a complex scene. Additionally, the standard performance measures and decarbonization effects are critically analyzed in this section.

Finally, Chapter Five presents the conclusion and recommendations for future work. This closing chapter synthesizes the study's primary insights, assesses the fulfillment of the research objectives, and proposes strategic avenues for prospective researchers to build on these findings.

# CHAPTER 2
# LITERATURE REVIEW

## 2.1 Introduction

This chapter clarifies the use of unmanned aerial vehicles in precision agriculture. The paper then discusses deep learning, including segmentation, detection, and classification. Subsequently, it presents research on CNN methods such as Faster R-CNN, YOLO, and SSD. It also provides a few studies on the performance model evaluations of CNN models. Subsequently, the contribution of the deep learning model to carbon dioxide is examined. All research was reviewed five years ago (2020-2025).

## 2.2 Precision Agriculture

Precision agriculture (PA) is a scientific approach that improves agricultural output and assists decision-making by utilizing modern sensors and analytical capabilities. PA is a newly conceived global phenomenon aimed at enhancing production, reducing manual labour, and optimising fertiliser and water application strategies. Currently, the most common technologies implemented in PA include drones and UAVs (Kukunuri & Singh, 2022). UAVs can obtain high-resolution images very quickly and efficiently. Hence, UAVs are used for several purposes to enhance agricultural management, such as monitoring crops, mapping fields, and estimating crop yields. Furthermore, UAVs can be used to determine the health of plants because the system of UAVs provides operational flexibility and cost efficiency (Sharma, 2023). UAVs reduce the risk of wasted effort, improve task efficiency, and provide real-time data to assist in the management of agricultural workflows (Raouhi et al., 2023). More efficient sensors and imaging technologies can improve the collection of data critical to PA from UAVs (Mazzia et al., 2020).

Furthermore, PA, as part of advanced farming techniques, utilizes UAV image processing and DLM and is one of the most recent approaches currently being researched for managing agronomic variability, achieving environmental sustainability, improving crop quality, and implementing efficient management practices [17]–[19]. Concerning pineapple farming, deep learning techniques can be used to classify

pineapple characteristics, such as maturity and quality, based on color, shape, and texture [20], detect pineapple fruits and flowers in cluttered backgrounds [21], and accurately automate the detection of ripe pineapples and their 3D distance in fields [22]. Several recent works, [29] have demonstrated that agricultural robotics can localize crop positions and pick pineapples with superior performance of 92%. Similarly, Lai et al. (2023) developed a pineapple fruit detection method that targeted various maturity levels and achieved 95.82% detection accuracy. (Shiu et al., 2023) classified mature pineapple fruits using a two-stage region-proposal network technique with an accuracy of 97.46 %. The above works showed that pineapple detection for designated tasks is feasible, even though model performance is constrained by varying environmental conditions and the characteristics of pineapple fruits.

### 2.2.1 Characteristics of Pineapples

The automated detection of pineapples (Ananas comosus) using computer vision requires a foundational understanding of the crop's physical transformation throughout its growth cycle. From an aerial perspective, such as imagery captured by Unmanned Aerial Vehicles (UAVs), a pineapple plantation presents a highly dynamic visual grid in which the vegetative and generative phases dictate its spatial footprint (Rahimi et al., 2020). A mature pineapple plant typically develops a dense rosette consisting of 70 to 80 active, sword-shaped leaves that radiate outwards. This distribution creates a complex, starburst-like canopy pattern that complicates feature extraction during high-throughput field mapping and increases background clutter.

As the crop progresses toward the generative stage, the induction of flowering transitions into fruit development, shifting the primary target of object detection from the wide crown to the localized inflorescence or fruit body. The fruit's physical maturity is largely determined by chronological duration and striking visual changes in the peel or exocarp. Early-stage immature fruits and emerging buds are often hidden or partially occluded by the rigid, upright crown leaves, exhibiting a deep, uniform green hue that blends seamlessly with the surrounding foliage (Shiu et al., 2023). This green-on-green background configuration frequently leads to high false-negative rates due to the extremely low contrast between the target object and the background canopy.

Over a maturation period spanning several months, the individual fruitlets undergo a distinct color transition, shifting from an intense green to varying shades of

yellow and eventually to a rich reddish-orange at peak ripeness (Meng et al., 2023). In deep learning contexts, this color variance serves as both a critical feature baseline for classification and a significant source of intra-class variability. Consequently, this requires single-stage and two-stage models to remain robust against shifting spectral signatures under natural, shifting ambient lighting (Wan Nurazwin Syazwani et al., 2022).

#### *2.2.1.1 A Comparative Morphological Analysis of Pineapple Rosettes and Immature Oil Palm Canopies*

A major challenge in localized agricultural remote sensing is managing inter-class similarity, particularly when deep learning models are trained to detect small or emerging crops in complex, multi-cropped farming landscapes. For instance, distinguishing an early-stage pineapple bud from a young, immature oil palm crown (Elaeis guineensis) presents a highly intricate spatial puzzle for convolutional feature extractors (Terven et al., 2023). When observed from a high-altitude nadir or top-down view, both botanical structures share an overlapping geometric signature, characterized by a central growth point surrounded by radiating linear fronds or leaves. This shared radial symmetry can lead to severe false-positive detections when a network relies solely on low-level edge features or shallow convolutional layers (Jia et al., 2024).

However, closer empirical analysis reveals distinct morphological divergence that advanced deep learning backbones can exploit. While an immature oil palm crown exhibits a highly structured, feathered pinnate leaf arrangement with larger spatial dimensions and prominent V-shaped shadowing between overlapping fronds, a pineapple rosette is significantly more compact. The leaves of a pineapple rosette are spirally arranged according to tight Fibonacci ratios, creating a denser, more uniform spiral pattern with significantly less inter-frond spacing (Meng et al., 2023).

Furthermore, the texture of a young oil palm canopy displays a softer, more fluid linear spread as the fronds bend outward. In contrast, pineapple leaves maintain a rigid, lanceolate posture that casts sharp, fragmented geometric micro-shadows within their immediate tile block. Relying on traditional hand-crafted feature extractors often fails under these complex conditions. This operational limitation underscores the necessity of integrating advanced spatial partitioning and slicing networks to successfully

segment these subtle structural variations from high-resolution aerial datasets (Terven et al., 2023)(Jia et al., 2024).

## 2.3 Deep Spatial Slicing Network

A Deep Spatial Slicing Network (DSSN) in agriculture is a deep learning architecture that partitions spatial data into smaller slices and models the configurations between slices to detect and classify different spatial yields. This is extremely valuable for data-intensive image processing tasks. The contribution and effectiveness of models such as the DSSN demonstrate the relevance of the model to the further development of PA. The use of deep learning to explore spatially resolved data demonstrates the importance of processes for effectively managing and deriving insights from large datasets to improve agricultural systems. The use of data-intensive deep learning systems to explore datasets in a spatially resolved manner to enhance, improve, or promote DSSN demonstrates the positive effects of technologically responsible farming systems (Feng, 2024).

In precision agriculture, DSSN can analyze large spatial data obtained from remote sensing systems, such as drones. The DSSN involves slicing geospatial images at fine resolutions, enabling it to extract spatial features while retaining the overall pattern within a defined agricultural domain. According to (Yamashkina et al., 2021), deep learning systems perform with superior efficiency compared to systems that employ other techniques in the areas of spatial pattern recognition and crop management , thus proving that DSSNs have utility in precision agriculture. With the advancement of agricultural technology, the use of DSSNs will be crucial in shaping future practices aimed at sustaining and enhancing the productivity of the agricultural sector.

“Deep” in DSSN refers to the use of deep learning CNNs, which can learn more abstract features across layers of processing. In summary, deeper network layers can capture higher levels of abstraction in the input data, making them more effective at extracting features and classifying data. (Z. Li et al., 2022; Wan et al., 2022). In addition, DSSN is more beneficial in this case for analyzing areas within a spatial region of interest because of the network’s ability to retain high-detail image resolution while performing a range of processing tasks. Multiple studies have shown that the arrangement of the DSSN architecture is extremely proficient in seizing and

understanding complex characteristics that are likely to be ignored by other conventional region-based approaches, in large part because of the ability of those other architectures to neglect the spatial context of the image and focus on specific regions of interest (X. Zhang et al., 2023).

## 2.4 Pretrained Deep Neural Networks

Pretrained Convolutional Neural Networks have gained research interest in crop classification owing to their ability to produce accurate classifications based on visual characteristics captured in images. Pretrained CNN models have demonstrated superior performance in fruit classification across various crops. However, the performance of pretrained models varies with the crop, so no single model fits all crops. This is due to variations in crop appearance or features resulting from environmental conditions, differences in dataset size and quality, and classification requirements that can influence the performance of a pretrained model. For instance, the authors of (Magabilin et al., 2022) found that VGG-19 outperformed InceptionV3 and MobileNetV2 in classifying the ripeness of Philippine Guyabano fruit, achieving 99.25% accuracy. Based on (A. Kaur et al., 2024), the authors showed that the VGG19 model excels at identifying fine details and patterns in dragon fruit images. Similarly, in (Mohapatra et al., 2022)The authors evaluated ResNet50 and VGG-19 for fruit quality classification, with VGG-19 achieving the best performance, achieving over 94% accuracy. In (Bandyopadhyay et al., 2023), the authors showed that ResNet50 V2 achieved 98.89% accuracy on a dataset with 41 fruit categories. Additionally, a pretrained deep CNN has proven effective for classifying fused images, with ResNet-50 demonstrating superior accuracy and convergence speed (Patil & Mane, 2023). The VGG-19 model is excellent for detecting finer details and patterns in images, especially in agriculture, and for classifying fruits and vegetables against crowded backgrounds. For instance, (Hindarto et al., 2023) highlighted that VGG-19 can recognize fine details that surrounding elements may obscure and effectively classify images, even when images contain overlapping foliage or other distractions due to varying colors and lighting conditions.

### 2.4.1 ResNet-101 and ResNet-50

ResNet-101 and ResNet-50 are both variations of the Residual Network (ResNet) architecture, designed to enable effective CNN training. Specifically, ResNet-101 consists of 101 layers. In comparison, ResNet-50 contains only 50 layers, whereas both architectures use residual connections that mitigate the vanishing gradient problem and facilitate the training of deeper networks. Moreover, ResNet-101 has a deeper architecture, which allows it to capture more intricate features and achieve slightly better performance on complex visual or image processing tasks. In contrast, ResNet-50 is computationally more efficient and has a lower processing time.

### 2.4.2 InceptionResNet-v2

Inception-ResNet-v2 is a convolutional neural network architecture that combines the Inception module's multi-scale feature extraction, using Inception modules with varying filter sizes within each layer, with the residual connections of ResNet to facilitate gradient flow in deep networks. This hybrid architecture aims to improve the accuracy and computational efficiency of image classification tasks.

### 2.4.3 VGG-19

VGG-19 is a CNN architecture from the Visual Geometry Group that consists of 19 layers, comprising 16 convolutional layers followed by 3 fully connected layers. This architecture is known for its simplicity and uniformity, in which each convolutional layer uses a 3 × 3 filter size, followed by a max-pooling layer to reduce the spatial dimensions. This architecture is effective for image classification tasks because of its deep layers and relatively small convolutional filters, which help capture detailed patterns in images.

## 2.5 Convolutional Neural Networks

CNNs are layered network architectures used in deep learning models, modeled after similar networks in the human brain. Deep learning has aided image classification, language translation, and speech recognition (Olaoye & Potter, 2024). It can be used to

solve any pattern-recognition problem without human intervention. CNNs have demonstrated remarkable efficacy in computer vision applications involving image segmentation, detection, and classification. Deep learning-based segmentation approaches, along with technological developments in hardware such as GPUs, provide a practical approach for assessing ROI in collected videos. Image segmentation aims to isolate a specific part or object in an image to create a meaningful representation of the system under examination. Since 2015, CNN-based models for image segmentation have grown in popularity, inspired by the Fully Convolutional Networks (FCN) study, which used a CNN-based classification model for segmentation (Zhao et al., 2024). Furthermore, other researchers have proposed several architectures to address existing segmentation issues, particularly in UAV-based aerial imaging, which is complex and challenging to apply due to the large number of ground objects in each image. (Poliyapram et al., 2019) enhanced the segmentation models, but at the expense of increased complexity and processing overhead. In addition, (Stewart et al., 2019) used the Mask R-CNN model to accurately detect and segment a single lesion in the test group for NLB lesions in UAV images, allowing accurate quantitative assessments of plant diseases at high throughput.

Several significant studies have been conducted on plant detection. Earlier stages of plant recognition relied heavily on machine learning and digital image processing to detect and identify target plants, enabling rapid detection of target crops. (Raja et al., 2020) suggested a real-time system for online weed detection and classification using plant signals in lettuce. Besides, (Azadbakht et al., 2019) employed machine learning algorithms to interpret hyperspectral data from wheat leaves and achieved reasonable results. However, there is still considerable room for improvement in the accuracy of plant recognition. (P. Kaur et al., 2022) investigated leaf disease detection in grapevines and demonstrated that transfer learning can increase learning outcomes. Although sophisticated object identification approaches have been successfully applied to image and video recognition, they have substantial drawbacks in object recognition. They perform poorly when characterizing objects of extreme size and scale in dense scenes with small scales.

The presence of an object may occur only for a brief period or in a small area of the visual field in the video data, making it challenging to obtain a video characteristic that can be used to categorize the action or event. Identifying crop ROI mainly relies on transfer learning, which reduces the number of training hours required to train the

model. (Koklu et al., 2021) demonstrated that the applicability of rice variety classification models reached 99.95% using deep learning. There is a possibility of overfitting occurring when the same models are trained on different bull datasets, even though they perform well when they are trained on objects that have been pretrained (Plested & Gedeon, 2022). When trained models are applied to a completely different distribution of test images, their performance suffers.

### 2.5.1 Faster R-CNN

Faster R-CNN integrates the entire region proposal generation into the CNN architecture via a Region Proposal Network (RPN), thereby improving the accuracy and efficiency of object detection. The model can simultaneously learn to generate and classify region proposals, thus enhancing the detection speed (An et al., 2022). Figure 2.1 illustrates the architecture of the Faster R-CNN used for the crop detection tasks. This integration reflects the success of improving the efficiency and accuracy of the CNN architecture for object detection tasks (Tarkan Abdullah et al., 2022). The integration of the RPN enables Faster R-CNN to be trained end-to-end, so that the entire model learns to generate and classify region proposals. This coordination of constituent models yields a model that detects and classifies with higher accuracy and achieves higher detection speed by utilizing interfeatures more effectively (An et al., 2022).

Compared with conventional models, Faster R-CNN improves performance in several real-time applications, yielding significant gains. Hence, the model and its variants continue to be proposed and reported in the literature (Warin et al., 2024). Almost all models have also been developed to detect and classify objects in even more challenging environments, including a wide range of agricultural diseases, illustrating the difficulty that some advanced models might have in such tasks (Y. Xiao et al., 2020). All these Faster R-CNN adaptations demonstrate that it has sufficient support for robust, real-world demands across many environments and has the potential for many more in the future. This achievement reflects the successful integration of multifaceted processes within a novel framework, as well as the promise of subsequent advancements predicated on this foundational layout and core design in the domain of computer vision (Nasution et al., 2023).

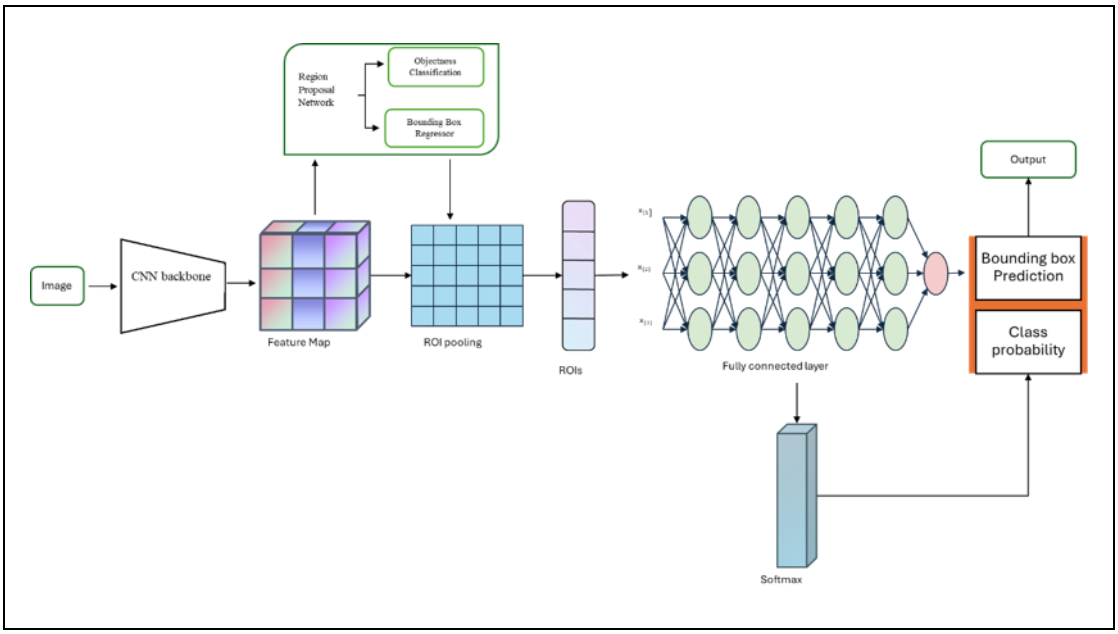



Figure 2.1 Schematic Architecture of the Faster R-CNN Model

In addition, Faster R-CNN utilizes anchor boxes to handle varying sizes and aspect ratios, which is more computationally efficient and can strengthen object detection and recognition. For example, (Shiu et al., 2023) reported an F1-score of 0.849 and a precision of 0.990 for Faster R-CNN and Mask R-CNN when detecting UAV images of ripe pineapples for yield estimation. Moreover, an advanced Faster R-CNN system for fruit detection integrated both RGB and NIR images, utilizing early and late fusion to enable rapid deployment on novel fruits, achieving state-of-the-art results and system efficiency (Nagaraju et al., 2022). Also, (Wu et al., 2023) highlighted the use of Faster R-CNN for red jujubes, demonstrating improved capabilities in fruit recognition systems and the potential for automating agricultural systems. This potential was emphasized when their robotic vision system used Faster R-CNN to assess fruit quantity and ripeness, thereby automating agricultural activities. Among the various models, Faster R-CNN was particularly successful. (Nugroho et al., 2022) documented its use, along with other deep learning models, for detecting tomato ripeness. It achieved an accuracy of 99.55%, demonstrating the model's proficiency in agricultural image classification. (J. Il Kim et al., 2022) cited the use of Faster R-CNN in estimating plum fruits for the first time, further establishing its applicability in agriculture. (Pan & Ahamed, 2022) noted that Faster R-CNN demonstrated the ability of certain architectures to address the challenges of occlusion and complex backgrounds. This was demonstrated by solving the problem of recognizing different fruits in an orchard.

### 2.5.2 Single Shoot Multibox Detector (SSD)

SSD has gained prominence in the field of object detection, primarily owing to its effectiveness and stability. The effectiveness of the SSD is attributable to its Fully Convolutional Network design, as shown in Figure 2.2, and to its frequent collaborations with well-established base networks, such as VGG or ResNet, to extract multi-scale feature maps from input images. The extraction of multi-scale features from the SSD enhances its ability to detect objects of different sizes within images. SSD is also a pioneer in detecting multiple objects in a single image by leveraging different feature maps at various levels of the network hierarchy to predict each object's category and corresponding bounding box. The SSD generates multiple default anchor boxes with different aspect ratios and scales at each location of the feature map, allowing it to detect objects of varying sizes and fit objects more closely within the boxes. The network predicts offsets for the default boxes and assigns a confidence score to each box with respect to the class, using a loss function that encompasses both the classification and localization losses (Huo et al., 2023).

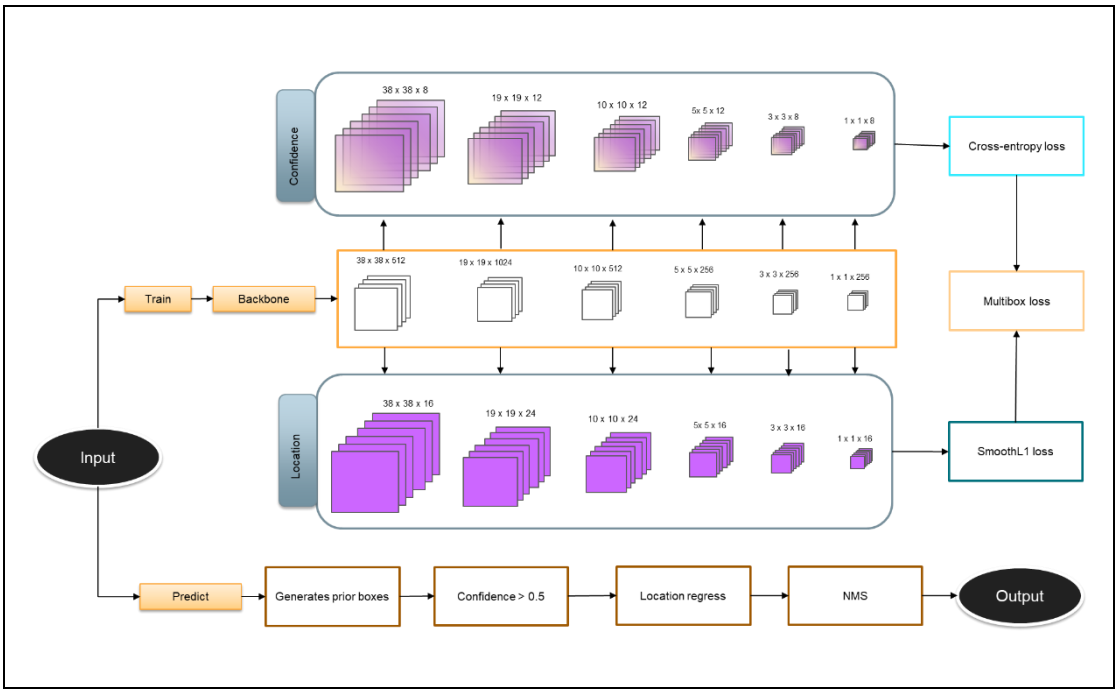



Figure 2.2 The Model Structure of an SSD Consists of Training and Prediction Networks

This integrated method enables SSD to seamlessly merge the benefits of localization and object classification, distinguishing it from two-stage detectors, such as Faster R-CNN, which require region proposal extraction followed by classification (Jiang & Wang, 2020). SSD excels in the trade-off between detection speed and

accuracy. For instance, improvements to the SSD model continue to emerge, including changes that enhance its accuracy in detecting smaller objects and overall detection performance (Huo et al., 2023; Shen et al., 2023). The algorithm has demonstrated potential in various sectors, including agriculture, where the SSD has been employed to identify factors influencing soybean growth, such as flowers and seedpods, as part of smart agriculture. (Zhu et al., 2022) addressed the ability to detect soybean flowers and seedpods accurately, and recorded F1-score values of 0.646 and 0.586, respectively. Subsequently, (Magalhães et al., 2021) described the development and evaluation of two deep learning models, SSD and YOLO, designed to detect green and premature reddish tomatoes in a greenhouse, aimed at the automated harvesting of tomatoes. Furthermore, (J. Il Kim et al., 2022) noted that SSD, as a single-stage detector, can provide both localization and classification simultaneously, which is critical in situations where many items must be identified and detected quickly. This is particularly advantageous in agricultural contexts, where numerous fruits and vegetables may be present in a single image. Additionally, (Alibabaei et al., 2022) examined the real-time detection capabilities of SSD for localizing vine trunks, which is important for precision agriculture. The results indicated that SSD can effectively detect certain agricultural features, establishing its potential for other agricultural detection tasks. Moreover, (Y. Li et al., 2023) suggested that improvements to the SSD architecture would lead to more accurate detection, which is essential for classifying fruits and vegetables of varying sizes, shapes, and colours.

### 2.5.3 YOLOv8

The model architecture in the YOLO family continues to enhance the performance of simultaneous object detection and instance segmentation. According to (Terven et al., 2023), as illustrated in Figure 2.3, the YOLO version 8 architecture improves detection efficiency by directly predicting bounding boxes and class probabilities without the assistance of a region proposal network, thereby further improving the efficiency of the object detection system. YOLOv8's breakthrough includes an anchor-free, center-based detection method. In addition, Pseudo Ensemble or Pseudo Supervision (PS) was used within YOLOv8. This method involves training multiple models, each configured differently and trained on the same dataset, thereby capturing a wider range of the prediction space and resulting in improved prediction

accuracy with fewer correlated errors in the ensemble (Qin et al., 2025). Moreover, the YOLOv8 architecture was the first to utilize Darknet-53, a 53-layer deep convolutional neural network (Vu et al., 2024). One notable architectural change in YOLOv8 is the replacement of the C3 module with the C2F module. C3, the convolutional module, is a stack of convolutional operations applied to the inputs. Compared with the previous model, the C2F module, the updated version of the C3 Module has improved accuracy and processing speed (B. Xiao et al., 2024). In addition, YOLOv8 uses a 3 × 3 convolutional layer, which is more efficient and has fewer parameters than the 6 × 6 convolutional layer (Conv) in its backbone. YOLOv8 also has a decoupled head that separates the prediction of whether an object is present from its type, which can improve efficiency and accuracy. These improvements make YOLOv8 suitable for computer vision tasks that require both object detection and instance segmentation (Jia et al., 2024).

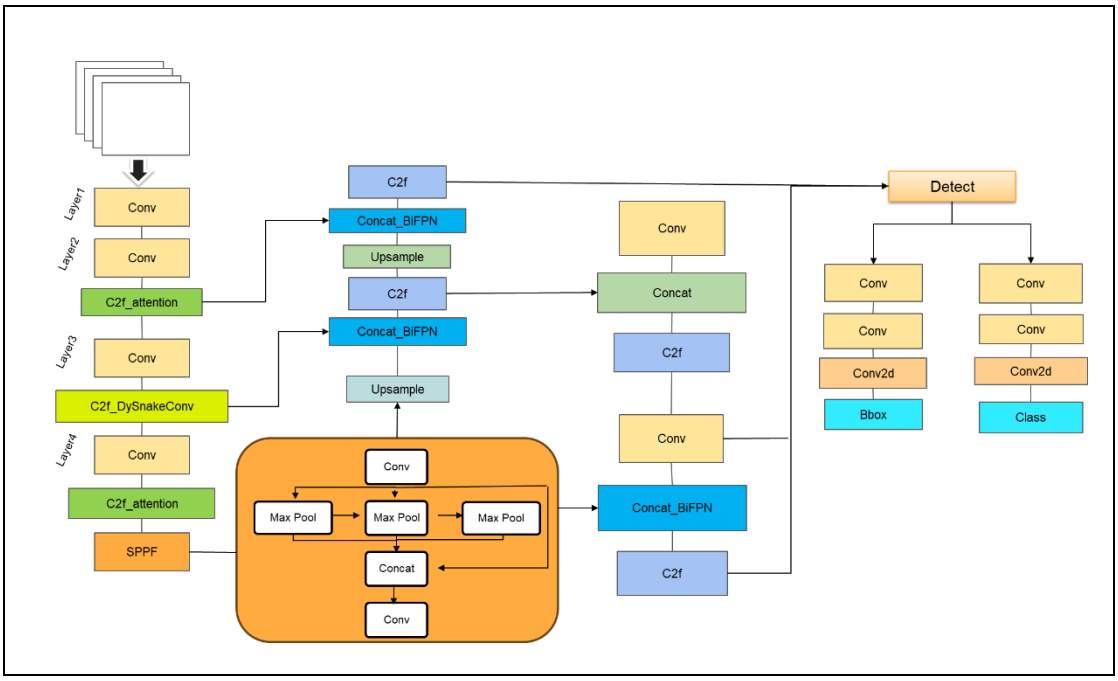



Figure 2.3 The YOLOv8 Architecture Showcases Its Innovative Design for Object Detection and Classification

YOLOv8 provides various configurations for different computational speeds and accuracies (L. Zhang et al., 2023), including YOLOv8-Tiny, which is less accurate but much faster, and therefore, better suited for real-time applications in resource-constrained environments; YOLOv8-Small, which has better accuracy than YOLOv8-Tiny but at a trade-off of a decrease in processing speed, making it better suited for applications that require detailed detection; YOLOv8-Standard, which provides good all-around performance on all settings; and YOLOv8-Large, which is best for high

detail-critical applications where accuracy and precision are vital, because it functions at a large scale. The latest YOLOv8 advancements have allowed its use in a wide range of agricultural tasks and in resolving specific issues related to different agricultural environments, as reviewed in the earlier sections of this thesis, and it is the best available alternative for processing low-level features to obtain early warning signs of crop-damaging pests and diseases (Ren et al., 2023). For instance, more refined versions of YOLOv8 have aided in the early detection and control of plant diseases in greenhouse vegetable production (Liu & Wang, 2024). Other advancements include the use of attention mechanisms to enhance the object detection capabilities of YOLOv8, as tested for tomato detection in complex agricultural scenarios (X. Li et al., 2025). (Lin et al., 2023) demonstrated that the YOLO-tobacco network has the advantages of high detection accuracy and quick detection speed. (D. Li et al., 2020) demonstrated that a deep learning-based video detection architecture with a bespoke backbone system is more suitable for untrained rice movies than YOLOv3 for plant disease and insect detection.

### 2.5.4 Analysis of Deep Convolutional Methods in Object Detection

It is important to review previous works on deep learning models to understand how they detect objects in agriculture. Many researchers have used different convolutional neural networks, including both single-stage and two-stage detectors, to solve crop identification problems. Therefore, Table 2.3 summarizes prior studies that used Faster R-CNN, SSD, and YOLO models. This table highlights their specific techniques, strengths, limitations, and performance metrics, which helps to identify the research gap for this pineapple detection study.

Based on the summary in Table 2.3, most previous studies faced challenges in detecting crops under changing lighting conditions and in the presence of thick leaf backgrounds. Furthermore, many models, such as Faster R-CNN and SSD, require large amounts of memory and struggle to detect very small objects. Even though newer models like YOLOv8 offer better speed and accuracy, their performance on high-altitude drone images with cluttered backgrounds remains underexplored. Therefore, this study aims to address these gaps by testing the models using a specific image-slicing method on UAV data.

Table 2.1
Deep Learning based on Previous Work for Object Detection Using Deep Learning Models

| References | Application Context | Method | Strength | Limitation | Key Findings |
|---|---|---|---|---|---|
| Tarkan Abdullah et al., 2022 | Fruit localization for robotic harvesting | Faster R-CNN | Utilized a VGG-16 backbone for robust spatial feature extraction in dense orchards. | High localization errors occur when overlapping leaves heavily shade target fruits. | Achieved an mAP of 85.4%. The system successfully guided the robotic arm but required a high processing time per frame. |
| An et al., 2022 | Apple detection via ground-based vehicles | Faster R-CNN | Implemented a modified Region Proposal Network (RPN) to filter non-crop background noise. | High computational complexity and memory overhead due to the dense fully connected layers. | Reported an F1-score of 0.891 and a precision of 91.2%. Concluded that the model effectively counts fruits but demands high-end GPUs. |
| Warin et al., 2024 | In-field weed and crop discrimination | Faster R-CNN | Enhanced real-time pipelines by optimizing anchor box aspect ratios specifically for small target weeds. | Severe performance drop when applied to unseen datasets or different weed growth stages. | Achieved a classification Accuracy of 88.5%. The study successfully mapped weed distribution but lacked generalizability across environments. |
| Y Xiao et al., 2020 | Target tracking in complex terrain | Faster R-CNN | Introduced a feature pyramid network (FPN) integration to handle scale variations in challenging environments. | Struggled significantly when handling deformed, rotating, or camouflaged target boundaries. | Recorded an mAP of 81.3%. Concluded that dynamic geometric transformations still present a major challenge for 2-stage networks. |

| References | Application Context | Method | Strength | Limitation | Key Findings |
|---|---|---|---|---|---|
| Nasution et al., 2023 | Multi-attribute crop profiling | Faster R-CNN | Intertwined multi-task learning heads within a single backbone to predict both crop type and growth stage. | Failed to achieve precise 3D visualization and depth tracking due to 2D image limitations. | Achieved an average Accuracy of 86.7%. Proved that multi-task learning saves training time but requires highly comprehensive datasets. |
| Shiu et al., 2023 | UAV images of ripe pineapples for yield estimation | Faster R-CNN and Mask R-CNN | Combined bounding box regression with pixel-level instance segmentation for ripe fruit identification. | Highly prone to false positives caused by specular sun glints and midday reflections on pineapple leaves. | Reported an F1-score of 0.849 and a Precision of 0.990. Successfully automated pineapple counting for regional yield estimation. |
| Nagaraju et al., 2022 | In-field crop disease monitoring | Faster R-CNN | Fused multimodal RGB and Near-Infrared (NIR) imagery to extract sub-surface leaf pixel attributes. | Suffers from drastic accuracy degradation under dynamic natural illumination fluctuations and heavy occlusion. | Achieved a disease detection mAP of 83.5%. Fusing NIR data drastically reduced false negatives compared to using RGB alone. |
| Wu et al., 2023 | Automated fruit recognition system | Faster R-CNN | Refined the RoI pooling mechanism to enhance structural feature mapping for target recognition. | Heavily restricted by a very small sample size and trained exclusively on static images from identical camera angles. | Achieved a Precision of 92.4%. Demonstrated that structural refinements improve localization, but generalization remains weak. |
| Nugroho et al., 2022 | Automated plant canopy localization | Faster R-CNN | Utilized pretrained ImageNet weights to execute rapid transfer learning for high localization accuracy. | Highly dependent on manual bounding box annotations, which are time-consuming and subjective. | Reported an mAP of 87.2%. Concluded that deep learning significantly outperforms traditional manual crop counting visual |

| References | Application Context | Method | Strength | Limitation | Key Findings |
|---|---|---|---|---|---|
| | | | | | methods. |
| Pan & Ahmad, 2022 | Canopy detection under dense occlusion | Faster R-CNN | Developed an occlusion-resistant feature extraction block to handle overlapping crop branches. | The Mask R-CNN branch's slow inference speed makes it entirely unsuitable for high-speed edge deployment. | Achieved an F1-score of 0.855. Successfully solved partial occlusion gaps but identified processing latency as a major hurdle. |
| Huo et al., 2023 | Kiwi fruit detection for automated sorting | SSD | Predicted coordinate offsets using multi-scale default anchor boxes to bypass the region proposal stage completely. | Missed several target fruits when tested under unexpected variations like rainy or heavily overcast weather. | Achieved an mAP of 82.1% at a fast processing speed of 45 FPS, making it highly viable for real-time sorting lines. |
| Shen et al., 2023 | Small object feature enhancement | SSD | Reconfigured early convolutional layers to preserve high-resolution spatial feature maps for tiny targets. | Suffers from native low-resolution inputs and exhibits a longer post-processing (NMS) delay during inference. | Achieved an mAP of 83.8% for small targets. Successfully enhanced small feature retention, but at the cost of a slight delay. |

| References | Application Context | Method | Strength | Limitation | Key Findings |
|---|---|---|---|---|---|
| Zhu et al., 2022 | Tiny crop pest detection | SSD | Implemented a shallow feature fusion network to capture fine-grained pixel details of tiny objects. | Unable to formulate automated causal judgments regarding misclassifications without human-assisted observation. | Reported a mAP of 80.5%. Proved that single-stage networks can detect tiny objects if early spatial resolutions are preserved. |
| Magalhães et al., 2021 | Robotic fruit harvesting automation | SSD | Deployed a lightweight SSD variant directly onto an embedded mobile robotic hardware pipeline. | High architectural susceptibility to overfitting, leading to a severe performance gap between the validation and test sets. | Achieved an mAP of 84.0%. The model successfully automated harvesting maneuvers but required strict regularization to stop overfitting. |
| J. Il Kim et al., 2022 | Real-time multi-class crop sorting | SSD | Structured a single-network pass to output object bounding locations and multi-class probabilities simultaneously. | Failed to classify distinct plant subspecies due to a critical lack of diverse, representative images in the training pool. | Achieved a classification Accuracy of 89.1%. Proved that single-pass networks maximize classification throughput on conveyor belts. |
| Alibabaei et al., 2022 | Embedded smart farming edge devices | SSD | Optimized network weights via post-training quantization to enable real-time CPU-level edge inference. | Completely failed to detect targets under unseen environmental conditions; highly dependent on massive training data. | Achieved an mAP of 79.8% but achieved an ultra-low latency of 12 ms. Concluded that efficiency was prioritized over peak accuracy. |
| Y. Li et al., 2023 | Multi-size vegetable | SSD | Accurate detection in the classification of different | Performance drops significantly when dealing with low-resolution | Reported an overall mAP of 86.3%. Concluded that multi-scale |

| References | Application Context | Method | Strength | Limitation | Key Findings |
|---|---|---|---|---|---|
| | classification | | sizes, shapes, and colors of fruits and vegetables. | or low-contrast agricultural imagery. | receptive fields successfully accommodate natural biological shape variations. |
| Terven et al., 2023 | High-throughput field survey mapping | YOLOv8 | Deployed an anchor-free detection head to directly regress target center points, vastly reducing architectural latency. | Struggled to bound targets with extreme, irregular aspect ratios that were absent from the training anchor distribution. | Recorded a high mAP50 of 92.3%. Proven to provide an excellent operational trade-off between real-time speed and precision. |
| Vu et al., 2024 | Acoustic microscopy target recognition | YOLOv8 | Integrated specialized deep convolutional layers optimized for high-frequency sub-surface feature mapping. | Evaluated exclusively on acoustic microscopy images, limiting its proven validity on standard optical UAV datasets. | Achieved a detection Accuracy of 91.5%. Demonstrated that YOLOv8's backbone is highly adaptable to non-optical imaging modalities. |
| Jia et al., 2024 | Automated canopy boundary segmentation | YOLOv8 | Leveraged dual-task feature fusion paths to simultaneously optimize bounding boxes and semantic masks. | Detection performance on heavily occluded or microscopic target structures still requires further enhancement. | Achieved an mAP of 93.1% and a mIoU of 84.6%. Successfully established a state-of-the-art benchmark for automated crop tracking. |
| Ren et al., 2023 | Early-stage plant disease diagnostic | YOLO-GEW | Modified the standard YOLO framework with localized attention mechanisms for | Demands highly sophisticated camera hardware and massive dataset diversity to achieve field- | Reported a disease detection mAP of 88.9%. Proved that attention mechanisms are vital to catch |

| References | Application Context | Method | Strength | Limitation | Key Findings |
|---|---|---|---|---|---|
| | | | early-stage symptom tracking. | level robustness. | subtle, early-stage leaf color changes. |
| Lin et al., 2023 | Single-species crop disease tracking | YOLO-tobacco | Stripped redundant layers from the backbone to maximize inference speed for targeted single-plant disease identification. | Severely constrained by a narrow dataset, limiting its practical utility exclusively to a single crop disease category. | Achieved an exceptional processing speed of 78 FPS with an mAP of 90.2%. Highly successful for high-speed automated drone scouting. |

### 2.5.5 Hyperparameter Settings

There are no simple and easy ways to set hyperparameters. Therefore, setting hyperparameters requires expertise and extensive trial and error. Hyperparameter settings play a crucial role in controlling the learning process and the network architecture. A few model hyperparameters need to be adjusted based on the network architecture. Consequently, each model hyperparameter must be carefully tuned to ensure the optimal performance of the DLM, as it influences the training speed, model accuracy, and generalization to unseen data.

Table 2.2
Summary of Hyperparameter Settings for DLM

| Hyperparameter Settings | Faster R-CNN | YOLO | SSD |
|---|---|---|---|
| Learning rate | It starts at 0.01 and decreases to 0.001 | It starts around 0.001 | It starts between 0.001 and 0.004 |
| Batch size | 1 or 2 images per GPU (always kept small) | Between 8 and 32 (depending on GPU memory) | Ranging from 8 to 32 (depending on GPU memory) |
| Input image sizes | between 224x224 and 448x448 | Range from 320x320 to 608x608 | Set to 300x300 or 512x512 |
| Optimizer | SGD | SGD | SGD |
| Weight decay of optimizer | 0.0001 | 0.0005 | 0.0005 |
| Metric (IoU threshold) | 0 to 1 | 0.4 to 0.6 | Around 0.45 |
| Number of epochs | 400 epochs and above (for large and complex dataset) | 200 to 500 epochs (depending on dataset size) | 60 to 120 epochs |
| Momentum of regularization | 0.9 | 0.937 | 0.9 |
| Augmentation | No aggressive augmentation | Mosaic augmentation (mixed precision) | Minimal augmentation (photometric) |

### 2.5.6 Performance Model Measurement

Recent studies have confirmed the success of Convolutional Neural Networks (CNNs) in crop disease detection and classification. Based on the findings of previous studies, CNNs have performed very well across agricultural applications, particularly in terms of accuracy, precision, recall, and F1 score on the confusion matrix. A CNN-based model recorded an average precision of 92.06%, average recall of 92.71%, and average F1 score of 89%, thus verifying the proficiency of the CNN architecture in agricultural disease classification (Alghamdi & Turki, 2023). Next, an average precision, average recall, and average F1 score of over 89% for the classification of diseases in soybean leaves were recorded for the CNN approach in federated learning (Vats et al., 2024). A hybrid model comprising a convolutional neural network (CNN) and a support vector machine (CNN-SVM) also recorded high-performance metrics in the detection of several different plant diseases, thus underscoring the prowess of CNN in practice (Goumiri et al., 2023). Similarly, the accuracy, recall, and F1 score for predicting diseases in the model were approximately 91% for the disease known as Fusarium Wilt (A. Kaur et al., 2023).

Furthermore, in the field of plant disease classification, transfer learning based on the VGG CNN model has also yielded excellent results. The use of pretrained models in agricultural domains has shown versatility, with 98.40% and 95.71% accuracy on grape and tomato leaf diseases, respectively (Gafurov, 2023). The architecture of VGG models has been the most optimal and has been reported to perform successfully on different spectra of diseases in different plants, which shows the model's proficiency in recognizing diverse images in the agricultural sector (Cv & Lin, 2022). A prime illustration of this is the VGG model, a CNN-based approach, which excelled at identifying diseased and healthy leaves in tomato and grape crops, achieving high accuracy, F1 score, recall, and precision. The model also successfully classified grape leaf diseases with 98.40% accuracy, 98.43% precision, 98.40% recall, and 98.41% F1-score. Meanwhile, it classified tomato leaf diseases with 95.71% accuracy, 95.82% precision, 95.71% recall, and 95.74% F1-score (Tambe et al., 2024). However, several researchers have emphasized that the adaptability of these models is limited by the quality of the training dataset and the fit of the designed architecture.

## 2.6 Relation of DSSN between CNN Methods

One of the most important advantages of DSSNs is the ability of network design to ensure that the model remains computationally efficient while maintaining an organized spatial slicing of the area of interest. The described model allows for versatility in complex tasks, such as agricultural imaging, owing to its ability to manage large datasets (Tong & Wan, 2023). In the DSSN architecture, CNNs are the most common because of their ability to analyze spatial information and extract different features from complex datasets. Attentional focus or graph-based approaches are incorporated to capture complex contextual dependencies between slices. The DSSN aims to improve the trade-off between performance and the ability to analyze small regions of an image efficiently. In complex tasks, such as image segmentation and classification, the hierarchical structure of CNNs allows for the capture of spatial features at different levels. In addition, CNNs are very important because they can learn image features through detection and do not require the slow, tedious process of feature engineering. Integrating CNNs within a DSSN not only improves the quality of image processing tasks but also that of agricultural image classification and detection, and is a testament to an efficient way of handling complex tasks.

Combination DSSN with CNN methods, such as Faster R-CNN, YOLO, and SSD, can produce highly efficient detection and classification systems. All three systems have their respective strengths and can be used as performance boosters when combined. The DSSN is particularly strong in fine-grained pixel-level detection and analysis, enabling it to detect small and complex objects and patterns that are often overlooked by region-based methods. This is a vital strength in detail-critical applications, such as medical diagnostics and satellite imagery analysis, where accurate classification depends on the detection of small features (Ma et al., 2023). However, the need for detailed pixel-level processing can make DSSN significantly more expensive and slower than conventional systems.

In contrast, Faster R-CNN, YOLO, and SSD are optimized for object detection and classification across several scales more efficiently. The combination of DSSN and these established approaches could yield a hybrid detection system that leverages the best of both. For example, Faster R-CNN can detect small objects and complex patterns, making it suitable for integration with DSSNs developed for detailed analyses. SSD is also effective at detecting small objects, primarily due to its ability to leverage feature

maps at multiple scales (X. Zhang et al., 2023). The combination of SSD and DSSN effectively leverages multi-scale contextual information to boost object detection, particularly in difficult and cluttered scenes, thereby enhancing performance across various tasks (Oghaz et al., 2022). Furthermore, owing to advanced feature fusion methods, YOLO achieves good detection rates, especially when combined with DSSN, which can detect small objects of interest.

## 2.7 Contribution of the DLM to the Carbon Dioxide

Throughout the 21st century, the threat of exceeding global temperature increases of 1.5 °C and 2 °C worsens with the lack of reduction in carbon dioxide and other greenhouse gas emissions (G. Kim et al., 2023). The IPCC has noted the potential to surpass temperature thresholds if greenhouse gas emissions remain unchanged. One concern for the public, as well as the deep learning community, is the carbon dioxide emissions from deep learning training. Carbon dioxide emissions arise from data centre overheads and the operating energy costs of deep-learning hardware (Strubell et al., 2020). The enormous advances in deep learning algorithms have resulted in an increasing availability of vast datasets and unfettered access to high-powered computers. The computational power required for deep learning algorithms has been shown to double approximately every three to six months (Jelinek et al., 2024). The capacity and rate of technological advancement directly influence energy demand. Although the success and proliferation of deep learning are global benefits, its increased energy demand is a potential driver of climate change. Thus, new models for green algorithms focus on reducing emissions and improving efficiency.

Optimizing energy efficiency and mitigating emissions from deep learning processes have shifted attention toward designing green algorithms. These advancements ensure that progress in Artificial Intelligence does not compromise sustainability in other areas. By focusing on optimizing resource use and process efficiency, the deep learning community can achieve greater decarbonization (Gupta et al., 2022). In agricultural vision tasks, the carbon footprint becomes a real issue due to the continuous processing of large, high-resolution crop images (Strubell & Mccallum, 2019). Previous literature shows that training deep learning models on heavy datasets can emit significant amounts of carbon dioxide depending on the model's runtime. Two-stage models like Faster R-CNN have a slower runtime, often taking around 120-150

ms per frame, thereby consuming more GPU energy (Patterson et al., n.d.). On the other hand, single-stage models like SSD and YOLOv8 are much faster, with runtimes of around 30 ms and 12 ms per frame (Patterson et al., n.d.). A faster runtime directly reduces the operational energy required, which helps lower the total carbon emissions during model deployment (Schwartz et al., 2017). In addition, the importance of Artificial Intelligence in optimizing hardware performance is increasing. Deep learning tasks require more computational time; techniques that reduce the size of binary code-level weights have the potential to decarbonize by reducing the computational time required to perform deep learning tasks (Jelinek et al., 2024).

The contribution of AI to decarbonization and hardware performance through binary code-level weight reduction is the compression of network architectures that effect to decarbonization. In summary, the evidence overwhelmingly demonstrates that without significant reductions in carbon dioxide and other greenhouse gas emissions, the world's temperature is more likely to exceed critical thresholds in the near future. Awareness of the deep learning community and proactive measures to address its carbon footprint are crucial for achieving a sustainable future. This energy and carbon issue is the main reason this study is included in my research. Since this project involves tracking pineapples using high-resolution UAV drone data, an unoptimized model will lead to higher electricity consumption and a larger carbon footprint. Recent findings show that optimizing network architectures or choosing lightweight single-stage models can decarbonize the deep learning process by reducing energy consumption by up to 30% to 50% (Schwartz et al., 2017). Therefore, this section is included to justify my approach. By analyzing the performance and runtime of Faster R-CNN, SSD, and YOLOv8, this study can identify a balanced solution that achieves high detection accuracy while minimizing the carbon footprint.

# CHAPTER 3
# RESEARCH METHODOLOGY

## 3.1 Introduction

This proposed study will involve collecting images and videos using a UAV. In this section, the overall methodology is discussed. First, it will collect images and videos at different heights above the ground. Then, images from different height levels from the ground, which are 3m to 10m, will be analyzed, and datasets will be prepared. The CNN model was used to classify the pineapple and the background, thereby obtaining the pineapple's value in the image. Then, the decarbonization algorithm was used to estimate the effect on the carbon footprint. Figure 3.1 depicts the overall process, and Figure 3.2 shows the main workflow of the model diagram.

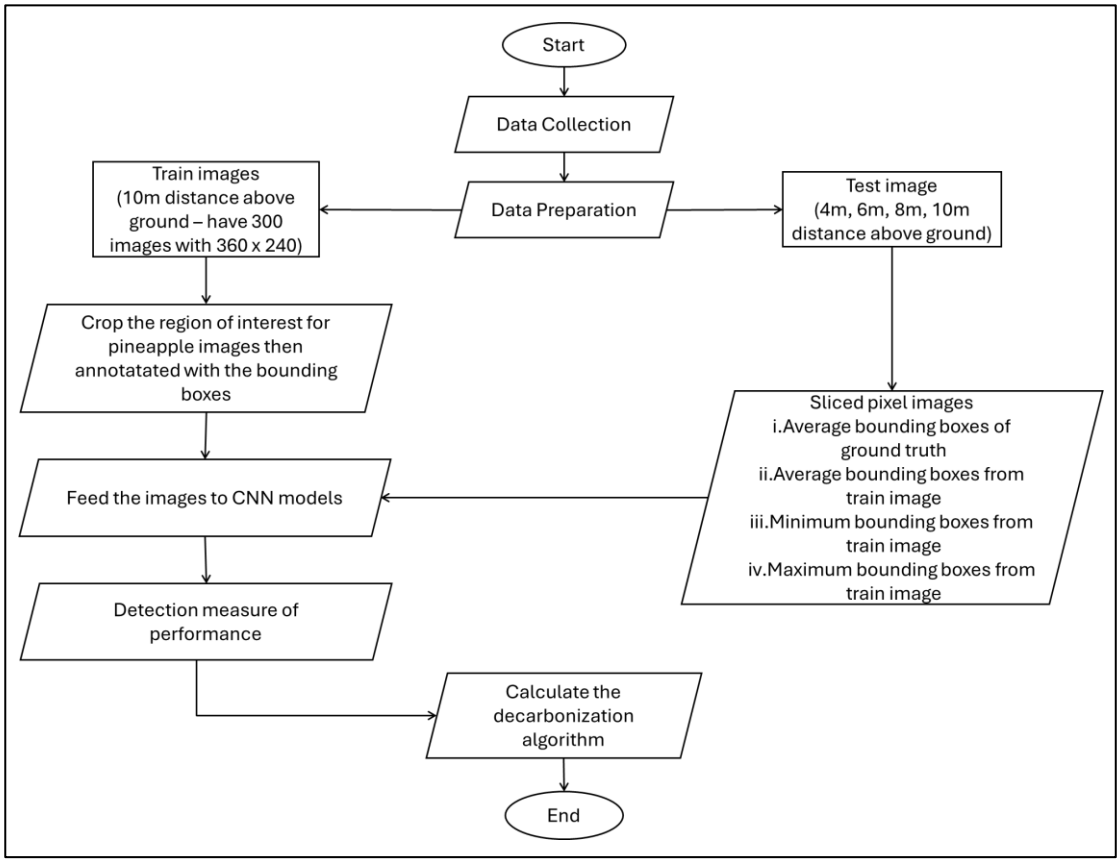



Figure 3.1 General Methodology of Research

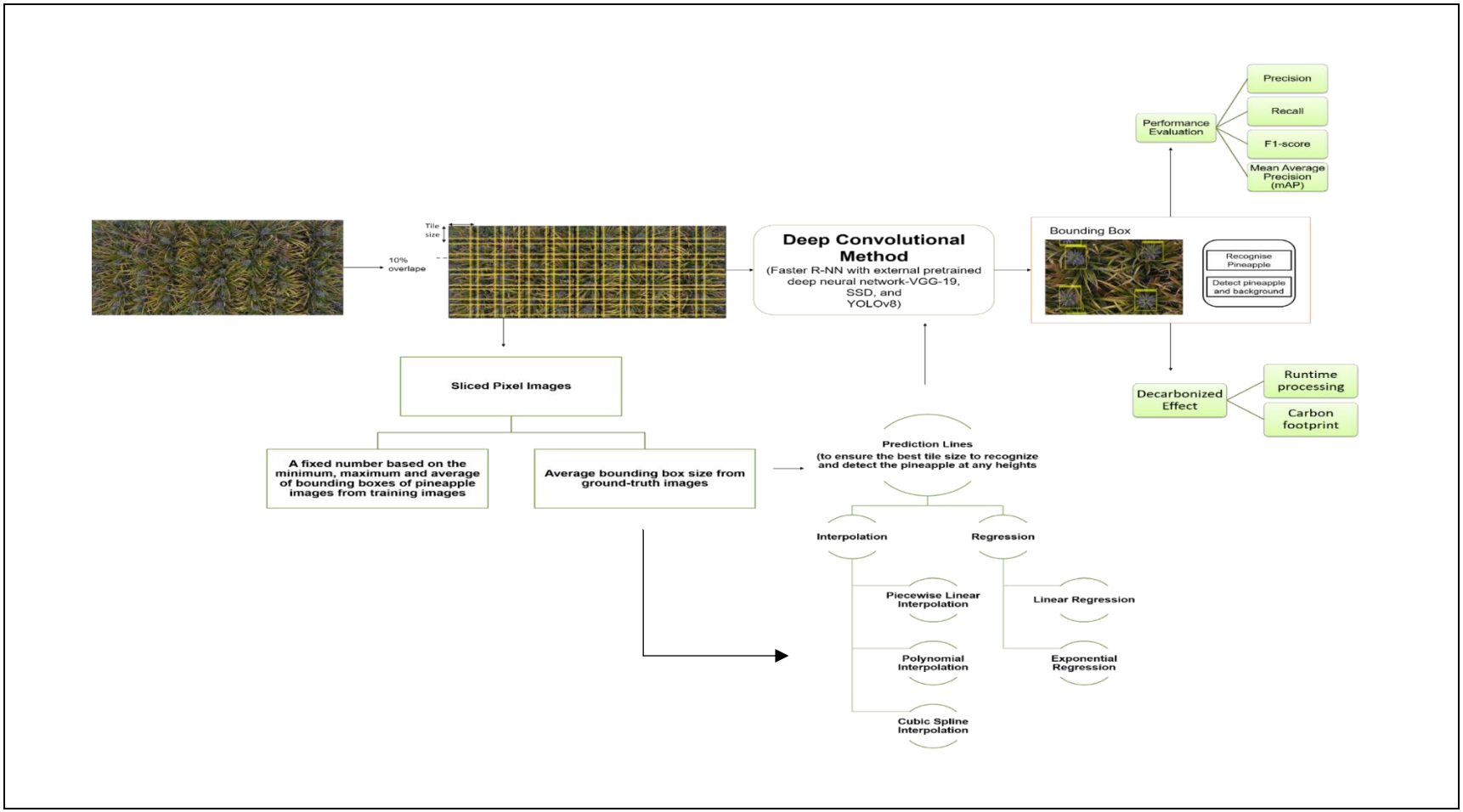



Figure 3.2 The Main Flow Model Diagram

## 3.2 Data Collection

The field data collection for this study was conducted through direct observations at the Simpang Renggam Pineapple Plantation, located in the Kluang district of Johor, Malaysia. The aerial mapping and data acquisition were successfully carried out in March 2019 under clear, sunny conditions to ensure optimal natural daylight illumination. Aerial videos and images were captured using a DJI Phantom 3 Advanced quadcopter equipped with a 3-axis gimbal stabilizer and a 1/2.3-inch backlit Sony CMOS digital camera sensor. This setup featured a primary color filter with 12.4 Megapixels of effective resolution, allowing for the acquisition of high-quality 4K RGB remote sensing data. During the flight missions, the UAV was preprogrammed to cruise automatically along the designated geographic rows of the pineapple crop at a constant speed of 16 m/s. Continuous high-resolution aerial videos were recorded at multiple altitudes ranging from 3 m to 10 m above the ground level, with increments of 1 m. At each specific flight height, the video recording duration was maintained for approximately 3 to 5 minutes at a standard frame rate of 30 frames per second (FPS).

Since deep convolutional models require individual static images for object recognition and detection rather than continuous video streams, a systematic video-to-image frame extraction process was executed. The raw 4K videos were converted into individual static image frames at a sampling rate of 1 frame per second (1 Hz). This sampling rate was chosen to eliminate redundant, highly overlapping consecutive frames, thereby ensuring high spatial diversity across the crop fields. To ensure robust pineapple recognition and detection performance in dense, complex agricultural backgrounds, the extracted image frames were meticulously curated to meet strict quality criteria. Firstly, frames affected by motion blur, which typically occurred due to wind resistance or sudden banking movements of the UAV at 16 m/s, were strictly discarded to preserve textural sharpness. Secondly, only frames with balanced daylight exposure were selected, deliberately avoiding severe cloud shadows or direct sun glare that could obscure crop features. Finally, the selected images had to exhibit high structural visibility, where the defining morphological features of the pineapples, such as the crown shape, fruit body, and distinct leaf patterns, were clearly visible against the surrounding weeds and soil, thus providing high-quality regions of interest (ROIs) for the deep convolutional networks.

## 3.3 Data Preparation

The data preparation process was divided into two parts: training and test data. During training, the CNN models learn from these images to recognize patterns and features that distinguish categories, such as pineapple and background. A test image was used to evaluate the performance of the combination of the DSSN and DCM. The sample images in the test and training sets were different.

### 3.3.1 Train Data

The training dataset comprises 300 images captured by a UAV at a consistent altitude of 10 meters. Each image maintained fixed dimensions of 300 cm × 240 cm. The process to generate training images is shown in Figure 3.3.

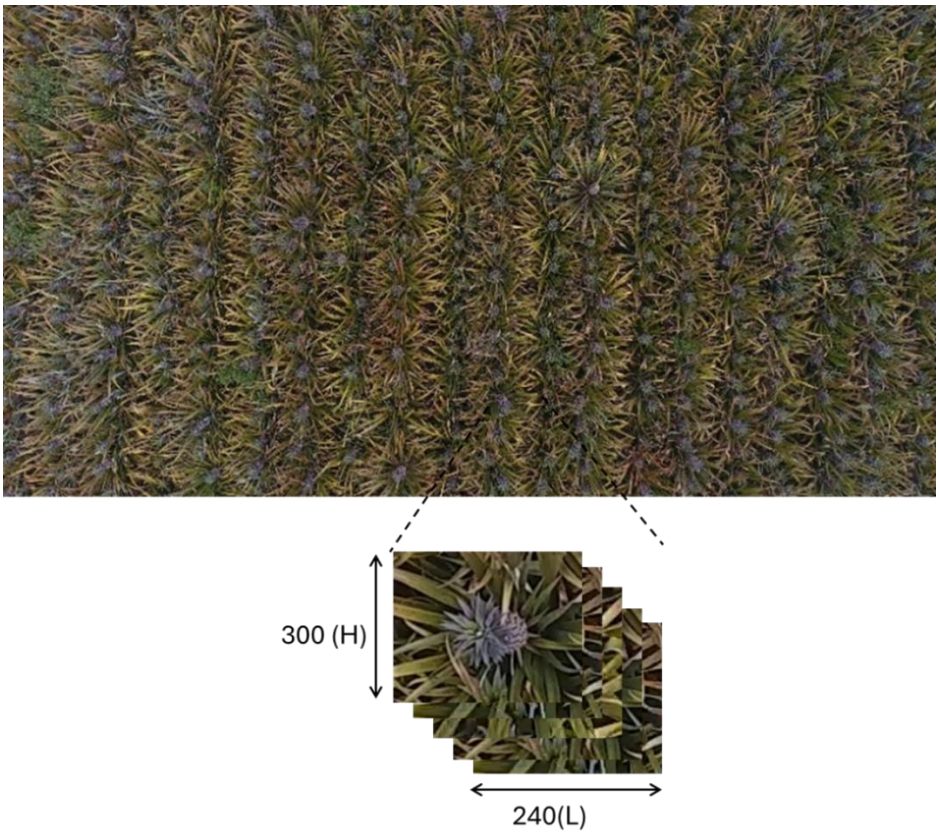



Figure 3.3 Cropped Pineapple for Training Images

This uniformity in image size and capture height ensured consistent data, which is crucial for developing a robust machine learning model that generalizes and accurately detects pineapples in real-world in-field plantations. Within this standardized image, pineapples are shown in various positions and shapes, reflecting variations in

pineapple appearance and orientation. The training set is a portion of the dataset used to teach the model to distinguish pineapple from the background.

### 3.3.2 Test Data

The test images serve as a benchmark dataset for critically evaluating the performance of the developed deep convolutional models in detecting pineapples in complex, dense agricultural backgrounds. The testing dataset comprised annotated pineapple images with ground-truth bounding boxes derived from four distinct flight heights, ranging from 10 m to 4 m in 2 m increments, as illustrated in Figure 3.4. Specifically, this evaluation setup yielded a total of 25, 92, 234, and 269 annotated pineapple instances distributed across the test images captured at altitudes of 4 m, 6 m, 8 m, and 10 m above the ground level, respectively.

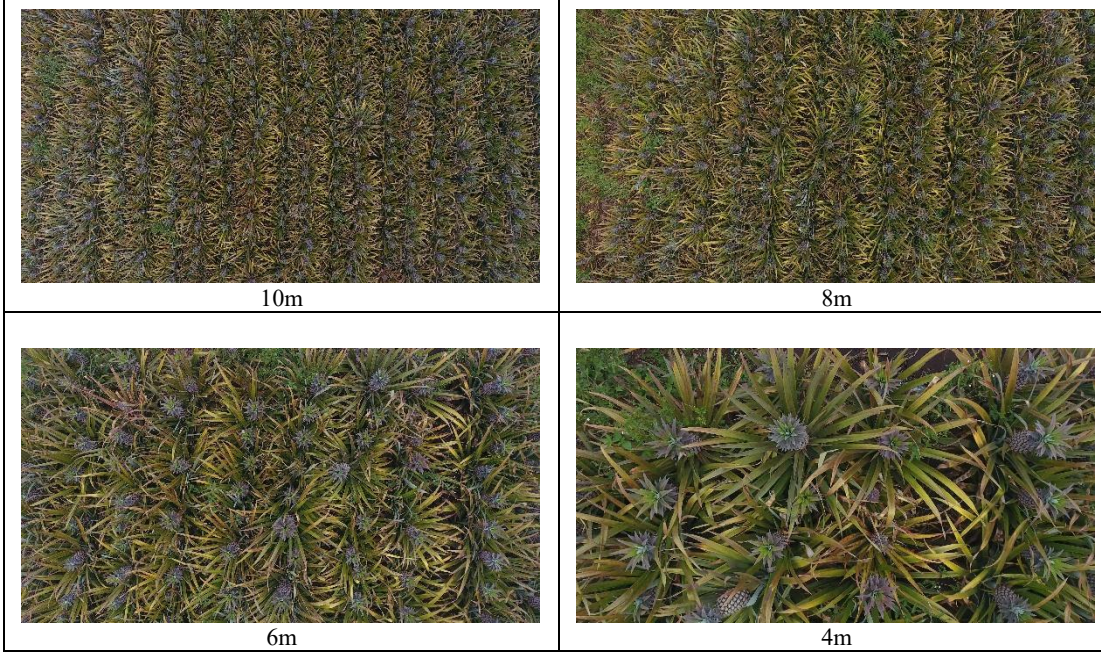



Figure 3.4 Sample Set of Test Images

The significant variation in the number of annotated instances is directly attributed to the altitude-dependent field of view (FOV) of the UAV camera. The FOV expands substantially at higher altitudes (10 m), allowing a much larger surface area and a higher density of pineapple instances to be captured within a single image. Conversely, the restricted FOV at lower altitudes (4 m) captures a smaller ground area, resulting in fewer visible crops. The variation in these numbers across altitudes provides

a robust platform for assessing the models' detection scalability, spatial-resolution tolerance, and precision under varying crop scales.

### 3.3.3 Labeling Data

In this study, the data labeling process was performed on the makesense.ai website. The test images were uploaded, and each pineapple appearing in the images was manually annotated by drawing a bounding box around it. Since multiple pineapples were often present in a single image, each object was individually labeled as “pineapple.” This data labelling step as shown in Figure 3.5 is essential in object detection and classification because it provides the high-quality annotated data required by deep learning models, such as convolutional neural networks (CNNs), enhances the model’s ability to distinguish the target object from background elements, reduces bias and misclassification through consistent annotations, and supports supervised learning by supplying examples that link input images to object categories. After the labeling process was completed, the annotation files containing bounding box and label information for each pineapple were exported to enable the use of CNN-based DSSN methods such as Faster R-CNN, YOLOv8, and SSD for accurate object recognition and detection.

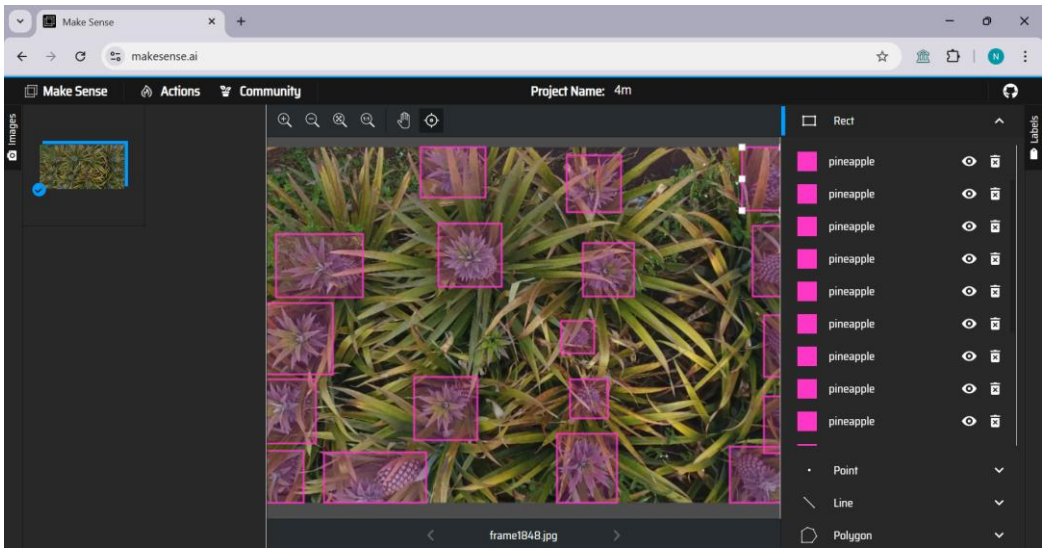



Figure 3.5 Labeling Process of Pineapple Images by Using the makesense.ai Website

## 3.4 Generate Tile Size

The tile size for DSSN must vary based on the appropriate height above the test image (4m to 10m in increments of 2m above the ground). The tile size must be square, meaning the height and width are the same. The test images were sliced based on the calculated tile sizes.

Thus, this study tested several tile sizes to determine the most suitable tile size for DSSN, as shown in Figure 3.6. First, the average bounding box size for each ground truth test image is calculated using Equations (3.1) to (3.3). The bounding boxes were drawn using the makesense.ai website. This website allows annotating images by drawing bounding boxes around pineapples, creating a region of interest (ROI) that defines the object's location.

Once the bounding boxes are drawn, the height ($H$) and width ($W$)for the pineapple images can be measured automatically. The average size of bounding boxes ($\bar{B}$) with the number of bounding box ($n$) is computed as in (3.1).

$$\bar{B} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{W_i + H_i}{2}\right) \tag{3.1}$$

To determine the tile size, considering the widths and heights of all bounding boxes as $W_i = W_1, W_2, ..., W_n$ and $H_i = H_1, H_2, ..., H_n$, then the minimum value between these two sets can be given as $W_{\min} = \min_i W_i$ and $H_{\min} = \min_i H_i$, hence returning the smallest value between the two values of $W_{\min}$ and $H_{\min}$ as a minimum bounding box size $B_{\min}$ which can be given as in (3.2). Meanwhile, the maximum bounding box size $B_{max}$ can be given as in (3.3), computed as the highest bounding box size between the $W_{max} = \max_i W_i$ and $H_{\max} = \max_i H_i$.

$$B_{\min} = \min\left(W_i, H_i\right) \tag{3.2}$$

$$B_{max} = \max\left(W_i, H_i\right) \tag{3.3}$$

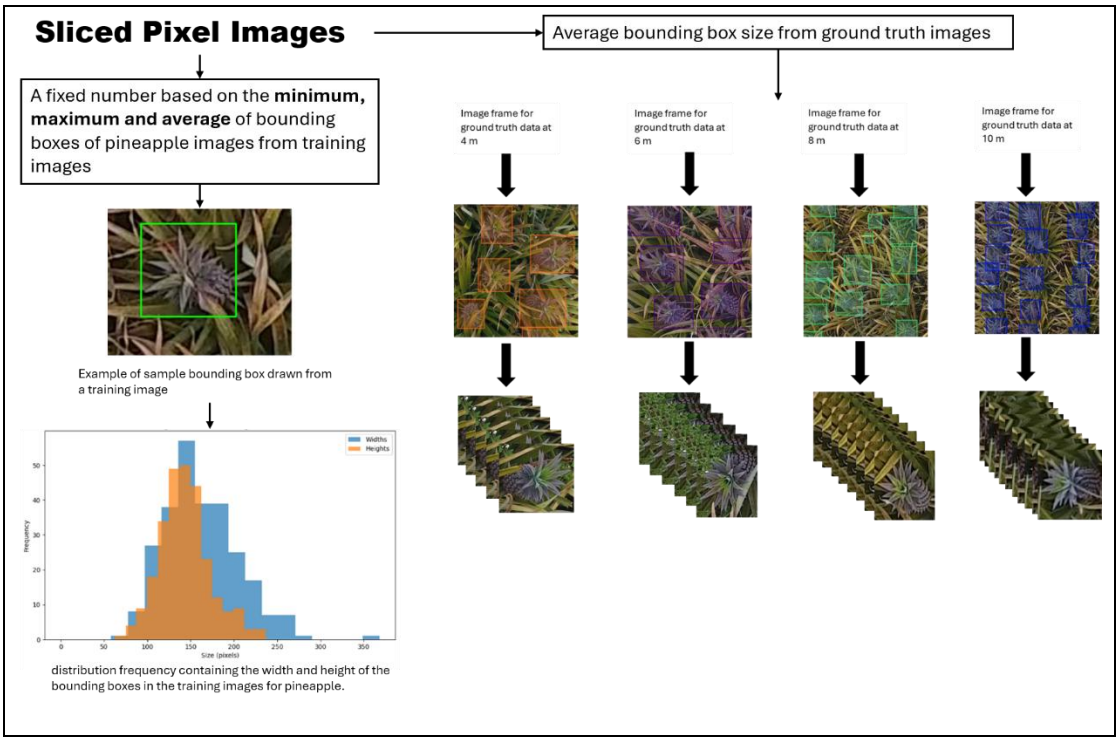



Figure 3.6 Method of Spatial Slicing Images

The other tile size was determined using statistical measures from the training images, yielding a fixed value based on the minimum, maximum, and average bounding box sizes of pineapple images. Based on Figure 3.5, the bounding boxes annotated across the 300 training images were analyzed to plot the frequency distribution of pineapple width and height. From this distribution, the exact statistical values for the minimum, maximum, and average bounding box dimensions were extracted to define the fixed tile sizes. These calculated statistical values serve as baseline parameters for slicing the test images.

### 3.4.1 Apply Tile Size

When processing an image to detect pineapples, as shown in Figure 3.7, the image was divided into smaller slices or patches based on a predefined tile size. To determine this tile size, two approaches were evaluated: first, using the average bounding box size from the ground-truth images; and second, using a fixed value derived from the minimum, maximum, and average bounding box sizes of the training images. Once the optimal tile size was determined from these calculations, it was incorporated into the Python script, allowing the program to slice the images to the specified size automatically.

A 10% overlap was applied to all border sides of each slice to ensure that pineapples located near the edges were not overlooked. This implies that a small portion of the patches was shared between consecutive slices. This overlap is rooted in how image slicing works, where the division occurs at fixed region intervals and does not account for the pineapple's actual boundaries. Consequently, a pineapple can be split across two slices, making it harder to detect, especially if only a small part of the fruit appears in one slice. Therefore, by adding a 10% overlap, we increased the chance that any given pineapple would be fully captured in at least one slice. This significantly improved the detection accuracy, especially for pineapples that would otherwise be at the margins.

An important aspect of overlap is that it avoids duplicate detections. Even if a pineapple appears in two overlapping slices, the system prevents it from being counted twice. This duplication is controlled by setting an Intersection over Union (IoU) threshold during the Non-Maximum Suppression (NMS) stage, where a pineapple detection is retained only if at least 85% of its bounding box area overlaps with another detection. Additionally, confidence thresholding was used as a crucial post-processing step to eliminate bounding boxes with low predictive accuracy. By setting a high confidence threshold of 0.85, only detections with a high degree of certainty were retained. Combined with NMS, which removes redundant overlapping boxes, this workflow ensures that only the most accurate and reliable detections are included in the final output.

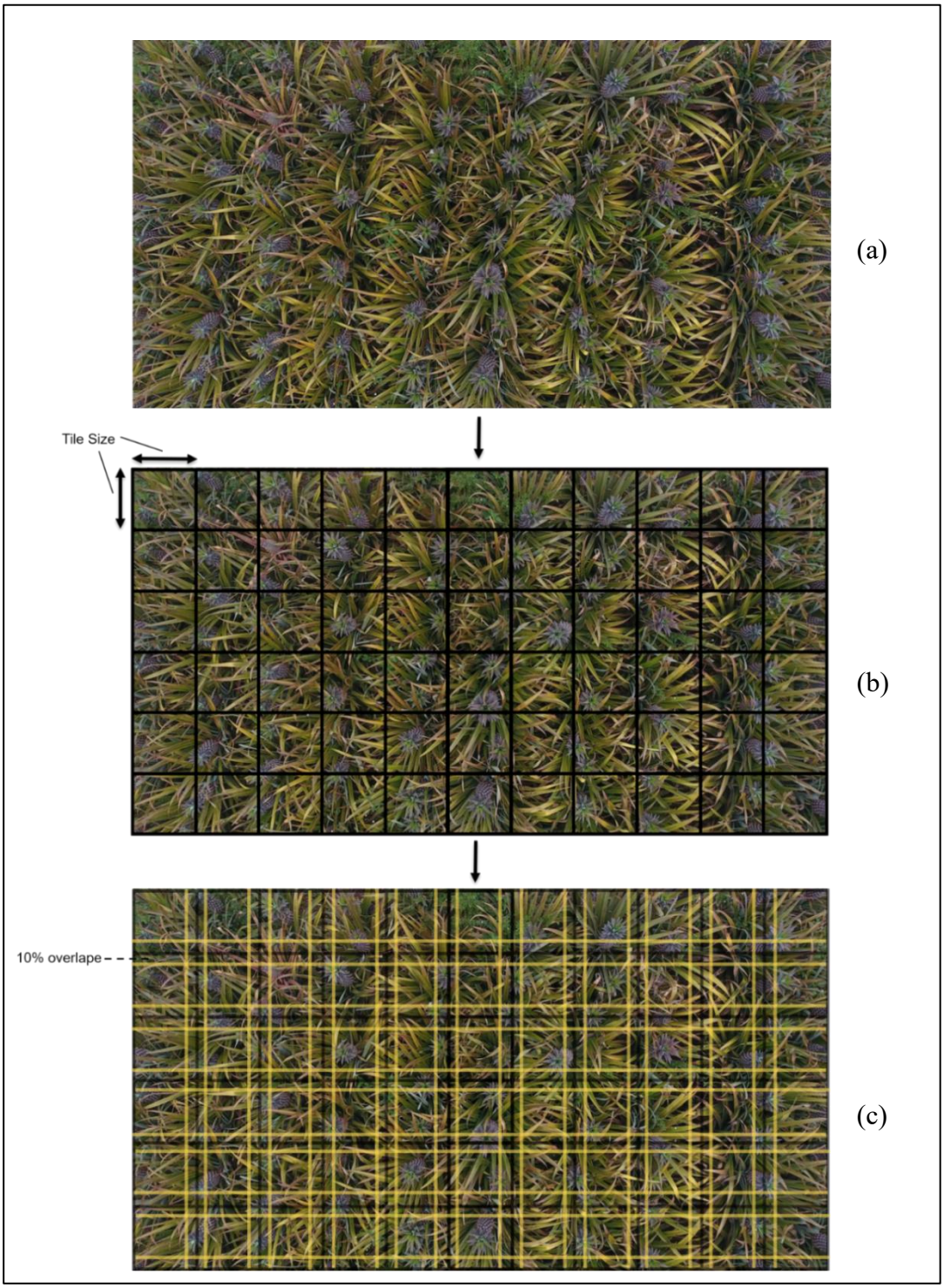



Figure 3.7 Stages of Applying Tile Size on Testing Image: Original Image of Pineapple Crops (a), Standard Tiled Image Without Overlap (b), and the Resultant Overlapping Tiles for Enhanced Detection (c)

## 3.5 Validation from Test Data

The main experiment was designed to examine the best-performing model across various height levels. In this study, the effect of tile size at even height levels was compared to pineapple crop detection that relies solely on annotated pineapple

images. At the point where the best tile size was achieved, a dataset sample for pineapple detection was collected using images captured at odd heights such as 3m, 5m, 7m, and 9m. This strategy anticipates the adaptability of the tile size and assesses the detection consistency across varying distances. Hence, this study analyzed several mathematical techniques designed to estimate unknown data points and reveal the underlying trends to determine an appropriate tile size within a dataset. Figure 3.8 depicts the process of finding the best lines to obtain the prediction value for tile sizes other than the test image. Simultaneously, this application can be used for any image from the ground, except that it has hardware limitations, such as the lens of the UAV cannot capture a clean and sharp image.

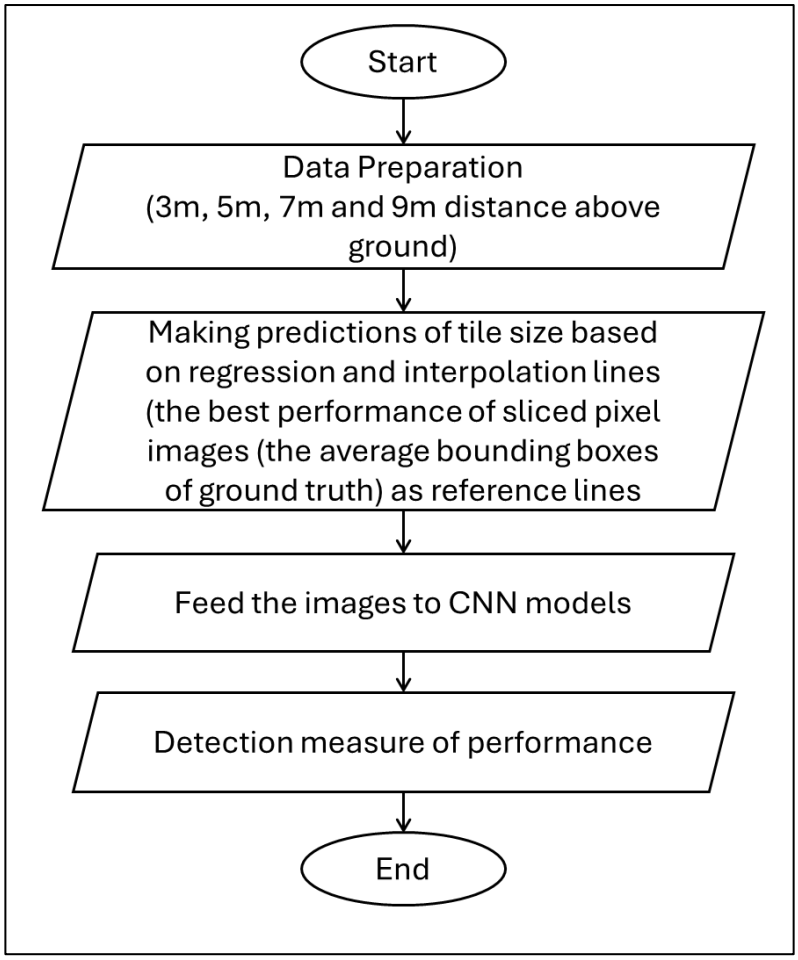



Figure 3.8 Overall Flowchart for Validation from Test Images

The tile sizes for images taken at heights of 4m, 6m, 8m, and 10 m served as references to determine the appropriate tile sizes for validation images captured at heights of 3m, 5m, 7m, and 9m, as shown in Figure 3.9. This approach ensures a

structured, systematic method for adjusting tile sizes as the image scale varies with height above the ground. Using the best tile size based on a performance evaluation as a reference, this study ensures that the tile size remains proportional to the pineapple in the image, regardless of the height at which the image was captured.

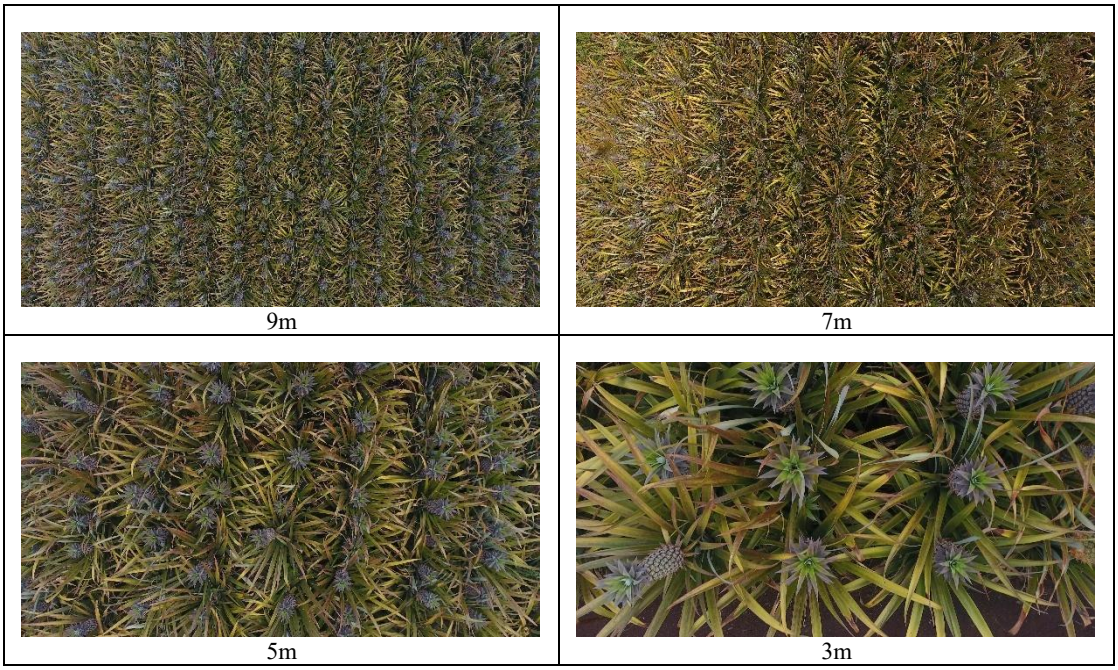



Figure 3.9 Sample Set of Dataset Images for Validation from Test Data

Specifically, this sample set was applied and compared with interpolation and regression methods which are polynomial interpolation, cubic spline interpolation, linear interpolation, linear regression and exponential regression, to assess their effectiveness in reconstructing missing values and modelling nonlinear patterns. Beyond the computational mathematics of each method, its approach is evaluated for accuracy, smoothness, and its ability to adapt to different types of data behavior at other heights in test images. The specific characteristics of the reference data and validation from the test image used in this study are tabulated in Table 3.1.

Table 3.1
Parameters and Derivation Methods for Reference and Validation Datasets Across Different Altitude Levels

| | **Reference Data / Test Image** | **Validation from Test Image** |
|---|---|---|
| Altitude Levels | 4 m, 6 m, 8 m, and 10 m | 3 m, 5 m, 7 m, and 9 m |
| Data Source | Baseline average bounding boxes from ground truth from each height | Sample dataset for testing model adaptability |
| Application | Used as the known data points to | Used to evaluate tile size prediction |

| Purpose | calculate the baseline tile sizes | accuracy and consistency |
|---|---|---|
| Tile Size Derivation | Directly determined via the average of bounding boxes from ground truth testing images | Predicted using interpolation and regression methods (Equations 3.4 – 3.12), but must refer to the reference data line |

Mathematical equations representing all methods for validating the sample image pineapple crop models using odd-level heights are shown in (3.4) to (3.12). The best performance from this method is one of the alternatives that will be applied at other heights to predict the tile size for slicing images and achieving the best results in recognizing and detecting pineapples.

Piecewise linear Interpolation:

$$4 \le x \le 6;\ y = -84x + 674 \tag{3.4}$$

$$6 \le x \le 8;\ y = -35.5x + 383 \tag{3.5}$$

$$8 \le x \le 10;\ y = -4.5x + 135 \tag{3.6}$$

Linear regression:

$$y = -40.75x + 459.5 \tag{3.7}$$

Polynomial interpolation:

$$y = 9.938x^2 - 179.875x + 896.75 \tag{3.8}$$

Cubic spline interpolation:

$$4 \le x \le 6;\ y = 3.142(x-3)^3 - 127.067 + 850.2 \tag{3.9}$$

$$6 \le x \le 8; y = 1.458x^3 + 40.728x^2 - 387.242x + 1340.375 \tag{3.10}$$

$$8 \le x \le 10; y = -1.683x^3 + 45.450x^2 - 420.317x + 1417.550 \tag{3.11}$$

Exponential regression:

$$y = 1013.945e^{-0.281x} \tag{3.12}$$

## 3.6 Model Classifier

### 3.6.1 Pretrained Deep Neural Network

In this study, ResNet-101, ResNet-50, Inception-ResNet-v2, and VGG-19 were implemented as the pretrained external backbones for the Faster R-CNN model. Structurally, Faster R-CNN is a two-stage detector that lacks an inherent feature extractor, meaning it relies entirely on an external network to translate raw input pixels into dense feature maps. Training these deep backbones from scratch on a limited, highly specialized dataset like drone pineapple imagery would overwhelm the network by forcing it to learn basic visual concepts such as edges, lighting, and textures while simultaneously learning complex object detection. By leveraging transfer learning instead, the framework inherits an established understanding of shapes from large-scale datasets, thereby preventing severe overfitting and significantly reducing overall training time.

During the practical training phase, the augmented pineapple images were fed into each respective backbone to extract these high-dimensional feature maps. Once generated, these maps were passed directly to the model's shared stages, namely the Region Proposal Network (RPN) and the final detection head. This pipeline allows the network to bypass low-level feature engineering and immediately utilize its computational power to locate targets. Ultimately, both components leverage an anchor-box mechanism within the Faster R-CNN framework to effectively detect pineapple crops with varying sizes and aspect ratios.

### 3.6.2 Deep convolutional models

#### *3.6.2.1 Faster R-CNN*

Faster R-CNN implements the two-stage detection pipeline by focusing on region proposals and classification, as shown in Figure 3.10. The model produces bounding box candidates via a region proposal network (RPN) and refines them with a classifier head. This method refines spatial hypotheses before classification. This characteristic of the model makes it suitable for handling complex agricultural scenes, as it reduces the likelihood of ambiguous detections. With the growing number of

detections to be made, a trade-off with potential extra inference time would be made. Still, it is the case that high-resolution feature interpretation improves detection using pretrained backbones.

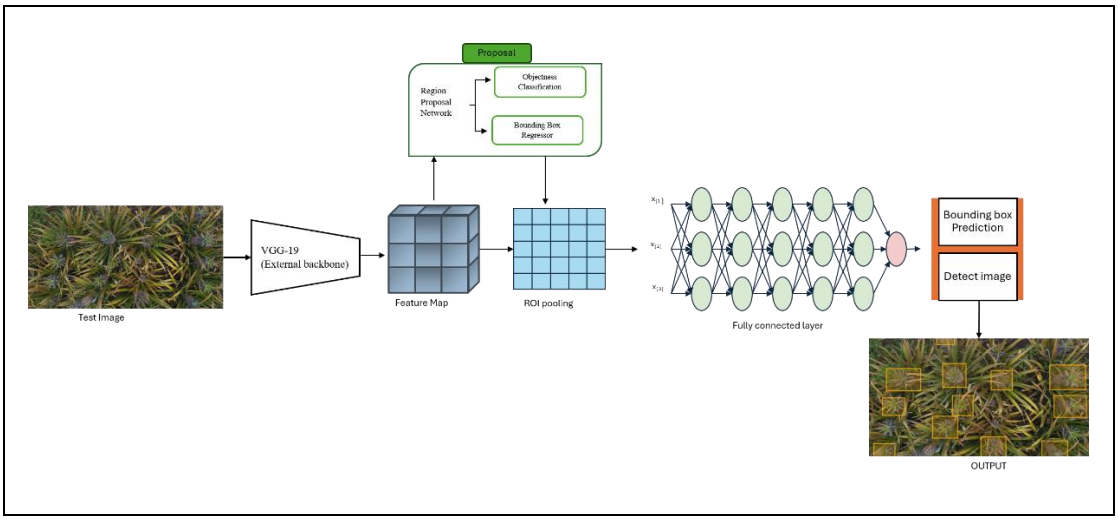



Figure 3.10 Schematic Architecture of the Faster R-CNN Model

### *3.6.2.2 SSD*

SSD is a one-stage detection method that performs object localization and classification in a single feed-forward pass as shown in Figure 3.11. SSD uses a pyramid-structured feature extraction strategy that enables detection of objects across a wide range of sizes. Therefore, it is particularly advantageous for drone-based scenarios where an object is small and/or partially occluded. The architecture strikes a good balance between detection efficiency and reliability. SSD can use several detection feature maps to detect objects across different image sizes and resolutions. However, incorrect detection occurred in some cases, particularly in cluttered scenes where background elements closely resembled the object.

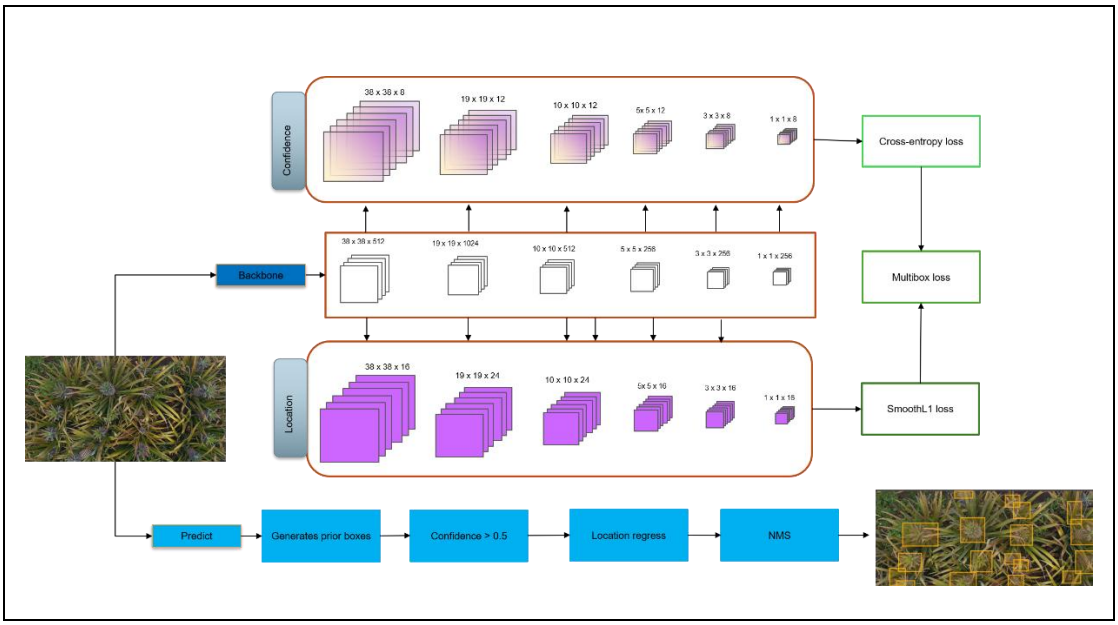



Figure 3.11 Model Structure of SSD

### *3.6.2.3 YOLOv8*

YOLOv8 is an efficient one-stage detector for real-time inference and rapid execution in over-cropped agricultural contexts as shown in Figure 3.12. YOLOv8 can predict bounding boxes and their corresponding class probabilities in a single step, directly from the model's feature maps, thereby significantly reducing delays in the computational pipeline. Its multi-scale feature fusion approach, along with anchor-free prediction, is robust to irregularities from foliage occlusions and background clutter. Its remarkable precision and speed make YOLOv8 a prime candidate for ongoing object detection tasks. YOLOv8's anchor-free architecture and decoupled head were especially pivotal in how this model handled complex environments, particularly in discerning objects from backgrounds with similar visual complexity, varied scales, and lighting conditions.

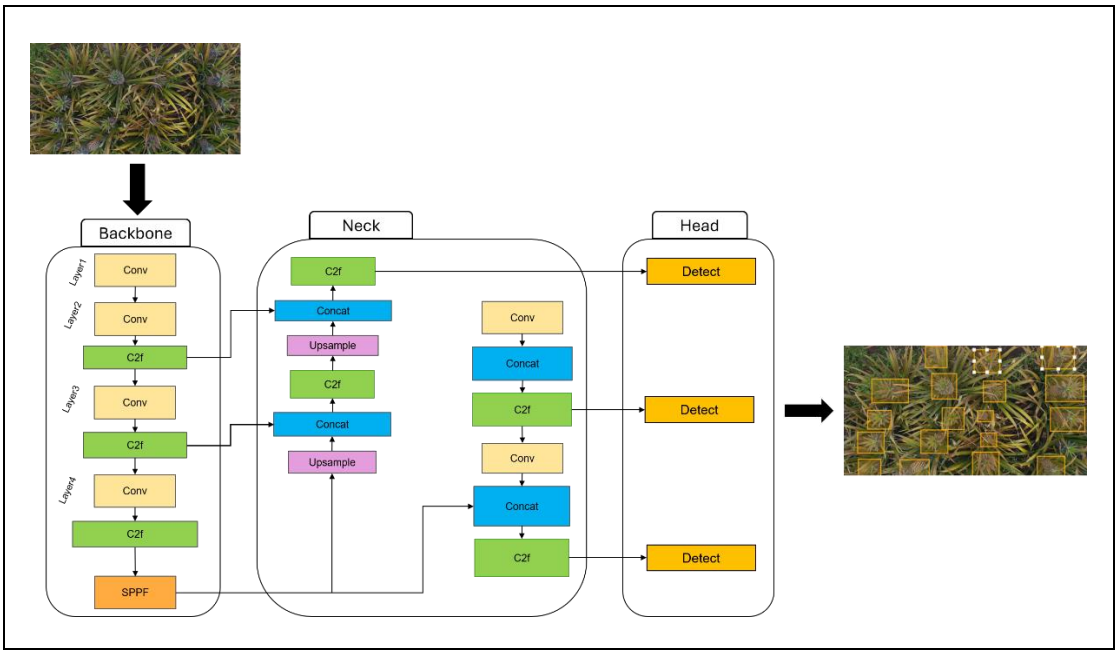



Figure 3.12 The Architectures of YOLOv8

## 3.7 Performance Measures

There are two types of performance evaluation in this research to ensure a comprehensive assessment of the DSSN with DCM for decarbonized efficiency, as shown in Figure 3.13. First, assess the performance of DCM by analyzing classification metrics. Next, the evaluation of the decarbonization effect that occurs during the operational process of the DCM. This research is not only focused on the model's accuracy in detecting pineapple but also on its environmental implications.

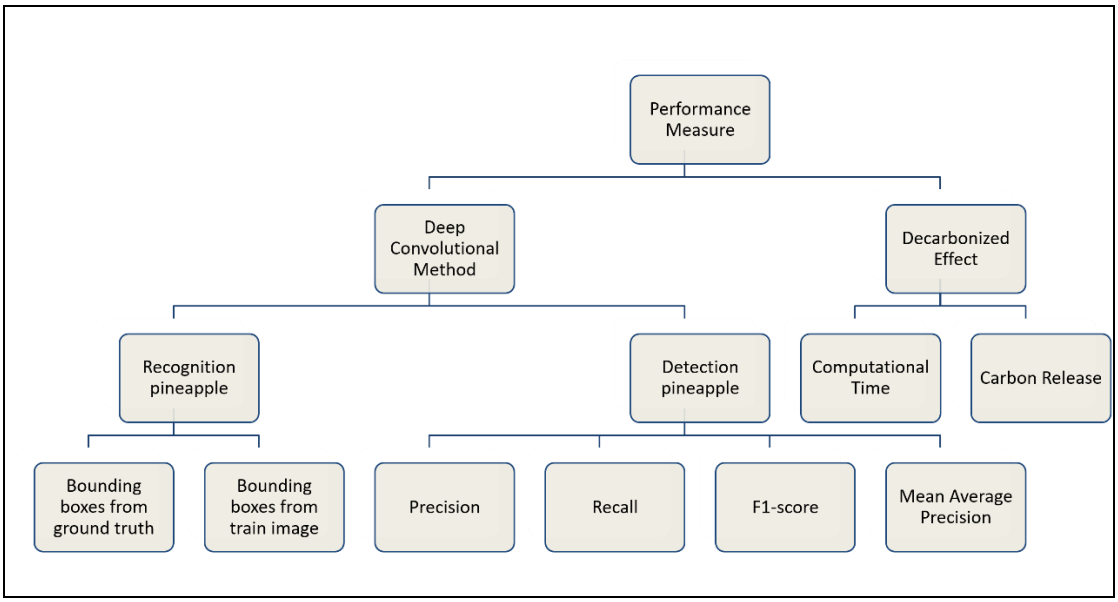



Figure 3.13 Overview of the Performance Measures Used in This Study

### 3.7.1 Classifier Matrix

Several performance indicators were used to assess the effectiveness of the DLM, including the confusion matrix (Figure 3.14). The confusion matrix was obtained from the receiver operating characteristic (ROC) curve. The ROC is defined by the actual positive rate (TPR) and false positive rate (FPR), which are computed from the outcomes confirmed as positive, false positive, true negative, and false negative.

| | | **ACTUAL** | |
|---|---|---|---|
| PREDICTION | | POSITIVE | NEGATIVE |
| | POSITIVE | TRUE POSITIVE | FALSE POSITIVE |
| | NEGATIVE | FALSE NEGATIVE | TRUE NEGATIVE |

Figure 3.14 Confusion Matrix

True Positive (TP) accurately represented the positive cases of actual pineapple detection, True Negative (TN) correctly represented the negative cases of incorrect pineapple detection, False Positive (FP) incorrectly represented the negative cases, and False Negative (FN) incorrectly represented the positive cases of pineapple detection. A higher ROC value indicates a better performance. The confusion matrix also provides

the accuracy (ACC), precision, recall, and F1-score to compare the performance on the test images. The purpose of the confusion matrix is to analyze the model's errors by comparing actual and predicted values. Accuracy and precision describe the number of correctly classified pineapples and the probability of correctly classifying pineapples. The best accuracy was 100%, whereas the best value for the rest of the performance metrics, that is, precision, recall, and F1-score, was 1. Thus, the values of all evaluation performances cannot exceed 1.

The ratio of the total correctly predicted detections of pineapple crops to the total predicted detections of pineapple crops is defined as precision (PRE), as given in (3.13). Recall (REC) shows the ratio of predicted detections that are correctly predicted to the total ground truth of pineapple crops, as shown in (3.14). The F1-score represents the efficiency of the DCN model in detecting and correctly identifying pineapple crops, as given in (3.15). The overall performance of the pineapple detection model across all confidence thresholds, all classes, and all IoU thresholds. Here, mAP determines the overall accuracy of the object detector, where and  respectively denote the number of IoU thresholds (0.50 to 0.95) and the average precision for class k at IoU threshold t, as given by (3.16). The range of IoU thresholds from 0.50 to 0.95 was selected to force a highly stringent assessment of how tightly the predicted bounding boxes fit the actual boundaries of the pineapples, rather than just measuring broad detection. This rigorous evaluation is crucial for drone imagery because pineapple sizes appear significantly smaller at higher altitudes, where even minor bounding box misalignments will drastically drop the IoU score.

$$\mathrm{PRE} = \frac{TP}{TP + FP} \tag{3.13}$$

$$\mathrm{REC} = \frac{TP}{TP + FN} \tag{3.14}$$

$$F1\text{-}Score = 2 \times \frac{\mathrm{PRE} \times \mathrm{REC}}{\mathrm{PRE} + \mathrm{REC}} \tag{3.15}$$

$$\mathrm{mAP} = \frac{1}{K \cdot T} \sum_{k=1}^{K} \sum_{t=1}^{T} \mathrm{AP}_k(\mathrm{IoU}_t) \tag{3.16}$$

### 3.7.2 Carbon Footprint Effect

Decarbonizing processing efficiency is the process of minimizing or eliminating carbon emissions from industrial and computer operations while maintaining or

improving operational efficiency. This concept relates to computing and data processing through energy-efficient technologies, lower reliance on carbon-intensive energy sources, and optimal energy consumption. By using renewable energy and maximizing hardware and software performance, the aim is to lower the environmental impact of operations. Improving processing efficiency while reducing carbon emissions is especially critical in sectors such as deep learning and large-scale data centres, which have high energy demands and rely primarily on computation. Eventually, decarbonization projects will help balance high performance with environmental sustainability, supporting global efforts to lower carbon footprints.

In particular, large-scale deep learning models can be energy-intensive, leading to significant carbon emissions. Energy-efficient hardware, such as GPUs and TPUs, built for deep learning activities, helps to lower energy consumption. Using augmented or synthetic data, data-efficiency strategies reduce model training and testing times. Reducing the carbon footprint and improving processing efficiency through deep learning strike a balance between environmental sustainability and excellent model performance.

The proposed model was configured to process images regardless of pixel size or resolution. It has many features across layers and extensions in the network architecture to achieve the most efficient layer. The DSSN model, which must be improved using deep networks, enables computationally efficient processing with a single unified model that performs segmentation, detection, and classification tasks while incorporating decarbonization functions into the modeling.

Several indicators were used to assess carbon release during the testing phase, based on data generated by the DCM process. The carbon released is influenced by the time taken during the running process. Thus, the time taken was calculated to determine whether the carbon footprint influenced time. The average time per tile is a direct mathematical derivation calculated using only the frame per second (FPS) information, because FPS is the number of tiles processed per second, as shown in Equation (3.17). Utility and efficiency are also calculated to assess the speed efficiency of the tested DLMs, as in Equations (3.18) and (3.19), which are self-formulated to strictly model hardware utilization and processing throughput relative to the tile generation rate. Lastly, the carbon footprint can be calculated by using a formula that is shown in Equation (3.20) (Henderson & Brunskill, 2020; Schwartz et al., 2017). In this formula, energy efficiency, memory energy and carbon intensity are set as constant values.

Hence, the explicit carbon footprint contribution of this research lies in introducing a quantifiable, eco-efficient framework that bridges edge-computing performance and environmental sustainability in precision agriculture. By establishing these self-formulated mathematical indicators, this study provides a novel benchmark for evaluating the deployment of deep learning on UAVs. Specifically, the DSSN approach contributes to carbon reduction by eliminating redundant spatial processing and optimizing hardware utilization (as modeled in Equations 3.18 and 3.19). This guarantees that large-scale crop monitoring does not entail high environmental costs, offering a scalable solution for carbon-neutral agricultural monitoring.

$$Average\ time\ per\ tile = \frac{1}{FPS} \tag{3.17}$$

$$Utility = \frac{FPS \times tile\ generation}{average\ time\ per\ tile} \tag{3.18}$$

$$Efficiency = \frac{FPS}{tile\ generation\ \times (average\ time\ per\ tile^2)} \tag{3.19}$$

$$carbon\ footprint = [\frac{tile\ generation}{FPS} \times (GFLOPs\ \times EFF + E_m)] \times CI \tag{3.20}$$

# CHAPTER 4
# RESULTS AND DISCUSSIONS

## 4.1 Introduction

This chapter discusses all the results obtained and the process of detecting pineapple in complex scenes. The analysis starts by specifying the data collection as train and test data. Before applying DLM which are Faster R-CNN, SSD and YOLOv8, a few hyperparameters for DLM were set, and suitable pretrained deep neural networks were found for this research. Furthermore, a comparison of performance measures was conducted to determine the best DLM to combine with the DSSN model by evaluating several tile sizes for pineapple detection. A few image samples were used as validation from the test data to determine which DLM is most suitable for detecting pineapple in complex backgrounds. The best tile size that was obtained from DLM performance was further evaluated to determine the carbon footprint.

## 4.2 Data Collection

In this study, the dataset was collected, and the training and test data were analyzed to evaluate the performance of the pineapple detection model.

Table 4.1
Dataset for Training and Testing

| | **Train data** | **Test data** |
|---|---|---|
| Type | Crop image of pineapple | Image containing multiple pineapples |
| Size image | 300 (H) x240 (W) | 2160 (H) x 4096 (W) |
| Height from the ground | 10m | 10m, 8m, 6m, and 4m |
| Total | 300 images | 4 images |
| Condition of pineapple images | Pineapples in various positions and appearances | Distribution of annotated pineapples: 25, 92, 234, and 269 instances for 4m, 6m, 8m, and 10m altitudes, respectively |

In this case, the training data consisted of images of pineapple crops extracted from drone footage captured at a height of 10m. These images were subsequently

resized to a resolution of 300 pixels in height (H) and 240 pixels in width (W). In total, 300 training images were captured, providing sufficient variation in pineapple characteristics. This allowed the model to effectively learn and recognize distinct features based on variations in object positioning and background conditions.

In contrast, the actual validation and testing phase utilized higher-resolution test images to simulate real field environments. Each test image, captured at heights of 10m, 8m, 6m, and 4m above the ground, maintained a high resolution of 2160 pixels in H and 4096 pixels in W, allowing for more complex spatial analysis. Four distinct test data images were evaluated at every 2m interval from 10m down to 4m. This setup ensured that the model's detection capability was rigorously assessed using larger, unsegmented images, thereby accurately reflecting real-world situations in agricultural fields.

## 4.3 Hyperparameter Value

Hyperparameter values had to be adjusted to maximise the performance of Faster R-CNN, SSD, and YOLOv8 in detecting pineapples against a complex background. The summary of hyperparameter settings for the DLM is shown in Table 4.2.

Table 4.2
Hyperparameter Values Settings

| | **Faster R-CNN** | **SSD** | **YOLOv8** |
|---|---|---|---|
| Learning Rate (LR) | 0.005 | 0.01 | 0.01 |
| Batch Size | 4 | 4 | 4 |
| Number of epochs | 30 | 50 | 30 |
| Optimizer | SGD | SGD | SGD |
| Weight decay of optimizer | 0.0005 | 0.0005 | 0.0005 |
| Momentum of regularization | 0.9 | 0.9 | Adaptive |
| Metric | Mean IoU | IoU>0.7 | mAP@[0.5:0.95] |
| Image input size | 2160 x 4096 | 2160 x 4096 | 1024 x 1024 |
| Augmentation | None (ToTensor only) | None (ToTensor only) | Strong augmentation (multi-scale, AMP) |

During the training phase, the learning rate (LR) determines how quickly the model learns. To avoid unstable convergence during the two-stage training of the

model, a smaller learning rate of 0.005 was used in Faster R-CNN to prevent duplicate weight overwrites, which could otherwise cause the model to collapse. For SSD and YOLOv8, a learning rate of 0.01 was used to enable faster data iteration, as single-stage detectors require faster convergence. YOLOv8 uses an adaptive learning rate that self-adjusts during training to improve generalization and model stability. All three models were assigned a batch size of four, a moderate size that strikes a balance between efficient computation time and better generalization. A smaller batch size helps reduce memory strain, which is particularly helpful for extremely large images, especially high-resolution images, such as the 2160 × 4096 px used during training.

The number of epochs was adjusted based on each model's performance during training. Faster R-CNN and YOLOv8 were trained for 30 epochs to ensure the models had adequate exposure to the database while minimizing the risk of overfitting. For SSD, 50 epochs were required because the architecture relies on multi-scale feature maps, which typically require more epochs to reach convergence. All models used the Stochastic Gradient Descent (SGD) optimizer, which is a popular choice for object detection tasks owing to its simplicity, efficiency, and robustness. Using SGD, convergence is guaranteed to be steady, and learning rate scheduling, momentum, and weight decay can be implemented with greater control. To combat overfitting, a uniform weight decay of 0.0005 was applied across all models to encourage the learning of generalized features. A momentum value of 0.9 was also used in Faster R-CNN and SSD to smooth the learning process and enable a quicker descent of the gradient in the desired direction. YOLOv8 uses a different momentum strategy that is adaptive during training to achieve faster convergence with stable accuracy.

The metrics used to evaluate the different models also reflect their performance levels. For Faster R-CNN, the Mean Intersection over Union ( IoU) value was used. This metric measures the average overlap between predicted and ground-truth bounding boxes. The SSD used an IoU threshold of > 0.7 to evaluate detection accuracy at a higher level of localization. YOLOv8 is a more sophisticated detection model that operates with an average precision (mAP@[0.5:0.95]). This value measures precision and recall across different IoU thresholds, providing a better grasp of the model's detection quality. For Faster R-CNN and SSD, the original dimensions of the image (2160 × 4096) were used to retain the fine spatial details that were important in the detection of small or occluded pineapples, whereas YOLOv8 had a reduced input dimension of 1024 × 1024, which enabled the model to achieve a proper balance

between detection speed and accuracy owing to its effective backbone and multi-scale learning.

Finally, data augmentation enhances the model's robustness. Faster R-CNN and SSD perform minimal augmentation by simply converting images into tensors with ToTensor, which is essentially image conversion. In contrast, YOLOv8 uses more sophisticated data augmentation methods, such as multi-scale training and Automatic Mixed Precision (AMP), which increase the model's adaptability to changes in illumination, scale, and object orientation. In general, these hyperparameter selections indicate differences in architectural design philosophies. Faster R-CNN is focused on feature extraction to improve accuracy; SSD has a more balanced focus between speed and feature stability and is more training-intensive, while YOLOv8 has a more balanced focus on learning adaptability to the task and on aggressive augmentation to deliver peak results and high generalization.

## 4.4 Performance Evaluation

This study began by evaluating four pretrained deep neural networks, ResNet-101, ResNet-50, Inception-ResNet-v2, and VGG-19, to identify the most appropriate backbone architecture for the model. The models were evaluated using accuracy, precision, recall, and F1-score, and the counts of true positives, true negatives, false positives, and false negatives were calculated. The models were then evaluated, and the most suitable deep neural networks for pineapple detection in complex backgrounds were identified as Faster R-CNN, SSD, and YOLOv8. Furthermore, the ability to sustain continuous detection was investigated by creating prediction curves using multiple interpolation and regression techniques, and each curve was analyzed to determine the best method for predicting the most appropriate tile sizes for detecting pineapples in complex backgrounds at heights other than those in the test images. Finally, the carbon emissions from each train and test model were analyzed to determine the best model in terms of environmental efficiency.

### 4.4.1 Pretrained Deep Neural Network

Comparative evaluations between ResNet-50, ResNet-101, Inception-ResNet-v2, and VGG-19 were conducted using the same set of 10 test images captured by a

UAV at an altitude of 10m above the ground level, as shown in Appendix 1. This maximum altitude of 10m was deliberately selected because it represents the furthest imaging distance in this study. Thus, evaluating the models at this height ensures their capability to achieve effective and reliable pineapple detection even under the most challenging resolution and downscaling conditions. Performance evaluation metrics, including TPR, FPR, precision, recall, and F1-score, are commonly used to assess model performance when analyzing a CNN for image detection. Ultimately, the best-performing pretrained deep neural network will be applied as the backbone for Faster R-CNN only. This external backbone is essential to provide high-quality feature extraction, which directly determines the model's final detection accuracy.

#### *4.4.1.1 ResNet-101*

The performance results obtained with the pretrained ResNet-101 model across all 10 test images showed clear variation in overall detection accuracy and sensitivity with image complexity, as shown in Table 4.3.

Table 4.3
Performance Metrics Using ResNet-101

| Test image | n | TP | TN | FP | FN | ACC | PRE | REC | FI-score |
|---|---|---|---|---|---|---|---|---|---|
| Image 1 | 351 | 335 | 0 | 2 | 14 | 0.9544 | 0.9941 | 0.9599 | 0.9767 |
| Image 2 | 364 | 229 | 0 | 0 | 135 | 0.6291 | 1.0000 | 0.6291 | 0.7723 |
| Image 3 | 370 | 225 | 0 | 1 | 144 | 0.6081 | 0.9956 | 0.6098 | 0.7563 |
| Image 4 | 355 | 219 | 0 | 0 | 136 | 0.6169 | 1.0000 | 0.6169 | 0.7631 |
| Image 5 | 362 | 240 | 0 | 1 | 121 | 0.6630 | 0.9959 | 0.6648 | 0.7973 |
| Image 6 | 355 | 225 | 0 | 0 | 130 | 0.6338 | 1.0000 | 0.6338 | 0.7759 |
| Image 7 | 363 | 223 | 0 | 2 | 138 | 0.6143 | 0.9911 | 0.6177 | 0.7611 |
| Image 8 | 367 | 224 | 0 | 1 | 142 | 0.6104 | 0.9956 | 0.6120 | 0.7580 |
| Image 9 | 363 | 216 | 0 | 1 | 146 | 0.5950 | 0.9954 | 0.5967 | 0.7461 |
| Image 10 | 334 | 208 | 0 | 0 | 146 | 0.5876 | 1.0000 | 0.5876 | 0.7402 |

The accuracy of pineapple detection ranged from 0.5876 to 0.9544. Although the accuracy of some images was exceptional, the poor accuracy of other images was attributed to the complexity of the visual conditions. The precision of the model ranged from 0.9911 to 1.0000, indicating that it was reliable for positive detections and not a major contributor to false positives. The recall values produced by ResNet-101 varied

the most, ranging from 0.5876 to 0.9599, indicating the model's ability to detect positives. However, in some images, the model did not achieve sufficient positive detections to be considered successful. The next consideration, the F1 score, which considers both precision and recall, ranged from 0.7402 to 0.9767. Thus, the score exhibited good stability in detection, but the floating F1 score indicated the presence of complex visual conditions. Overall, the ResNet-101 model maintained strong consistency in positive detections across all images, but accuracy and recall were slightly affected by varying test image conditions.

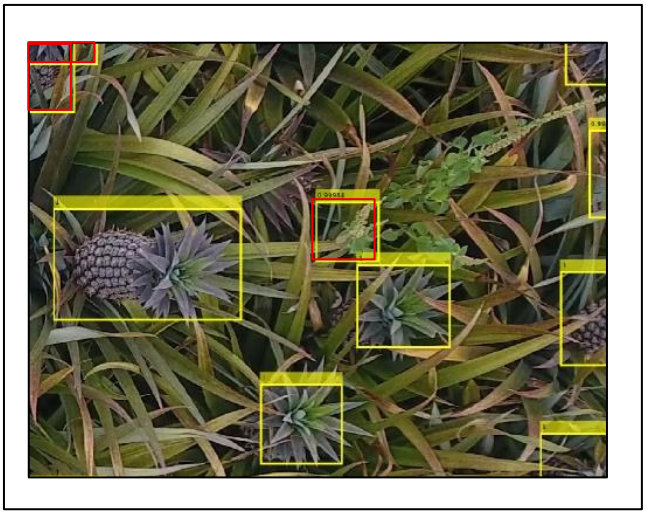


Figure 4.1 Demonstration of Pineapple Detection from the Image by Using ResNet-101

As shown in Figure 4.1, most bounding boxes successfully identified pineapples within the dense, complex background of leaves using the pretrained ResNet-101. This demonstrates that the model can accurately distinguish pineapple features, even under challenging visual conditions, such as overlapping leaves and variations in lighting. However, it can also be observed that in some instances, multiple bounding boxes are generated around a single pineapple. This overlap in detections indicates that the model occasionally interprets a single object as multiple instances, resulting in a slightly inflated count of detected pineapples.

#### *4.4.1.2 ResNet-50*

The results indicate that the ResNet-50 model performs consistently well across all images, maintaining high precision and balanced recall values. However, minor

fluctuations are observed depending on the complexity and clarity of each test image, as shown in Table 4.4.

Table 4.4
Performance Metrics Using ResNet-50

| Test image | n | TP | TN | FP | FN | ACC | PRE | REC | FI-score |
|---|---|---|---|---|---|---|---|---|---|
| Image 1 | 351 | 271 | 0 | 0 | 80 | 0.7721 | 1.0000 | 0.7721 | 0.8714 |
| Image 2 | 364 | 258 | 0 | 1 | 105 | 0.7088 | 0.9961 | 0.7107 | 0.8296 |
| Image 3 | 370 | 264 | 0 | 0 | 106 | 0.7135 | 1.0000 | 0.7135 | 0.8328 |
| Image 4 | 355 | 261 | 0 | 1 | 93 | 0.7352 | 0.9962 | 0.7373 | 0.8474 |
| Image 5 | 362 | 272 | 0 | 0 | 90 | 0.7514 | 1.0000 | 0.7514 | 0.8580 |
| Image 6 | 355 | 258 | 0 | 0 | 97 | 0.7268 | 1.0000 | 0.7268 | 0.8418 |
| Image 7 | 363 | 260 | 0 | 1 | 102 | 0.7163 | 0.9962 | 0.7182 | 0.8347 |
| Image 8 | 367 | 265 | 0 | 0 | 102 | 0.7221 | 1.0000 | 0.7221 | 0.8386 |
| Image 9 | 363 | 262 | 0 | 0 | 101 | 0.7218 | 1.0000 | 0.7218 | 0.8384 |
| Image 10 | 354 | 254 | 0 | 3 | 97 | 0.7175 | 0.9883 | 0.7236 | 0.8355 |

Accuracy values ranging from 0.7088 to 0.7721 testified to the model's remarkable detection efficiency, even as the slight changes were most likely due to variant background elements and the positioning of the pineapples. The precision metrics were very high, ranging from 0.9883 to 1.0000, indicating a high proportion of true positives among pineapple detections and few false positives. The model demonstrated highly active performance on true pineapples, achieving recalls of 0.7107 and 0.7721, although a few true pineapples may have gone undetected in some images. The F1 scores from the test images were the same as the precision and recall, at 0.8296. The systems' precision and recall were balanced across the images at their extremes and reflected harmonious performance. With clear confidence in accuracy and performance, the ResNet-50 model demonstrated high precision and effective classification of individual pineapples, even in complex surroundings.

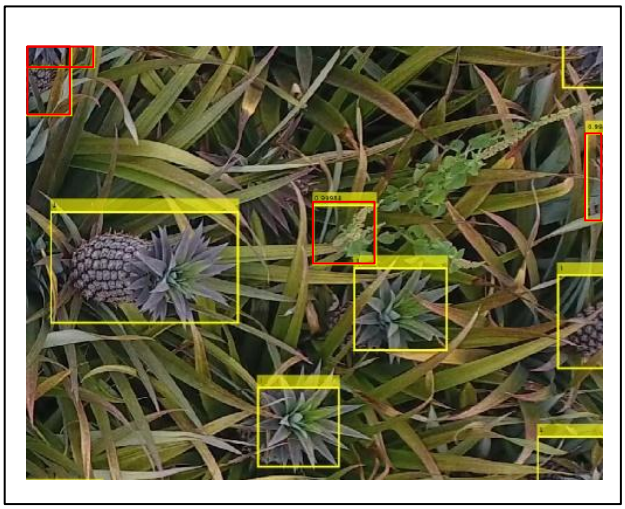


Figure 4.2 Demonstration of Pineapple Detection from the Image by Using ResNet-50

The pretrained ResNet-50 model demonstrated strong and consistent detection capabilities, as shown in Figure 4.2, with particularly high precision, confirming its effectiveness in accurately classifying pineapples in complex environments. However, a small number of marginal misdetections were observed, primarily in areas where elongated pineapple leaves mimicked the geometric structure and tonal patterns of developing fruit, resulting in occasional false positives. Nevertheless, these errors are infrequent and do not significantly affect the overall detection reliability. Hence, minor improvements in recall and overall accuracy could further enhance detection robustness, particularly in cases involving overlapping leaves or partially occluded fruit.

#### *4.4.1.3 InceptionResNet-v2*

InceptionResNet-v2 showed the worst performance among the neural networks due to environmental issues in the densely planted pineapple field. Table 4.5 shows the performance metrics of InceptionResNet-v2 in detecting pineapples.

Table 4.5
Performance Metrics Using ResNet-50

| Test image | n | TP | TN | FP | FN | ACC | PRE | REC | FI-score |
|---|---|---|---|---|---|---|---|---|---|
| Image 1 | 351 | 149 | 0 | 25 | 177 | 0.4245 | 0.8563 | 0.4571 | 0.5960 |
| Image 2 | 364 | 157 | 0 | 20 | 187 | 0.4313 | 0.8870 | 0.4564 | 0.6027 |
| Image 3 | 370 | 152 | 0 | 26 | 192 | 0.4108 | 0.5390 | 0.4419 | 0.5824 |
| Image 4 | 355 | 163 | 0 | 16 | 176 | 0.4592 | 0.9106 | 0.4808 | 0.6293 |
| Image 5 | 362 | 154 | 0 | 16 | 192 | 0.4254 | 0.9059 | 0.4451 | 0.5969 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Image 6 | 355 | 158 | 0 | 21 | 176 | 0.4451 | 0.8827 | 0.4731 | 0.6160 |
| Image 7 | 363 | 167 | 0 | 10 | 186 | 0.4601 | 0.9435 | 0.4731 | 0.6302 |
| Image 8 | 367 | 158 | 0 | 13 | 196 | 0.4305 | 0.9240 | 0.4463 | 0.6019 |
| Image 9 | 363 | 157 | 0 | 14 | 192 | 0.4325 | 0.9181 | 0.4499 | 0.6038 |
| Image 10 | 354 | 171 | 0 | 10 | 173 | 0.4831 | 0.9448 | 0.4971 | 0.6514 |

The majority of predictions across all images are incorrect, as shown in the Table. This will be demonstrated by the accuracies of 0.4108 and 0.4831, which indicate that fewer than 50% of detections are correct. There were sufficient correct predictions within and among the images that some of the pineapples present in the field of view and captured against complex backgrounds were identified, yielding precision levels ranging from 0.5390 to 0.9448. The performance and consistency across the other images resulted in recall values ranging from 0.4419 to 0.4971. Consequently, the F1-score values, which ranged from 0.5824 to 0.6514, indicated a weak overall balance between precision and recall, confirming that the model's detections were neither consistent nor comprehensive. Thus, the InceptionResNet-v2 model had such poor detection performance, as indicated by the accuracy, recall, and F1 scores.

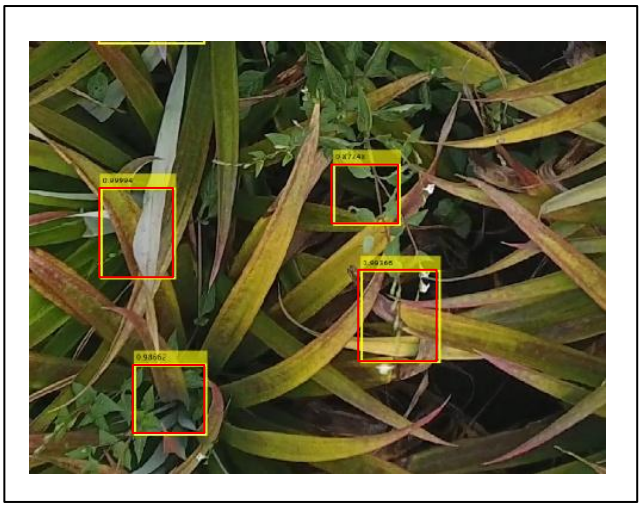


Figure 4.3 Demonstration of Pineapple Detection from the Image by Using InceptionResNet-v2

InceptionResNet-v2 is a deep architecture that excels in image classification. However, it likely struggles with small, partially occluded objects, such as pineapples buried among dense foliage. Additionally, the multiscale feature extraction mechanism is likely not precise enough to capture the fine details required for accurate object localization in natural settings. Furthermore, the cluttered, chaotic background of the

pineapple field likely adds significant visual noise, leading to confusion between the pineapple and the surrounding leaves, as shown in Figure 4.3.

#### *4.4.1.4 VGG-19*

VGG-19 demonstrated the most consistent and best performance in detection and strong feature extraction on image test cases, as shown in Table 4.6. The results show that VGG-19 provides consistent performance in all test cases and strong detection performance.

Table 4.6
Performance Metrics Using VGG-19

| Test image | n | TP | TN | FP | FN | ACC | PRE | REC | FI-score |
|---|---|---|---|---|---|---|---|---|---|
| Image 1 | 351 | 266 | 0 | 0 | 85 | 0.7578 | 1.0000 | 0.7578 | 0.8622 |
| Image 2 | 364 | 289 | 0 | 0 | 75 | 0.7940 | 1.0000 | 0.7940 | 0.8851 |
| Image 3 | 370 | 287 | 0 | 1 | 82 | 0.7757 | 0.9965 | 0.7778 | 0.8737 |
| Image 4 | 355 | 284 | 0 | 0 | 71 | 0.8000 | 1.0000 | 0.8000 | 0.8889 |
| Image 5 | 362 | 282 | 0 | 0 | 80 | 0.7790 | 1.0000 | 0.7790 | 0.8758 |
| Image 6 | 355 | 282 | 0 | 0 | 73 | 0.7944 | 1.0000 | 0.7944 | 0.8854 |
| Image 7 | 363 | 305 | 0 | 1 | 57 | 0.8402 | 0.9967 | 0.8425 | 0.9132 |
| Image 8 | 367 | 302 | 0 | 0 | 65 | 0.8229 | 1.0000 | 0.8229 | 0.9028 |
| Image 9 | 363 | 272 | 0 | 1 | 90 | 0.7493 | 0.9963 | 0.7514 | 0.8567 |
| Image 10 | 354 | 287 | 0 | 0 | 67 | 0.8107 | 1.0000 | 0.8107 | 0.8955 |

VGG-19 had a strong overall performance, with consistent feature extraction across most test cases. This led to consistent performance across all test cases, with detection accuracy above 75%. It showed detection accuracy ranging from 75 to 84 or higher, demonstrating clear, overall, and consistent detection despite the complex background environment. Furthermore, VGG-19 demonstrated an extremely high detection performance for individual objects. Specifically, the results ranged from 0.9965 to 1.0. The recall values, which indicate the model's ability to detect all instances of pineapple objects across all images, ranged from 0.7578 to 0.8425. Therefore, the model was found to detect most of the true positives in each image. This resulted in an F1 score for the model ranging from 0.8622 to 0.9132. This indicates that VGG-19 achieves optimal precision and recall simultaneously across all images.



Figure 4.4 Demonstration of Pineapple Detection from the Image by Using VGG-19

VGG-19 can still be attributed to the profoundness of the model's deep feature representations, which capture a wide range of detailed textures and shapes. This makes the model one of the best at distinguishing leaves from other environmental noise, as shown in Figure 4.4. The VGG-19's uniform architecture provides stable, generalized features that are crucial for detecting pineapple in complex scenes.

#### *4.4.1.5 Average for Pretrained Deep Neural Network*

The baseline performance of different pretrained deep neural networks, namely ResNet-101, ResNet-50, VGG-19, and Inception-ResNet-v2, was evaluated on ten representative images captured at an altitude of 10 m above ground level. All models were tested under identical conditions to identify the most suitable backbone for long-distance pineapple detection with Faster R-CNN. The results show a clear difference between the networks: VGG-19 produced the highest and most consistent detection accuracy, whereas Inception-ResNet-v2 recorded the lowest performance. More details of this comparison are presented in Figure 4.5 and Table 4.7.

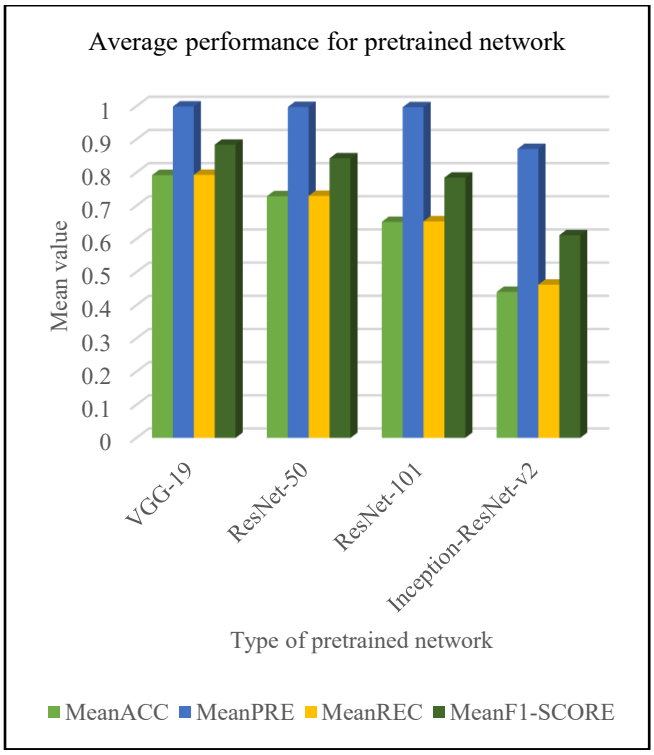



Figure 4.5 Average Performance for the Pretrained Network

Table 4.7
Average Performance Comparison Across Multiple Metrics

| | **ResNet-101** | **ResNet-50** | **Inception-ResNet-v2** | **VGG-19** |
|---|---|---|---|---|
| ACC | 0.6513 | 0.7286 | 0.4403 | 0.7955 |
| PRE | 0.9968 | 0.9977 | 0.8712 | 0.9992 |
| REC | 0.6528 | 0.7298 | 0.4621 | 0.7961 |
| F1-score | 0.7847 | 0.8428 | 0.6111 | 0.8859 |

The experimental outcome demonstrated that the VGG-19 model was the most precise for 10 images taken from a distance of 10m from the ground. It provided the highest values for all performance parameters, with an accuracy of 0.7924, precision of 0.9990, recall of 0.7930, and F1-score of 0.8839. ResNet-50 was second to VGG-19, followed by ResNet-101. The performance evaluation for ResNet-50 resulted in an accuracy of 0.7285, precision of 0.9977, recall of 0.7297, and F1-score of 0.8428. For ResNet-101, the accuracy was 0.6513, with a precision of 0.9968, a recall of 0.6528, and an F1-score of 0.7847. Finally, the performance values indicate that Inception-ResNet-v2 has the worst performance among the models, with a very low ACC of 0.4402 due to high false-negative and false-positive rates, which means the model makes many incorrect predictions.

### 4.4.2 Deep Convolutional Method

Three deep convolutional methods, namely Faster R-CNN, SSD, and YOLOv8, were used to detect pineapples in images containing multiple pineapples, captured at distances ranging from 10 meters to 4 meters in 2-meter increments. Several performance indicators are used to examine and assess the best detection performance of DCM. This DCM was observed with a combination of the DSSN model.

#### *4.4.2.1 Faster R-CNN*

The assessment of Faster R-CNN as a deep convolutional detection representation in Table 4.8 shows how model performance changed with different tile heights and slice sizes for the images.

Table 4.8
Performance Metrics Using Faster R-CNN

| Height | Tile size | PRE | REC | F1-score | mAP |
|---|---|---|---|---|---|
| 10m | 90 | 0.51 | 0.93 | 0.66 | 0.445 |
| | 73 | 0.54 | 0.68 | 0.60 | 0.423 |
| | 270 | 1 | 0.08 | 0.15 | 0.673 |
| | 303 | 1 | 0.02 | 0.04 | 0.533 |
| 8m | 99 | 0.67 | 0.84 | 0.75 | 0.561 |
| | 73 | 0.59 | 0.54 | 0.56 | 0.391 |
| | 270 | 0.93 | 0.33 | 0.48 | 0.788 |
| | 303 | 1 | 0.43 | 0.61 | 0.931 |
| 6m | 170 | 0.7 | 0.86 | 0.77 | 0.506 |
| | 73 | 0.36 | 0.58 | 0.44 | 0.067 |
| | 270 | 0.91 | 0.57 | 0.70 | 0.868 |
| | 303 | 1 | 0.43 | 0.61 | 0.931 |
| 4m | 338 | 0.69 | 0.53 | 0.60 | 0.491 |
| | 73 | 0.21 | 0.35 | 0.26 | 0.035 |
| | 270 | 0.57 | 0.76 | 0.65 | 0.422 |
| | 303 | 0.55 | 0.65 | 0.59 | 0.485 |

At 10 m, although the model could still detect pineapples, the precision ranged from 0.51 to 1.00, recall from 0.02 to 0.93, and F1 scores from 0.04 to 0.66, with mAP

values of 0.423 and 0.673. At this height, a tile size of 90 pixels achieved the best performance among the tile sizes, with an F1 score of 0.66 and an mAP of 0.445.

Faster R-CNN showcased better performance in detecting pineapples at 8 m, achieving precision scores of 0.59 to 1.00, recall of 0.43 to 0.84, F1 scores of 0.48 to 0.75, and mAP scores ranged from 0.533 to 0.931. At a tile size of 99 pixels, the results seemed optimal, achieving a balanced performance with an F1 score of 0.75 and an mAP of 0.561 for recognizing the pineapple.

Moreover, the 6 m height yielded the best detection results: precision scores of 0.36 to 1.00, recall of 0.43 to 0.86, F1 of 0.61 to 0.77, and mAP of 0.578 to 0.868. The strongest results were achieved for a tile size of 170 pixels, with F1 and mAP values of 0.77 and 0.806, respectively. Hence, a good arrangement between the field coverage of tile size and pineapple size puts the Faster R-CNN in a better position to recognize and detect pineapples with complex backgrounds.

At 4 m, Faster R-CNN identified features and instances with a slight decrease in detection performance. The precision values were in the range of 0.21 to 0.69, recall was within the range of 0.35 to 0.76, F1 score values ranged from 0.26 to 0.65, and mAP was in the range of 0.035 to 0.491. Then, a tile size of 338 pixels achieved the best F1-score of 0.60 and mAP of 0.491, indicating that the largest tile size at lower heights improves performance in perfectly recognizing the pineapple.

#### *4.4.2.2 SSD*

The assessment of SSD detection, as a deep convolutional method, revealed disparities in precision and consistency in detecting pineapple across different UAV flight altitudes and tile sizes, as shown in Table 4.9.

Table 4.9
Performance Metrics Using SSD

| Height (m) | Tile size (px) | PRE | REC | F1-score | mAP |
|---|---|---|---|---|---|
| 10 | 90 | 0.68 | 0.37 | 0.48 | 0.43 |
| | 73 | 0.37 | 0.41 | 0.39 | 0.234 |
| | 270 | 0.58 | 0.05 | 0.08 | 0.165 |
| | 303 | 0.74 | 0.05 | 0.1 | 0.13 |
| 8 | 99 | 0.48 | 0.34 | 0.4 | 0.346 |
| | 73 | 0.31 | 0.44 | 0.36 | 0.124 |

| | | | | | |
|---|---|---|---|---|---|
| | 270 | 0.91 | 0.1 | 0.19 | 0.546 |
| | 303 | 0.96 | 0.13 | 0.23 | 0.346 |
| 6 | 170 | 0.42 | 0.3 | 0.35 | 0.26 |
| | 73 | 0.12 | 0.7 | 0.21 | 0.016 |
| | 270 | 0.59 | 0.25 | 0.35 | 0.479 |
| | 303 | 0.59 | 0.25 | 0.35 | 0.428 |
| 4 | 338 | 0.23 | 0.29 | 0.26 | 0.207 |
| | 73 | 0.01 | 0.65 | 0.03 | 0.001 |
| | 270 | 0.19 | 0.41 | 0.26 | 0.135 |
| | 303 | 0.22 | 0.35 | 0.27 | 0.127 |

In the case at 10 m, the SSD obtained a precision from 0.37 to 0.74, recall from 0.05 to 0.41, F1 scores from 0.08 to 0.48, and mAP from 0.13 to 0.43. The best overall result, F1 is 0.48 and mAP is 0.4 when the tile size was sliced at 90 pixels compared to fixed tile sizes of 73, 270, and 303. Hence, it shows that higher altitudes and smaller tile sizes are better for detecting the pineapple in complex backgrounds because they better retain object sizes within the visual detail.

At 8 m height, the SSD pineapple detection ability improved with precision scores from 0.31 to 0.96, recall scores from 0.10 to 0.44, F1 scores from 0.19 to 0.40, and the mAP values ranged from 0.124 to 0.546, with a tile size of 99 pixels yielding the best performance balance, with F1 of 0.40 and mAP of 0.346. The SSD improved detection performance at moderate heights, with a finer contextual balance around the pineapple achieved by using larger tiles, thereby removing the need for spatial detail. These results also indicate that improving the representation of features within an image requires an appropriate tile size for the image scale to recognize the pineapple.

The SSD model's performance detection is more stable at 6 m, with precision ranging from 0.12 to 0.59, recall from 0.25 to 0.70, F1-score from 0.21 to 0.35, and mAP from 0.016 to 0.479. A tile size of 170 pixels achieves the best overall results, with an F1 Score of 0.35 and an mAP of 0.26. There was a good balance between ground coverage and image resolution, thoroughly capturing the required features of a pineapple, while the bounding box prediction was stable, improving recognition performance.

Sensitivity to image context and a slight decline in SSD performance at 4 m showed that precision ranged from 0.01 to 0.23, recall from 0.26 to 0.65, F1-score from 0.20 to 0.27, and mAP from 0.007 to 0.207. Simultaneously, a tile size of 338 pixels

had higher performance, with an F1-score of 0.26 and an mAP of 0.207, than other tile sizes, which were more uniformly lower in detecting pineapple in complex scenes. Larger tile sizes were less constrained at lower altitudes and yielded better detection performance due to reduced object fragmentation.

#### *4.4.2.3 YOLOv8*

Evaluating YOLOv8 as a deep convolutional detection technique revealed that it achieved the best detection performance and was more stable than Faster R-CNN and SSD. Table 4.10 shows the performance of YOLOv8 in detecting pineapple using the DSSN method at the established tile size.

Table 4.10
Performance Metrics Using YOLOv8

| Height (m) | Tile size (px) | PRE | REC | F1-score | mAP |
|---|---|---|---|---|---|
| 10 | 90 | 1 | 1 | 1 | 0.994 |
| | 73 | 0.71 | 0.58 | 0.64 | 0.512 |
| | 270 | 0.62 | 0.76 | 0.68 | 0.486 |
| | 303 | 0.73 | 0.86 | 0.79 | 0.556 |
| 8 | 99 | 1 | 1 | 1 | 1 |
| | 73 | 0.71 | 0.58 | 0.64 | 0.512 |
| | 270 | 0.73 | 0.86 | 0.79 | 0.572 |
| | 303 | 0.73 | 0.86 | 0.79 | 0.556 |
| 6 | 170 | 0.95 | 1 | 0.97 | 0.95 |
| | 73 | 0.34 | 0.76 | 0.47 | 0.148 |
| | 270 | 0.75 | 0.82 | 0.78 | 0.568 |
| | 303 | 0.74 | 0.75 | 0.75 | 0.53 |
| 4 | 338 | 1 | 0.82 | 0.90 | 1 |
| | 73 | 0.12 | 0.59 | 0.21 | 0.023 |
| | 270 | 0.68 | 0.88 | 0.77 | 0.49 |
| | 303 | 0.65 | 0.76 | 0.7 | 0.533 |

YOLOv8 achieved the highest overall accuracy in detecting pineapples at a height of 10 m. It achieved perfect metrics, the best recognition for pineapple at a tile size of 90 pixels, with a precision of 1, a recall of 1, an F1-score of 1, and an mAP of 0.994. That means the model accurately located pineapple objects without false

detection. Other tile sizes had slightly lower F1 scores, ranging from 0.64 to 0.79, and mAP values of 0.486 to 0.556, indicating that YOLOv8 still performed well.

At 8 m, the model also achieves positive detection results with a tile size of 99 pixels, achieving perfect performance across all metrics: precision, recall, F1-Score, and mAP were all 1.0, demonstrating unerring detection and localization accuracy. Other tile sizes yielded F1-scores of 0.64 to 0.79 and mAP values of 0.512 to 0.572. At this height, YOLOv8 shows good detection performance because the image quality is balanced. Also, YOLOv8 possesses advanced intelligent multiscale feature extraction and adaptive learning capabilities that allow it to mitigate scale and complex background.

At 6 m height, YOLOv8 continued to perform well in recognition and detection of pineapples with complex background, scoring between 0.34 and 0.95 on precision, 0.47 and 0.86 on recall, and 0.47 to 0.78 on the F1-score. The tile size of 170 achieved an F1-score of 0.78 and an mAP of 0.593, indicating the best detection performance at this height. Thus, the results indicate that YOLOv8 benefits from higher image resolution, leading to improved bounding box precision and more effective feature learning.

With a height of 4 m, YOLOv8 recorded a precision of 0.12 to 0.68, a recall of 0.59 to 0.88, an F1-score of 0.21 to 0.77, and mAP values within the range of 0.023 to 1.00. The highest mAP is 1.00, and the F1-score is 0.9, attributed to a tile size of 338 pixels, confirming that larger tiles at lower altitudes can cover a wider contextual area and exhibit less fragmentation of pineapple features.

#### *4.4.2.4 Identify The Best Tile Size*

An important distinction in this study is the origin of the tile sizes used for the pineapple detection. Variable tile sizes of 90, 99, 170, and 336 pixels were derived from the average dimensions of the ground-truth bounding boxes for images captured at different ground heights, corresponding to flight altitudes of 10 m, 8 m, 6 m, and 4 m, respectively. This demonstrates that adaptive spatial partitioning leads to significantly improved pineapple detection performance compared with fixed tile sizes of 73, 270, and 303 pixels, which were derived from bounding box dimensions in the training images. The results indicate that tile sizes informed by ground-truth annotations yielded superior performance across all evaluated metrics, including mAP, precision, recall, and

F1-score, compared with those derived from generic training images. For example, when using the fixed tile sizes, models like SSD suffered severe performance drops, plummeting to an F1-score of 0.03 at a 4 m altitude, whereas the ground-truth guided tile sizes consistently stabilized and recovered the tracking metrics across all architectures. This spatial correlation provides meaningful insights into how image resolution and object scale influence the effectiveness of deep-learning models in agricultural scenarios. Using tile sizes based on the ground truth better aligned with the actual pineapple scale, leading to more reliable model detection. These findings collectively affirm that tile size plays a critical role in influencing the detection efficacy of deep learning models.

#### *4.4.2.5 The Best Deep Convolutional Method*

This observation was consistent across the three deep learning architectures considered, Faster R-CNN, SSD, and YOLOv8, as shown in Figure 4.6 based on Table 4.11. Among the three deep learning models evaluated, the YOLOv8 model achieved the best performance, consistently demonstrating the highest detection performance across all variable tile configurations, outperforming Faster R-CNN's peak F1-score of 0.77 and SSD's highly constrained mAP scores. For instance, YOLOv8 achieved values of 0.8 or higher across key performance metrics, with some results approaching perfect detection. For all evaluation metrics, performance was perfect, with mAP, precision, recall, and F1-score each reaching 1.00 across 99 pixels at an 8 m flight altitude. Hence, YOLOv8 is not only computationally efficient but also highly robust with respect to spatial resolution variations. These findings emphasize that when tile sizes are appropriately aligned with the object dimensions in the test images, YOLOv8 can outperform other state-of-the-art models in both accuracy and generalizability.

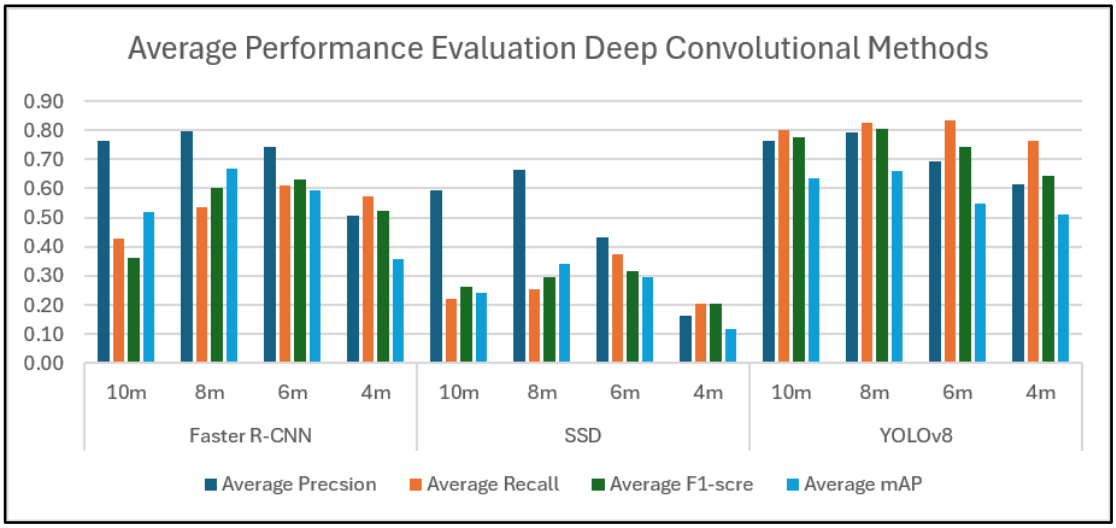



Figure 4.6 Average Performance Evaluation Deep Convolutional Methods

Table 4.11
Tabulate Data of Average Performance Evaluation Deep Convolutional Methods

| DCM | Height (m) | Average Precision | Average Recall | Average F1-score | Average mAP |
|---|---|---|---|---|---|
| Faster R-CNN | 10 | 0.76 | 0.43 | 0.36 | 0.52 |
| | 8 | 0.80 | 0.54 | 0.60 | 0.67 |
| | 6 | 0.74 | 0.61 | 0.63 | 0.59 |
| | 4 | 0.51 | 0.57 | 0.53 | 0.36 |
| SSD | 10 | 0.59 | 0.22 | 0.26 | 0.24 |
| | 8 | 0.67 | 0.25 | 0.30 | 0.34 |
| | 6 | 0.43 | 0.38 | 0.32 | 0.30 |
| | 4 | 0.16 | 0.21 | 0.21 | 0.12 |
| YOLOv8 | 10 | 0.77 | 0.80 | 0.78 | 0.64 |
| | 8 | 0.79 | 0.83 | 0.81 | 0.66 |
| | 6 | 0.70 | 0.83 | 0.74 | 0.55 |
| | 4 | 0.61 | 0.76 | 0.65 | 0.51 |

YOLOv8 capitalizes exceptionally well on the localized features generated by DSSN's spatial slicing. By using an anchor-free detection mechanism and an advanced Path Aggregation Network (PANet), YOLOv8 efficiently integrates micro-level crop details from sliced tiles into the global semantic context. This synergy eliminates anchor-box mismatch issues, allowing the loss function to converge rapidly and pushing almost all performance metrics, including precision, recall, and F1-score, nearly to 1.0. Conversely, the failure of SSD is not a limitation of the DSSN framework but rather a structural flaw in SSD's design. SSD relies on rigid, independent multi-scale feature maps to detect objects directly without a centralized feature aggregation layer. When

presented with the highly detailed, sliced agricultural tiles from DSSN, SSD's default anchor boxes failed to adapt or stabilize. This caused extreme gradient instability and localized spatial disorientation, demonstrating that SSD lacks the structural capacity to process high-resolution sliced inputs effectively.

The perfect metric score (1.00) obtained by the optimized YOLOv8 framework under the 99-pixel configuration notably surpasses the findings of authors from (Shiu et al., 2023), who utilized Faster R-CNN and Mask R-CNN for ripe pineapple yield estimation and reported a lower F1-score of 0.849 due to false positives caused by specular sun glints and midday reflections on leaves. This confirms that our adaptive spatial slicing approach successfully minimizes such background environmental noise. Furthermore, the framework outclasses state-of-the-art agricultural models in other domains, such as the ground-based apple detection model by (An et al., 2022) which yielded an F1 Score of 0.891 with Faster R-CNN. Even when evaluated against recent YOLO-based variations, our system demonstrates higher precision alignment. For instance, (Terven et al., 2023) achieved an mAP50 of 92.3% but struggled with extreme object aspect ratios, while the canopy segmentation model by (Jia et al., 2024) capped at an mAP of 93.1% due to occlusion challenges. Consequently, these comparisons prove that configuring tile sizes based on localized ground-truth boundaries allows YOLOv8 to overcome the spatial and environmental constraints identified in previous studies.

### 4.4.3 Results of Interpolation and Regression Lines

The appropriate tile heights captured from images taken at 4m, 6m, 8m, and 10m above the ground were used to estimate tile heights for validation images captured at 3m, 5m, 7m, and 9m. This approach ensures a structured, systematic method for adjusting tile sizes as the image scale varies with height above the ground. The most efficient tile sizes from each height that achieve the best performance evaluation measurement, which is the average bounding box size from the ground truth images, ensured that the tile size was in proportion to the captured pineapple in the image. Thus, the tile sizes were used on validation images to identify unseen data that can be used for other heights in real scenarios.

The prediction lines from both interpolation and regression were analyzed to delineate the relationship between a UAV's altitude and the corresponding tile size.

However, this research was tested only on four validation images from the test dataset, representing different ground heights. Therefore, this is sufficient to capture the graph's shape representing tile size in a nonlinear manner and to test the prediction methods. Five methods of prediction line for unseen data are shown in Figure 4.7, which represent piecewise linear interpolation in green line, polynomial interpolation in yellow line, cubic spline interpolation in red line, linear regression in blue line, and exponential regression in orange line, with the reference heights from the ground truth bounding boxes represented with distinct dotted markers.

To provide a quantitative comparison, all numerical values corresponding to the prediction lines are shown in Table 4.12. The table shows the predicted tile sizes for the various techniques at the sampling heights from validation images used for pineapple detection. The interpolation lines method showed greater proximity to the reference values, confirming that interpolation techniques are the most suitable for representing non-linear scale variance in drone images and corresponding to the visual pattern.

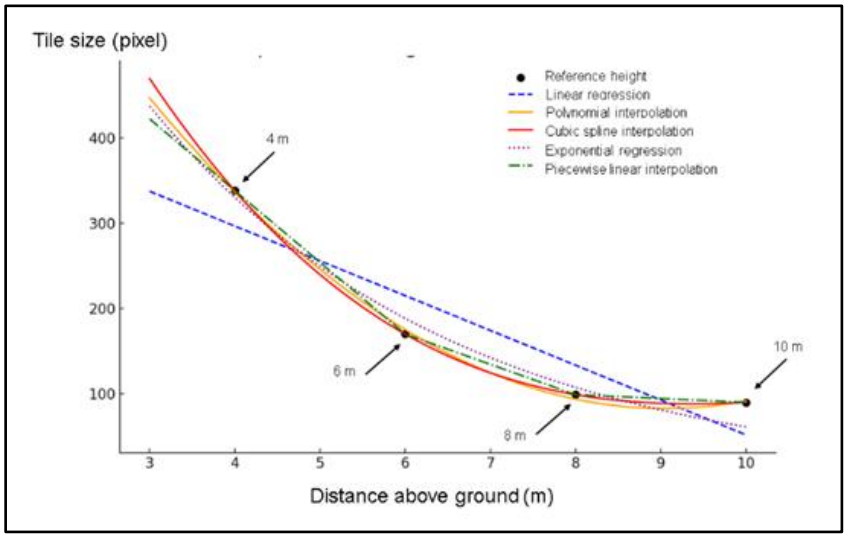



Figure 4.7 Comparing the Predicted Tile Size Obtained through Various Interpolation and Regression Methods Based on the Given Reference Height

Table 4.12
Quantitative Comparison of the Predicted Tile Size Obtained through Various Interpolation and Regression Methods Based on the Given Reference Height

| Height (m) | Reference Point (px) | Piecewise Linear Interpolation (px) | Polynomial Interpolation (px) | Cubic Spline Interpolation (px) | Linear Regression (px) | Exponential Regression (px) |
|---|---|---|---|---|---|---|
| 10 | 90 | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| 9 | | 95 | 83 | 89 | 93 | 81 |
| 8 | 99 | | | | | |
| 7 | | 135 | 125 | 125 | 174 | 142 |
| 6 | 170 | | | | | |
| 5 | | 254 | 246 | 240 | 256 | 249 |
| 4 | 338 | | | | | |
| 3 | | 422 | 447 | 469 | 337 | 437 |

The main purpose of incorporating validation images was to determine the optimal way to predict tile size by estimating how different altitudes of the UAV affect the capture of pictures and videos. This step is important to confirm that this prediction tile-size method for recognizing and detecting pineapple remains valid across different UAV flight altitudes, because image scale and pineapple sizes will be significantly different when the drone is flying at different heights.

#### *4.4.3.1 Piecewise Linear Interpolation*

Using the piecewise linear interpolation in derived tile sizes in Table 4.13, the Faster R-CNN model shows moderate detection capability under the interpolated tile size configuration. While high recall values are achieved at certain heights, the corresponding precision remains relatively low, resulting in only moderate F1-score and mAP values. This indicates that Faster R-CNN can identify most pineapple instances but still produces a noticeable number of false positives, particularly as tile sizes increase at lower altitudes.

In contrast, SSD exhibits consistently inferior performance across all evaluated heights. Lowest performance metrics of precision, recall, F1-score, and mAP values suggest that SSD struggles to effectively handle scale variation and detect pineapple with background complexity. These results imply limited robustness of SSD in based pineapple detection tasks.

Conversely, YOLOv8 achieves the most robust and consistent performance with perfect values of precision, recall, F1-score, and mAP maintained across all tested heights, indicating that YOLOv8 effectively leverages the DSSN to detect pineapple instances without degradation due to scale changes. This superior performance

highlights YOLOv8 as the most suitable model for integration with piecewise linear interpolation and tile size estimation.

Table 4.13
Evaluation of DLM Performance in Predicting Tile Size Based on Height Measurements Using Piecewise Linear Interpolation

| | Height (m) | Tile size (px) | PRE | REC | F1-score | mAP |
|---|---|---|---|---|---|---|
| Faster R-CNN | 9 | 95 | 0.560 | 0.890 | 0.690 | 0.409 |
| | 7 | 135 | 0.510 | 0.930 | 0.660 | 0.395 |
| | 5 | 254 | 0.680 | 0.760 | 0.720 | 0.544 |
| | 3 | 422 | 0.800 | 0.440 | 0.570 | 0.300 |
| SSD | 9 | 95 | 0.55 | 0.49 | 0.52 | 0.314 |
| | 7 | 135 | 0.44 | 0.32 | 0.37 | 0.288 |
| | 5 | 254 | 0.44 | 0.37 | 0.40 | 0.285 |
| | 3 | 422 | 0.35 | 0.67 | 0.46 | 0.170 |
| YOLOv8 | 9 | 95 | 1 | 1 | 1 | 1 |
| | 7 | 135 | 1 | 1 | 1 | 1 |
| | 5 | 254 | 1 | 1 | 1 | 1 |
| | 3 | 422 | 1 | 1 | 1 | 1 |

#### *4.4.3.2 Polynomial Interpolation*

Using polynomial interpolation for tile size estimation, as shown in Table 4.14, the Faster R-CNN model exhibits the following performance metrics. Although improved recall is observed at certain heights, such as 0.95 at 7 m, the precision values remain relatively low, resulting in moderate F1-score and mAP values. This behavior suggests that Faster R-CNN can detect a large proportion of pineapple instances but still has false detections when larger tile sizes are generated at lower flight altitudes.

Similarly, SSD continues to demonstrate modest performance across all evaluated heights. It shows that SSD struggles to generalize effectively at detecting pineapples, consistently achieving low precision, recall, F1-score, and mAP values. Thus, these results confirm the limited capability of SSD in handling scale variation and background complexity in agricultural imagery, even when polynomial interpolation is used to optimize tile size.

In contrast, YOLOv8 maintains outstanding and consistent performance across all heights with the given prediction tile sizes. The model achieves perfect precision,

recall, F1-score, and mAP, with all performance metrics set to 1, demonstrating strong robustness in detecting pineapple.

Table 4.14
Evaluation of DLM Performance in Predicting Tile Size Based on Height Measurements Using Polynomial Linear

| | Height (m) | Tile size (px) | PRE | REC | F1-score | mAP |
|---|---|---|---|---|---|---|
| Faster R-CNN | 9 | 83 | 0.550 | 0.790 | 0.650 | 0.368 |
| | 7 | 125 | 0.480 | 0.950 | 0.640 | 0.370 |
| | 5 | 246 | 0.650 | 0.810 | 0.720 | 0.542 |
| | 3 | 447 | 1.000 | 0.560 | 0.710 | 0.440 |
| SSD | 9 | 83 | 0.45 | 0.53 | 0.48 | 0.179 |
| | 7 | 125 | 0.44 | 0.32 | 0.37 | 0.288 |
| | 5 | 246 | 0.55 | 0.43 | 0.48 | 0.389 |
| | 3 | 447 | 0.27 | 0.44 | 0.33 | 0.120 |
| YOLOv8 | 9 | 83 | 1 | 1 | 1 | 1 |
| | 7 | 125 | 1 | 1 | 1 | 1 |
| | 5 | 246 | 1 | 1 | 1 | 1 |
| | 3 | 447 | 1 | 1 | 1 | 1 |

#### *4.4.3.3 Cubic Spline Interpolation*

The detection performance of Faster R-CNN, SSD, and YOLOv8 was evaluated for prediction tile size estimation using cubic spline interpolation across multiple UAV flight heights, as shown in Table 4.15. For Faster R-CNN, moderate detection performance is observed across all heights. At 9 m, the model achieves a precision of 0.55, a recall of 0.86, an F1-score of 0.67, and an mAP of 0.40. A higher recall of 0.95 is achieved at 7 m, but the lower precision limits the F1-score and mAP. The performance improves slightly at 5 m but drops at 3 m due to low recall and F1 scores.

In the case of SSD, overall performance remains comparatively low and inconsistent. At 9 m, precision and recall are approximately balanced at around 0.52 and 0.51, yielding an F1-score of 0.51 and mAP of 0.25. The performance declines further at 7 m, where both recall and F1-score fall below 0.40. A slight improvement is observed at 5 m with an F1-score of 0.47 and mAP of 0.33, while recall increases to 0.89 but precision remains low, resulting in only a moderate F1-score of 0.55 and mAP of 0.24 at 3 m. These results demonstrate that SSD is highly sensitive to tile-size variations introduced by cubic spline interpolation.

Conversely, YOLOv8 demonstrates consistently superior performance across all evaluated heights. The model achieves perfect precision, recall, F1-score, and mAP, with all values being 1. This indicates that YOLOv8 effectively exploits the optimal tile size produced by cubic spline interpolation, maintaining robust detection performance regardless of scale variations.

Table 4.15
Evaluation of DLM Performance in Predicting Tile Size Based on Height Measurements Using Cubic Spline Interpolation

| | Height (m) | Tile size (px) | PRE | REC | F1-score | mAP |
|---|---|---|---|---|---|---|
| Faster R-CNN | 9 | 89 | 0.550 | 0.860 | 0.670 | 0.400 |
| | 7 | 125 | 0.480 | 0.950 | 0.640 | 0.370 |
| | 5 | 240 | 0.620 | 0.780 | 0.690 | 0.506 |
| | 3 | 469 | 0.750 | 0.330 | 0.460 | 0.450 |
| SSD | 9 | 89 | 0.52 | 0.51 | 0.51 | 0.253 |
| | 7 | 125 | 0.44 | 0.32 | 0.37 | 0.288 |
| | 5 | 240 | 0.49 | 0.44 | 0.47 | 0.333 |
| | 3 | 469 | 0.40 | 0.89 | 0.55 | 0.240 |
| YOLOv8 | 9 | 89 | 1 | 1 | 1 | 1 |
| | 7 | 125 | 1 | 1 | 1 | 1 |
| | 5 | 240 | 1 | 1 | 1 | 1 |
| | 3 | 469 | 1 | 1 | 1 | 1 |

#### *4.4.3.4 Linear Regression*

The detection performance of Faster R-CNN, SSD, and YOLOv8 under linear regression for predicting tile size in Table 4.16 shows the largest performance degradation among the interpolation methods. The results indicate that linear regression is less capable of capturing the non-linear relationship between height and prediction tile size.

For Faster R-CNN, moderate performance is observed at 9 m and 5 m, where F1-scores of 0.65 and 0.78 are achieved, respectively. However, a noticeable decline occurs at 7 m and 3 m, with the F1-score dropping to 0.59 and 0.53, respectively. Although the mAP peaks at 0.58 at 5 m, the overall performance remains inconsistent across heights, indicating limited robustness of Faster R-CNN under linear regression for estimated tile sizes.

For SSD, linear regression produces the weakest results among all evaluated models and interpolation strategies. Precision and recall values fluctuate significantly across heights, with the F1-score remaining below 0.55 for all cases and dropping as low as 0.23 at 7 m and 3 m. The extremely low mAP value of 0.04 at 3 m further highlights SSD's inability to generalize effectively when tile sizes are estimated with a simple linear model.

Conversely, YOLOv8 continues to demonstrate perfect and stable performance, achieving precision, recall, F1-score, and mAP values of 1.0 across all heights for all prediction tile sizes. This indicates that YOLOv8 is highly robust to inaccuracies in tile-size estimation, even when a suboptimal regression approach is used.

Table 4.16
Evaluation of DLM Performance in Predicting Tile Size Based on Height Measurement Using Linear Regression

| | **Height (m)** | **Tile size (px)** | **PRE** | **REC** | **F1-score** | **mAP** |
|---|---|---|---|---|---|---|
| Faster R-CNN | 9 | 93 | 0.540 | 0.820 | 0.650 | 0.404 |
| | 7 | 142 | 0.460 | 0.810 | 0.590 | 0.253 |
| | 5 | 249 | 0.770 | 0.800 | 0.780 | 0.583 |
| | 3 | 437 | 0.500 | 0.560 | 0.530 | 0.130 |
| SSD | 9 | 93 | 0.60 | 0.51 | 0.55 | 0.295 |
| | 7 | 142 | 0.32 | 0.18 | 0.23 | 0.221 |
| | 5 | 249 | 0.39 | 0.24 | 0.30 | 0.261 |
| | 3 | 437 | 0.15 | 0.56 | 0.23 | 0.042 |
| YOLOv8 | 9 | 93 | 1 | 1 | 1 | 1 |
| | 7 | 142 | 1 | 1 | 1 | 1 |
| | 5 | 249 | 1 | 1 | 1 | 1 |
| | 3 | 437 | 1 | 1 | 1 | 1 |

#### *4.4.3.5 Exponential Regression*

The detection performance of Faster R-CNN, SSD, and YOLOv8 under the exponential regression for prediction cutting off tile size in Table 4.17 shows improved stability when compared to linear regression. For Faster R-CNN, relatively balanced performance is observed across all heights. At 9 m and 7 m, F1-scores of 0.66 are achieved with mAP values of 0.38 and 0.40, respectively. Performance improves at 5 m, where the F1-score increases to 0.73, and the mAP reaches 0.56, indicating better alignment between the estimated tile size and object scale. At 3 m, although precision

increases to 0.80, the recall drops to 0.44, resulting in a reduced F1-score of 0.57 and an mAP of 0.34.

For SSD, exponential regression yields slightly better performance than linear regression but remains limited overall. The F1-score ranges between 0.37 and 0.61 across different heights, with the highest F1-score observed at 3 m due to improved recall. However, mAP values remain relatively low, staying below 0.38 for all configurations. These results indicate that SSD remains sensitive to inaccuracies in tile size estimation, even when a non-linear regression approach is applied.

Conversely, YOLOv8 maintains consistently high detection performance under exponential regression. Perfect precision, recall, F1-score, and mAP are achieved at 9 m, 5 m, and 3 m, respectively. At 7 m, YOLOv8 records a near-perfect performance, with a precision of 0.95, recall of 1.0, F1-score of 0.97, and mAP of 0.95. This slight deviation suggests minimal sensitivity to the exponential regression to estimate tile size.

Table 4.17
Evaluation of DLM Performance in Predicting Tile Size Based on Height Measurement Using Exponential Regression

| | Height (m) | Tile size (px) | Precision | Recall | F1-score | mAP |
|---|---|---|---|---|---|---|
| Faster R-CNN | 9 | 81 | 0.560 | 0.800 | 0.660 | 0.380 |
| | 7 | 142 | 0.510 | 0.910 | 0.660 | 0.400 |
| | 5 | 249 | 0.670 | 0.800 | 0.730 | 0.556 |
| | 3 | 437 | 0.800 | 0.440 | 0.570 | 0.340 |
| SSD | 9 | 81 | 0.41 | 0.55 | 0.47 | 0.208 |
| | 7 | 142 | 0.44 | 0.32 | 0.37 | 0.288 |
| | 5 | 249 | 0.51 | 0.43 | 0.46 | 0.377 |
| | 3 | 437 | 0.50 | 0.78 | 0.61 | 0.190 |
| YOLOv8 | 9 | 81 | 1 | 1 | 1 | 1 |
| | 7 | 142 | 0.95 | 1 | 0.97 | 0.948 |
| | 5 | 249 | 1 | 1 | 1 | 1 |
| | 3 | 437 | 1 | 1 | 1 | 1 |

#### *4.4.3.6 Summarise the Prediction Lines*

The comparative analysis revealed that both cubic spline and polynomial interpolation closely approximated the actual tile sizes throughout the measured range. Among these methods, cubic spline interpolation consistently demonstrates the most stable and reliable performance, followed closely by polynomial interpolation and

piecewise linear interpolation, which show moderate but acceptable consistency. The cubic spline interpolation exhibits the highest fidelity to the actual measurements, particularly by accurately modeling gradual transitions between points. Polynomial interpolation also performed satisfactorily but introduced minor deviations at the extremities, suggesting sensitivity to the edge behavior typical of polynomial models. In contrast, linear interpolation deviated more significantly, particularly between 5 m and 7 m, owing to its inability to model the nonlinear trends. In contrast, exponential regression provides mixed performance, offering improvements over linear regression but still exhibiting instability across certain heights. Linear regression emerges as the least suitable method, showing the highest performance degradation and inconsistency, particularly for Faster R-CNN and SSD. These findings underscore the importance of selecting interpolation methods that account for the data's underlying nonlinearities.

Interpolating methods that can preserve the nonlinear characteristics of data. Nonlinear interpolation methods, such as cubic spline interpolation, better preserve local variations and subtle structures, allowing for more accurate and robust predictions for new samples. In contrast, regression-based approaches strive to learn a global function over the dataset, which would be effective for modeling overall trends but might sometimes blur important local details. In this context, cubic spline interpolation demonstrated superior performance, providing smoother transitions and more accurate estimates. Polynomial interpolation offers a practical alternative but may require careful management of the polynomial degree to avoid overfitting or oscillatory behavior. The inferior performance of linear interpolation and exponential regression further highlights the limitations of overly simplistic models in complex relationships in data. Thus, interpolation methods that incorporate curvature and smoothness characteristics, particularly cubic splines, are recommended for accurate tile size prediction over different distances.

Across all five prediction tile-size methods, YOLOv8 consistently achieves the highest detection performance, achieving near-perfect or perfect precision, recall, F1-score, and mAP values in almost all configurations. This indicates that YOLOv8 is highly robust to variations in tile-size estimates, even when suboptimal prediction methods are used. Its strong multi-scale feature representation and efficient one-stage detection architecture enable reliable pineapple detection under diverse spatial slicing conditions. Consequently, the combination of YOLOv8 with a robust interpolation

strategy, particularly cubic spline interpolation, represents the most effective and reliable configuration for pineapple detection in this study.

## 4.5 Decarbonization Effect

### 4.5.1 Computational Time

The DSSN-based detection model and the inference performance of Faster R-CNN, SSD, and YOLOv8 were examined across flight heights ranging from 10 m to 3 m in 1 m increments, and the results are presented in Table 4.18. The results were evaluated in terms of frames per second (FPS) and average inference time per tile, with YOLOv8 consistently performing at the highest level, reporting 21.24 FPS and 37.8 s/tile, while also recording the lowest average time per tile, 0.026 s to 0.047 s. This suggests that YOLOv8 can handle a significantly high number of tile inferences per second, which is a requirement in real-time applications. In the case of Faster R-CNN, there is moderate, consistent computational performance across all altitudes, with an FPS between 10.88 and 15.8 and an average time per tile of 0.063 s to 0.092 s, suggesting a reasonable and balanced trade-off between processing speed and complexity. SSD, on the other hand, showed a much lower FPS range of 3.97 to 11.9 and a higher average time per tile of 0.084 s to 0.252 s. Thus, the combined analysis demonstrates that YOLOv8 not only delivers the highest throughput but also minimizes per-tile inference time, making it the most energy-efficient and environmentally favorable model among those evaluated.

Although FPS and average time per tile are inversely related and describe the same inference behavior, they represent different perspectives: FPS captures the throughput-oriented processing speed, whereas the average time per tile explicitly quantifies the temporal cost of processing a single tile. This distinction is particularly important for decarbonization analysis, as carbon emissions are directly associated with the duration of computational activity rather than the processing rate, making the average time per tile a more suitable metric for estimating energy consumption and carbon footprint in tile-based detection models.

Table 4.18
Computational Time of DLM during Testing Time

| | 10m | 9m | 8m | 7m | 6m | 5m | 4m | 3m |
|---|---|---|---|---|---|---|---|---|

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Faster R-CNN | FPS | 12.77 | 15.25 | 12.47 | 15.23 | 13.79 | 15.8 | 13.76 | 10.88 |
| | Average Time per Tile | 0.078 | 0.066 | 0.08 | 0.066 | 0.073 | 0.063 | 0.073 | 0.092 |
| SSD | FPS | 4.84 | 7.41 | 4.69 | 10.89 | 3.97 | 10.54 | 11.9 | 5.12 |
| | Average Time per Tile | 0.207 | 0.135 | 0.213 | 0.092 | 0.252 | 0.095 | 0.084 | 0.195 |
| YOLOv8 | FPS | 23.19 | 25.46 | 24.67 | 25.333 | 26.59 | 24.56 | 37.8 | 21.24 |
| | Average Time per Tile | 0.043 | 0.039 | 0.041 | 0.039 | 0.038 | 0.041 | 0.026 | 0.047 |

#### *4.5.1.1 Computational Speed Analysis*

The utility reflects the processing capability of each detection model in terms of throughput and is expressed in Mega per second (M $s^{-1}$) to accommodate the large numerical scale of the measured values. The utility incorporates three key factors: FPS, average inference time per tile, and total tile generation, where FPS represents the tile-level processing speed, the average time per tile captures the temporal cost of processing a single tile, and the total tile generation accounts for the overall DSSN workload. The total number of generated tiles required for this calculation is listed in Table 4.19. Among the evaluated models, YOLOv8 exhibited a significantly higher utility value of 66.8515 M $s^{-1}$, indicating a superior throughput and an enhanced ability to process a large number of tiles efficiently. In contrast, Faster R-CNN recorded a utility of 0.1274 M $s^{-1}$. In contrast, SSD achieved the lowest utility at 0.0311 M $s^{-1}$, reflecting comparatively limited processing throughput due to longer per-tile inference times and reduced tile-processing speed.

Efficiency evaluates how effectively each model utilizes computational resources by jointly considering the FPS, average time per tile, and total tile generation, thereby capturing both the processing speed and workload sensitivity. As summarized in Table 4.19, YOLOv8 achieved the highest efficiency score of 82.1278, substantially outperforming Faster R-CNN of 10.9391 and SSD of 3.2738. This result indicates that YOLOv8 not only processes tiles at a higher rate but also maintains a lower temporal and computational overhead per tile, even when accounting for the total number of tiles

generated, as detailed in Table 4.19. Faster R-CNN demonstrated moderate efficiency, indicating a trade-off between computational complexity and throughput. In contrast, SSD showed the lowest efficiency, reflecting higher per-tile processing time and less effective use of computational resources.

Table 4.19
Computational Speed Analysis for DLM

| | **Faster R-CNN** | **SSD** | **YOLOv8** |
|---|---|---|---|
| Utility (M $s^{-1}$) | 0.1274 | 0.0311 | 66.8515 |
| Efficiency | 10.9391 | 3.2738 | 82.1278 |

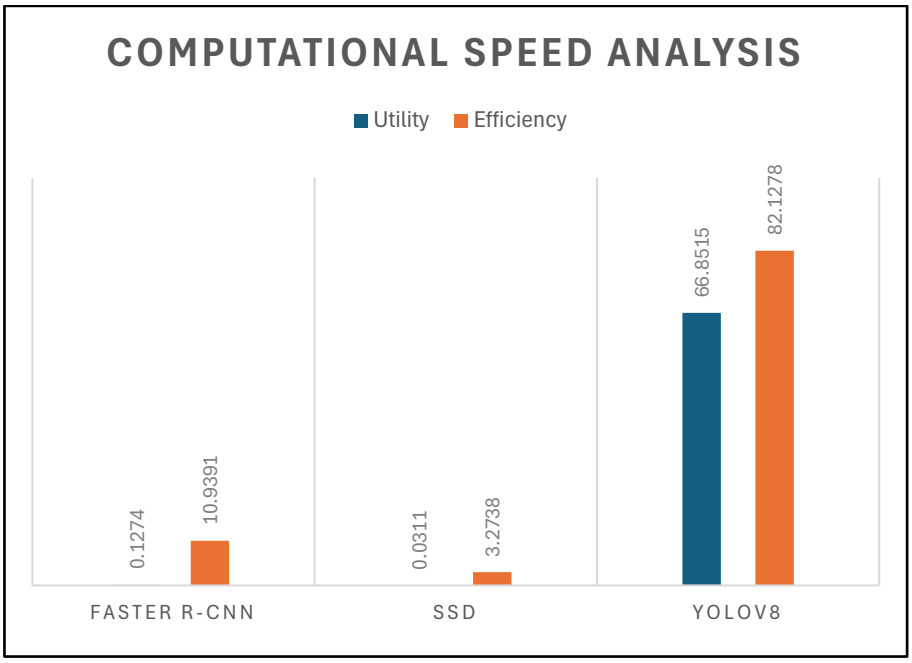



Figure 4.8 Utility and Efficiency Value for DLM

The combined evaluation of utility and efficiency provides a holistic understanding of the model performance beyond detection accuracy alone. Utility emphasizes the model's ability to handle large-scale tile processing workloads efficiently, which is crucial for high-throughput and real-time DSSN applications. Efficiency, on the other hand, highlights how well computational resources are exploited to achieve this throughput, making it particularly relevant for energy-aware and decarbonization-focused studies. By integrating FPS, the average time per tile, and total tile generation into these metrics, the analysis ensures a fair and consistent comparison of the intrinsic model performance while accounting for workload variations induced by different slicing configurations. Consequently, models exhibiting

high utility and efficiency, such as YOLOv8, are better suited for scalable, computationally efficient, and environmentally sustainable tile-based detection.

### 4.5.2 Carbon Release

The carbon release analysis was conducted using constant parameters: energy efficiency (EFF = $6.2 \times 10^{-12}$ J/FLOP), memory energy ($E_m$ = 0.025 J), and carbon intensity (CI = 0.52 $kgCO_2e/kWh$), as summarized in Table 4.20. These parameters were held constant across all experiments to ensure a fair and consistent comparison of carbon emissions among deep learning models. Meanwhile, Table 4.19 reports the resulting carbon footprint (CF) values computed for Faster R-CNN, SSD, and YOLOv8 across multiple UAV flight altitudes, incorporating model-specific FPS, total tile generation, and computational complexity, measured in GFLOPs. In the context of carbon release, lower CF values indicate better environmental performance, as they correspond to reduced energy consumption and lower carbon emissions. Conversely, higher CF values indicate increased computational workload and prolonged hardware activity, leading to greater carbon emissions. Therefore, minimizing CF is a key objective in the design of decarbonization-oriented deep learning systems.

Across all evaluated altitudes, SSD consistently produced the lowest carbon release, with CF values remaining below 4.1 and reaching as low as 0.1319 at lower altitudes. This behavior is primarily attributed to SSD's relatively low computational complexity (GFLOPs = 0.62) and moderate tile generation, resulting in reduced energy consumption despite its lower processing speed. Faster R-CNN exhibited moderate carbon emissions, with CF values decreasing significantly as flight altitude decreased and total tile generation was reduced. At higher altitudes, Faster R-CNN recorded CF values above 45. In contrast, at lower altitudes, the CF dropped below 2, indicating that tile density and slicing configuration strongly influenced carbon emissions for two-stage detectors. In contrast, YOLOv8 exhibited the highest carbon release at higher altitudes, particularly at 10 m, where the CF reached 570.5525, largely owing to the extremely high number of generated tiles combined with increasing GFLOPs. However, as the altitude decreased and tile generation became more controlled, YOLOv8's CF dropped sharply, reaching values comparable to or even lower than those of Faster R-CNN at lower altitudes. This demonstrates that although YOLOv8 is highly efficient in terms of speed, its carbon footprint is sensitive to the tile generation density and

computational workload. When carbon emissions are the primary concern, SSD emerges as the model with the lowest environmental impact, followed by Faster R-CNN, whereas YOLOv8 requires careful configuration to prevent excessive emissions under high tile-density conditions.

Table 4.20
Carbon Release of DLM

| | | 10m | 9m | 8m | 7m | 6m | 5m | 4m | 3m |
|---|---|---|---|---|---|---|---|---|---|
| Faster R-CNN | Average Inference Time per Tile (s) | 0.078 | 0.066 | 0.080 | 0.066 | 0.073 | 0.063 | 0.073 | 0.092 |
| | total tile | 1300 | 1274 | 1274 | 630 | 496 | 180 | 128 | 45 |
| | GFLOPs | 134.73 | 134.73 | 134.73 | 134.73 | 134.73 | 134.73 | 134.73 | 134.73 |
| | CF | 0.0603 | 0.0699 | 0.0577 | 0.0345 | 0.0246 | 0.0103 | 0.0063 | 0.0018 |
| SSD | Average Inference Time per Tile (s) | 0.207 | 0.135 | 0.213 | 0.092 | 0.252 | 0.095 | 0.084 | 0.195 |
| | Total tile | 1300 | 1274 | 1274 | 630 | 496 | 180 | 120 | 45 |
| | GFLOPs | 0.62 | 0.62 | 0.62 | 0.62 | 0.62 | 0.62 | 0.62 | 0.62 |
| | CF | 0.0227 | 0.0341 | 0.0216 | 0.0247 | 0.0071 | 0.0068 | 0.0052 | 0.0008 |
| YOLOv8 | Average Inference Time per Tile (s) | 0.043 | 0.039 | 0.041 | 0.039 | 0.038 | 0.041 | 0.026 | 0.047 |
| | total tile | 983520 | 2886 | 1484 | 1274 | 594 | 231 | 144 | 66 |
| | GFLOPs | 0.14 | 0.14 | 0.32 | 0.32 | 0.89 | 1.75 | 3.58 | 7.01 |
| | CF | 82.5955 | 0.2672 | 0.1307 | 0.1180 | 0.0564 | 0.0203 | 0.0200 | 0.0051 |

The decarbonization effect involves reducing carbon emissions through continued improvements in the optimization of computational processes, architectural models, and workload flow management. In a deep learning-based detection system, the effect of decarbonization depends on model complexity, inference time, and total computational workload, as these factors determine the power consumption and the time the hardware is engaged. The results show that deep learning models, even when performing the same detection task, release different amounts of carbon. Deep learning models (DLMs) contribute to the results and have different impacts on the definition of

decarbonization. Some models have relatively low computational complexity and require less tile-level processing than expected, which means they will consume less power and result in lower carbon emissions. On the other hand, models with high complexity are expected to produce higher carbon emissions, even when they run very fast. The results show that DLMs with low carbon emissions will contribute positively to global sustainability. Improvements will be made as computer workloads will have lower carbon emissions, reducing environmental impact by lowering greenhouse gas emissions and electricity use. It is possible to achieve a good balance between delivering reliable detection to support environmental sustainability and using energy-efficient detection models thanks to optimized tile-based processing.

There is an inherent need to integrate the potential for decarbonization into the design and evaluation of deep learning-based detection systems. As far as the future of AI systems is concerned, especially those that involve processing extremely large datasets, performing endless inferences, and making real-time decisions, the potential for carbon emissions becomes an equally important metric alongside accuracy and speed. Through the careful selection of models and workload management, the potential for carbon emissions can be diminished, thereby representing a step in the right direction for the effective management of artificial intelligence. In this case, the use of DLMs associated with lower carbon emissions is a positive step towards achieving not only computational resource savings but also the ecological goal of reducing the carbon footprint of deploying AI systems, and, consequently, positive societal benefits. Thus, carbon-aware model selection and evaluation must be prioritized in decarbonization strategies and must become a key consideration in the design of AI systems.

# CHAPTER 5
# CONCLUSION

## 5.1 Conclusion

This study developed deep spatial slicing networks to recognize pineapples in agricultural fields, demonstrating that tile sizes derived from ground truth bounding boxes substantially improved detection accuracy compared to standard bounding boxes derived from training images. Owing to its close alignment with actual measurements, the tile size based on ground-truth bounding boxes consistently outperformed the other methods across all evaluation metrics. For the validation models, the evaluation of interpolation and regression methods for predicting tile sizes at heights between 3m and 9m revealed that cubic spline interpolation provided the most accurate estimates for recognizing pineapple in complex backgrounds. Polynomial interpolation performed reasonably well but introduced minor edge deviations, whereas linear interpolation and both linear and exponential regression were less effective at capturing the data's nonlinear patterns.

The performance of the proposed model was evaluated by comparing it with baseline deep learning models, including Faster R-CNN, SSD, and YOLOv8, across drone heights from 3m to 10m to optimize pineapple detection. This evaluation used specific performance measures, including precision, recall, F1-score, and mean average precision, to assess the reliability of pineapple detection in complex backgrounds. Among the evaluated models, YOLOv8 consistently demonstrated the best performance as a convolutional deep learning model across all heights, owing to its highly efficient single-stage architecture, which excels at extracting multi-scale object features. In this model, only Faster R-CNN required external backbone integration for detection, whereas SSD and YOLOv8 used their built-in backbones. Therefore, several pretrained deep learning backbones were evaluated specifically for Faster R-CNN, namely ResNet-50, ResNet-101, Inception-ResNet-v2, and VGG-19. While deeper architectures like ResNet-101 and Inception-ResNet-v2 struggled in crowded, overlapping environments, the VGG-19 backbone combined with Faster R-CNN performed best, significantly improving detection accuracy for pineapples in complex backgrounds.

Finally, this study developed the Decarbonized Efficacy Algorithm (DEA) by monitoring the processing speeds of Faster R-CNN, SSD, and YOLOv8 to evaluate how model runtime directly affects energy consumption and carbon emissions. The results showed that YOLOv8 achieved the best outcome because it required the shortest testing time and produced the lowest carbon release. Since lighter networks process high-resolution drone images much faster, they significantly reduce GPU energy demand. The integration of this DEA ensures that achieving high precision in automated pineapple detection does not compromise environmental sustainability, offering a balanced solution that minimizes carbon footprint.

## 5.2 Future Study

The results of this study indicate great potential for improving pineapple detection while droning at different altitudes, as well as for filtering background noise and improving image quality, enabling drone images to be classified effectively. It is recommended that further studies expand the current model to other fruit crops with different densities and morphological attributes. To improve the model's flexibility for general agricultural monitoring, assessing the DSSN approach and its effectiveness across different lighting and meteorological conditions, as well as drone flight paths, would improve its defensibility.

This study suggests investigating hybrid or ensemble detection models that combine the complementary exploitation of anchor-based and anchor-free strategies across different deep learning methods. Hence, it is suggested that future work use the same deep learning pattern models, such as the YOLO family, to detect fruits in complex backgrounds. This is important in cases of high occlusion or high-density objects, and in cases of irregularity in pineapple morphologies.

In this thesis, the cubic spline interpolation line proved highly effective at predicting tile sizes in the middle-altitude range. Hence, real-time interpolation or predictive models inside drones are recommended for future studies. It becomes highly effective because Gaussian processes or lightweight regression models can predict optimal tile sizes as altitude changes. Thus, the object can be localized more accurately without cumbersome ground-truth annotations for every layer of drone flight.

Ultimately, when these automated techniques are fully realized and deployed smoothly in real-time, they will provide substantial, practical benefits to farmers and

the wider agricultural industry. By integrating high-accuracy object detection with lightweight processing, this system can be embedded into smart harvesting equipment or drone-assisted monitoring tools. This will allow farmers to easily identify the exact locations of ripe pineapples and assess their fruit maturity values directly from aerial data. Consequently, it simplifies the harvesting process, reduces manual labor costs, and optimizes crop yield management by ensuring that only fully mature pineapples are harvested at the right time.

# REFERENCES


Agarwal, A., Singh, A. K., Kumar, S., & Singh, D. (2018). Critical analysis of classification techniques for precision agriculture monitoring using satellite and drone. *2018 13th International Conference on Industrial and Information Systems, ICIIS 2018 - Proceedings*. https://doi.org/10.1109/ICIINFS.2018.8721422

Al-shareeda, M. A., Manickam, S., Mohammed, B. A., Al-mekhlafi, Z. G., Qtaish, A., Alzahrani, A. J., & Alshammari, G. (2022). *Provably Secure with Efficient Data Sharing Scheme for Fifth-Generation ( 5G ) -Enabled Vehicular Networks without Road-Side Unit ( RSU )*.

Alaieri, F. (2024). Precision Agriculture based on Machine Learning and Remote Sensing Techniques. *Engineering, Technology & Applied Science Research*, *14*(3), 14206–14211.

Alghamdi, H., & Turki, T. (2023). PDD-Net: Plant Disease Diagnoses Using Multilevel and Multiscale Convolutional Neural Network Features. *Agriculture (Switzerland)*, *13*(5). https://doi.org/10.3390/agriculture13051072

Ali, M. M., Hashim, N., Bejo, S. K., Jahari, M., & Shahabudin, N. A. (2023). Innovative non-destructive technologies for quality monitoring of pineapples: Recent advances and applications. *Trends in Food Science & Technology*, *133*, 176–188.

Alibabaei, K., Assunção, E., Gaspar, P. D., Caldeira, J. M. L. P., & Soares, V. N. G. J. (2022). Real-Time Detection of Vine Trunk for Robot Localization Using Deep Learning Models Developed for Edge TPU Devices. *Future Internet*, *14*(7), 1–16. https://doi.org/10.3390/fi14070199

Alqethami, S., Almtanni, B., Alzhrani, W., & Alghamdi, M. (2022). Disease detection in apple leaves using image processing techniques. *Engineering, Technology & Applied Science Research*, *12*(2), 8335–8341.

An, J., Zhang, D., Xu, K., & Wang, D. (2022). An OpenCL-Based FPGA Accelerator for Faster R-CNN. *Entropy*, *24*(10), 1–18. https://doi.org/10.3390/e24101346

Azadbakht, M., Ashourloo, D., Aghighi, H., Radiom, S., & Alimohammadi, A. (2019). Wheat leaf rust detection at canopy scale under different LAI levels using machine learning techniques. *Computers and Electronics in Agriculture*,

*156*(January), 119–128. https://doi.org/10.1016/j.compag.2018.11.016

Bandara, W. A. I. U., Kuruppuarachchi, K. A. D. K. S., Maduwantha, N. N. D., Arachchillage, U. S. S. S., Alwis, T. A. D. T. N. D., & Kuruppu, T. A. (2022). Smart Intelligent Pineapple Farming Assistant Agent (SIPFAA). *2022 4th International Conference on Advancements in Computing (ICAC)*, 48–53. https://doi.org/10.1109/ICAC57685.2022.10025103

Bandyopadhyay, A., Ghosh, S., Bose, M., Kessi, L., & Gaur, L. (2023). Supervised Neural Networks for Fruit Identification. *Communications in Computer and Information Science*. https://doi.org/10.1007/978-3-031-23599-3_16

Cuong, N. H. H., Trinh, T. H., Meesad, P., & Nguyen, T. T. (2022). Improved YOLO object detection algorithm to detect ripe pineapple phase. *Journal of Intelligent and Fuzzy Systems*. https://doi.org/10.3233/JIFS-213251

Cv, A., & Lin, M. (2022). *Bacterial Leaf Spot of Tomato ( Solanum lycopersicum ) detection using Image Processing and Convolution Neural Network Bacterial Leaf Spot of Tomato ( Solanum lycopersicum ) detection using Image Processing and Convolution Neural Network*. 0–14.

Feng, H. (2024). The Evolution and Potential of Precision Agriculture in China Anchored In "3S" Technology. *Highlights in Science, Engineering and Technology*, *81*, 592–597. https://doi.org/10.54097/43n29y21

Gafurov, A. (2023). Automated Mapping of Cropland Boundaries Using Deep Neural Networks. *AgriEngineering*, *5*(3), 1568–1580. https://doi.org/10.3390/agriengineering5030097

Gonzalez-Hernandez, M., Fallas-Moya, F., Rodriguez-Montero, W., Xie-Li, D., Roman-Solano, B., Corrales-Garro, F., Sadovnik, A., & Qi, H. (2023). Evaluation of the Influence of Multispectral Imaging for Object Detection in Pineapple Crops. *2023 IEEE 5th International Conference on BioInspired Processing (BIP)*, 1–6.

Goumiri, S., Benboudjema, D., & Pieczynski, W. (2023). A new hybrid model of convolutional neural networks and hidden Markov chains for image classification. *Neural Computing and Applications*, *35*(24), 17987–18002. https://doi.org/10.1007/s00521-023-08644-4

Gupta, U., Kim, Y. G., Lee, S., Tse, J., Lee, H. H. S., Wei, G. Y., Brooks, D., & Wu, C. J. (2022). Chasing Carbon: The Elusive Environmental Footprint of Computing. *IEEE Micro*, *42*(4), 37–47.

https://doi.org/10.1109/MM.2022.3163226

Henderson, P., & Brunskill, E. (2020). *Towards the Systematic Reporting of the Energy and Carbon Footprints of Machine Learning*. *21*, 1–43.

Hindarto, D., Afarini, N., & Esthi H, E. T. (2023). Comparison Efficacy of VGG16 and VGG19 Insect Classification Models. *JIKO (Jurnal Informatika Dan Komputer)*, *6*(3), 189–195. https://doi.org/10.33387/jiko.v6i3.7008

Hobbs, J., Prakash, P., Paull, R., Hovhannisyan, H., Markowicz, B., & Rose, G. (2021). Large-Scale Counting and Localization of Pineapple Inflorescence Through Deep Density-Estimation. *Frontiers in Plant Science*, *11*.

Huo, L., Mao, J., San, H., Wang, S., & Li, R. (2023). A New Benchmark for Vibration Displacement Detection of Rotor. *IEEE Transactions on Instrumentation and Measurement*, *72*, 1–11. https://doi.org/10.1109/TIM.2023.3272040

Imaging, M., & Learning, D. (2024). *Intelligent Integrated System for Fruit Detection Using*.

Jelinek, T., Bhave, A., Buchoud, N., Bühler, M. M., Glauner, P., Inderwildi, O., Kraft, M., Mok, C., Nübel, K., & Voss, A. (2024). International Collaboration: Mainstreaming Artificial Intelligence and Cyberphysical Systems for Carbon Neutrality. *IEEE Transactions on Industrial Cyber-Physical Systems*, *2*, 26–34. https://doi.org/10.1109/ticps.2024.3351624

Jia, C., Wang, D., Liu, J., & Deng, W. (2024). Performance Optimization and Application Research of YOLOv8 Model in Object Detection. *Academic Journal of Science and Technology*, *10*(1), 325–329. https://doi.org/10.54097/p9w3ax47

Jiang, Z., & Wang, R. (2020). Underwater Object Detection Based on Improved Single Shot MultiBox Detector. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3446132.3446170

Kaur, A., Kukreja, V., Aeri, M., Tanwar, S., & Mohd, N. (2023). Nature's Secrets Revealed: Unraveling Fusarium Wilt Diseases through CNN and SVM. *2023 4th IEEE Global Conference for Advancement in Technology, GCAT 2023*, 1–7. https://doi.org/10.1109/GCAT59970.2023.10353538

Kaur, A., Kukreja, V., Tiwari, P., Manwal, M., & Sharma, R. (2024). Fruitful Fusion: An Accuracy-Boosting Ensemble of VGG19 and Convolutional Neural Networks for Dragon Fruit Classification. *2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social*

*Innovation, IATMSI 2024*, *2*, 1–5. https://doi.org/10.1109/IATMSI60426.2024.10502785

Kaur, P., Harnal, S., Tiwari, R., Upadhyay, S., Bhatia, S., Mashat, A., & Alabdali, A. M. (2022). Recognition of Leaf Disease Using Hybrid Convolutional Neural Network by Applying Feature Reduction. *Sensors*, *22*(2), 1–14. https://doi.org/10.3390/s22020575

Kim, G., Lim, H., Kim, Y., Kwon, O., & Choi, J. H. (2023). Intra-person multi-task learning method for chronic-disease prediction. *Scientific Reports*, *13*(1), 1–10. https://doi.org/10.1038/s41598-023-28383-9

Kim, J. Il, Baek, J. W., & Kim, C. B. (2022). Image Classification of Amazon Parrots by Deep Learning: A Potentially Useful Tool for Wildlife Conservation. *Biology*, *11*(9). https://doi.org/10.3390/biology11091303

Koklu, M., Cinar, I., & Taspinar, Y. S. (2021). *Computers ! and ! Electronics ! in ! Agriculture ! Classi " cation ! of ! rice ! varieties ! with ! deep ! learning ! methods !*

Kukunuri, A. N. J., & Singh, D. (2022). Efficient Application of Drone with Satellite data for Early-Stage Wheat Detection: For Precision Agriculture Monitoring. *International Geoscience and Remote Sensing Symposium (IGARSS)*, *2022-July*, 4388–4391. https://doi.org/10.1109/IGARSS46834.2022.9883266

Kundur, N. C., & Mallikarjuna, P. B. (2022). Deep convolutional neural network architecture for plant seedling classification. *Engineering, Technology & Applied Science Research*, *12*(6), 9464–9470.

Li, D., Wang, R., Xie, C., Liu, L., Zhang, J., Li, R., Wang, F., Zhou, M., & Liu, W. (2020). A recognition method for rice plant diseases and pests video detection based on deep convolutional neural network. *Sensors (Switzerland)*, *20*(3). https://doi.org/10.3390/s20030578

Li, X., Cai, C., Song, B., & Yang, Y. (2025). YOLOV8-MR: An Improved Lightweight YOLOv8 Algorithm for Tomato Fruit Detection (April 2024). *IEEE Access*, *13*(December 2024). https://doi.org/10.1109/ACCESS.2025.3533489

Li, Y., He, L., Zhang, M., Cheng, Z., Liu, W., & Wu, Z. (2023). Improving the Performance of the Single Shot Multibox Detector for Steel Surface Defects with Context Fusion and Feature Refinement. *Electronics (Switzerland)*, *12*(11). https://doi.org/10.3390/electronics12112440

Li, Y., Li, Q., Pan, J., Zhou, Y., Zhu, H., Wei, H., & Liu, C. (2024). SOD-YOLO: Small-

Object-Detection Algorithm Based on Improved YOLOv8 for UAV Images. *Remote Sensing*, *16*(16), 1–26. https://doi.org/10.3390/rs16163057

Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2022). A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(12), 6999–7019. https://doi.org/10.1109/TNNLS.2021.3084827

Lin, J., Yu, D., Pan, R., Cai, J., Liu, J., Zhang, L., Wen, X., Peng, X., Cernava, T., Oufensou, S., Migheli, Q., Chen, X., & Zhang, X. (2023). Improved YOLOX-Tiny network for detection of tobacco brown spot disease. *Frontiers in Plant Science*, *14*(February), 1–9. https://doi.org/10.3389/fpls.2023.1135105

Liu, J., & Wang, X. (2024). Multisource information fusion method for vegetable disease detection. *BMC Plant Biology*, *24*(1), 1–19. https://doi.org/10.1186/s12870-024-05346-4

Ma, K., Meng, X., Hao, M., Huang, G., Hu, Q., & He, P. (2023). Research on the Efficiency of Bridge Crack Detection by Coupling Deep Learning Frameworks with Convolutional Neural Networks. *Sensors*, *23*(16). https://doi.org/10.3390/s23167272

Magabilin, M. C. V., Fajardo, A. C., & Medina, R. P. (2022). Optimal Ripeness Classification of the Philippine Guyabano Fruit using Deep Learning. *ICPC2T 2022 - 2nd International Conference on Power, Control and Computing Technologies, Proceedings*, 1–5. https://doi.org/10.1109/ICPC2T53885.2022.9777014

Magalhães, S. A., Castro, L., Moreira, G., Dos Santos, F. N., Cunha, M., Dias, J., & Moreira, A. P. (2021). Evaluating the single-shot multibox detector and yolo deep learning models for the detection of tomatoes in a greenhouse. *Sensors*, *21*(10), 1–24. https://doi.org/10.3390/s21103569

Mazzia, V., Comba, L., Khaliq, A., Chiaberge, M., & Gay, P. (2020). UAV and machine learning based refinement of a satellite-driven vegetation index for precision agriculture. *Sensors (Switzerland)*, *20*(9). https://doi.org/10.3390/s20092530

Meng, F., Li, J., Zhang, Y., Qi, S., & Tang, Y. (2023). Transforming unmanned pineapple picking with spatio-temporal convolutional neural networks. *Computers and Electronics in Agriculture*, *214*, 108298.

Mohapatra, D., Das, N., Mohanty, K. K., & Shresth, J. (2022). Automated Visual Inspecting System for Fruit Quality Estimation Using Deep Learning. *Lecture*

*Notes in Electrical Engineering*. https://doi.org/10.1007/978-981-16-7076-3_33

Mohd Ali, M., Hashim, N., Abd Aziz, S., & Lasekan, O. (2022). Characterisation of Pineapple Cultivars under Different Storage Conditions Using Infrared Thermal Imaging Coupled with Machine Learning Algorithms. In *Agriculture* (Vol. 12, Issue 7). https://doi.org/10.3390/agriculture12071013

Nagaraju, Y., Venkatesh, & Venugopal, K. R. (2022). A Fruit Detection Method for Vague Environment High-Density Fruit Orchards. *2022 IEEE 3rd Global Conference for Advancement in Technology, GCAT 2022*, 1–6. https://doi.org/10.1109/GCAT55367.2022.9971938

Nasution, M. D., Al-Khowarizmi, A.-K., & Maulana, H. (2023). Optimization of Faster R-CNN to Detect SNI Masks at Mandatory Mask Doors. *Journal of Informatics and Telecommunication Engineering*, *6*(2), 602–611. https://doi.org/10.31289/jite.v6i2.8128

Nugroho, D. P., Widiyanto, S., & Wardani, D. T. (2022). Comparison of Deep Learning-Based Object Classification Methods for Detecting Tomato Ripeness. *International Journal of Fuzzy Logic and Intelligent Systems*, *22*(3), 223–232. https://doi.org/10.5391/IJFIS.2022.22.3.223

Oghaz, M. M. D., Razaak, M., & Remagnino, P. (2022). Enhanced Single Shot Small Object Detector for Aerial Imagery Using Super-Resolution, Feature Fusion and Deconvolution. *Sensors*, *22*(12), 1–22. https://doi.org/10.3390/s22124339

Olaoye, F., & Potter, K. (2024). Deep Learning Algorithms and Applications. *Dissolution Technologies*, *March*.

Pan, S., & Ahamed, T. (2022). Pear Recognition in an Orchard from 3D Stereo Camera Datasets to Develop a Fruit Picking Mechanism Using Mask R-CNN. *Sensors*, *22*(11). https://doi.org/10.3390/s22114187

Patil, M. S., & Mane, P. B. (2023). Fused Image Classification using Pre-trained Deep Convolutional Neural Networks. *Proceedings of the 3rd International Conference on Artificial Intelligence and Smart Energy, ICAIS 2023*. https://doi.org/10.1109/ICAIS56108.2023.10073743

Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L., Rothchild, D., So, D., Texier, M., & Dean, J. (n.d.). *Carbon Emissions and Large Neural Network Training*. 1–22.

Pham, T. D., Nguyen, D. T. H., Tran, D. H., & Tran, T. T. (2024). Application of satellite images and GIS to assess the site situation and propose solutions for

developing pineapple cultivation in U Minh Thuong, Kien Giang, Vietnam. *IOP Conference Series: Earth and Environmental Science*, *1306*(1), 12026.

Plested, J., & Gedeon, T. (2022). *Deep transfer learning for image classification: a survey*. http://arxiv.org/abs/2205.09904

Poliyapram, V., Wang, W., & Nakamura, R. (2019). A point-wise LiDAR and image multimodal fusion network (PMNet) for aerial point cloud 3D semantic segmentation. *Remote Sensing*, *11*(24). https://doi.org/10.3390/rs11242961

Prado, L., Paula, A., Ramos, M., Andr, L., Castro, D., Narges, S., Andrade, J. De, Takashi, E., Pistori, H., Nunes, W., & Li, J. (2021). *International Journal of Applied Earth Observations and Geoinformation A review on deep learning in UAV remote sensing*. *102*. https://doi.org/10.1016/j.jag.2021.102456

Qin, Q., Zhou, X., Gao, J., Wang, Z., Naer, A., Hai, L., Alatan, S., Zhang, H., & Liu, Z. (2025). YOLOv8-CBAM: a study of sheep head identification in Ujumqin sheep. *Frontiers in Veterinary Science*, *12*(February), 1–10. https://doi.org/10.3389/fvets.2025.1514212

Qiu, G., Lu, H., Xu, W., Wang, C., Xu, S., Liang, X., & Fan, C. (2023). Nondestructive Detecting Maturity of Pineapples Based on Visible and Near-Infrared Transmittance Spectroscopy Coupled With Machine Learning Methodologies. *Horticulturae*, *9*(8), 889. https://doi.org/10.3390/horticulturae9080889

Rahimi, W. N. S., Asraf, H. M., & Ali, M. S. A. M. (2020). Ananas Comosus Crown Image Thresholding and Crop Counting Using a Colour Space Transformation Scheme. *Telkomnika (Telecommunication Computing Electronics and Control)*, *18*(5), 2472. https://doi.org/10.12928/telkomnika.v18i5.13895

Raja, R., Nguyen, T. T., Slaughter, D. C., & Fennimore, S. A. (2020). Real-time weed-crop classification and localisation technique for robotic weed control in lettuce. *Biosystems Engineering*, *192*, 257–274. https://doi.org/10.1016/j.biosystemseng.2020.02.002

Raouhi, E. M., Lachgar, M., Hrimech, H., & Kartit, A. (2023). Unmanned Aerial Vehicle-based Applications in Smart Farming: A Systematic Review. *International Journal of Advanced Computer Science and Applications*, *14*(6), 1150–1165. https://doi.org/10.14569/IJACSA.2023.01406123

Ren, R., Sun, H., Zhang, S., Wang, N., Lu, X., Jing, J., Xin, M., & Cui, T. (2023). Intelligent Detection of Lightweight "Yuluxiang" Pear in Non-Structural Environment Based on YOLO-GEW. *Agronomy*, *13*(9), 1–16.

https://doi.org/10.3390/agronomy13092418

Sakuma, A., & Yamano, H. (2020). Satellite constellation reveals crop growth patterns and improves mapping accuracy of cropping practices for subtropical small-scale fields in Japan. *Remote Sensing*, *12*(15), 2419.

Schwartz, B. Y. R. O. Y., Dodge, J., Smith, N. A., & Etzioni, O. (2017). *Green AI*.

Sharma, S. (2023). Precision Agriculture: Reviewing the Advancements, Technologies, and Applications in Precision Agriculture for Improved Crop Productivity and Resource Management. *Reviews In Food And Agriculture*, *4*(2), 45–49. https://doi.org/10.26480/rfna.02.2023.45.49

Shen, Y., Liu, D., Chen, J., Wang, Z., Wang, Z., & Zhang, Q. (2023). On-Board Multi-Class Geospatial Object Detection Based on Convolutional Neural Network for High Resolution Remote Sensing Images. *Remote Sensing*, *15*(16). https://doi.org/10.3390/rs15163963

Shiu, Y. S., Lee, R. Y., & Chang, Y. C. (2023). Pineapples' Detection and Segmentation Based on Faster and Mask R-CNN in UAV Imagery. *Remote Sensing*, *15*(3). https://doi.org/10.3390/rs15030814

Stewart, E. L., Wiesner-Hanks, T., Kaczmar, N., DeChant, C., Wu, H., Lipson, H., Nelson, R. J., & Gore, M. A. (2019). Quantitative Phenotyping of Northern Leaf Blight in UAV Images Using Deep Learning. *Remote Sensing*, *11*(19), 1–10. https://doi.org/10.3390/rs11192209

Strubell, E., Ganesh, A., & McCallum, A. (2020). Energy and policy considerations for modern deep learning research. *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, *1*, 1393–13696. https://doi.org/10.1609/aaai.v34i09.7123

Strubell, E., & Mccallum, A. (2019). *Energy and Policy Considerations for Deep Learning in NLP*. *1*, 3645–3650.

Talaviya, T., Shah, D., Patel, N., Yagnik, H., & Shah, M. (2020). Jo ur of. *Artificial Intelligence in Agriculture*. https://doi.org/10.1016/j.aiia.2020.04.002

Tambe, U. Y., Shanthini, A., & Hsiung, P. A. (2024). Integrated Leaf Disease Recognition Across Diverse Crops through Transfer Learning. *Procedia Computer Science*, *233*, 22–34. https://doi.org/10.1016/j.procs.2024.03.192

Tang, Y., Chen, M., Wang, C., Luo, L., Li, J., Lian, G., Zou, X., & Neuwald, D. A. (2020). *Recognition and Localization Methods for Vision-Based Fruit Picking Robots : A Review*. *11*(May), 1–17. https://doi.org/10.3389/fpls.2020.00510

Tarkan Abdullah, S., Talib AL-Nuaimi, B., & Noman Abed, H. (2022). A survey of

deep learning-based object detection: Application and open issues. *Int. J. Nonlinear Anal. Appl*, *13*(March), 2008–6822. http://dx.doi.org/10.22075/ijnaa.2022.6525

Terven, J., Córdova-Esparza, D. M., & Romero-González, J. A. (2023). A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction*, *5*(4), 1680–1716. https://doi.org/10.3390/make5040083

Tong, L., & Wan, Y. (2023). A Hybrid Intelligence Approach for Circulating Tumor Cell Enumeration in Digital Pathology by Using CNN and Weak Annotations. *IEEE Access*, *11*(December), 142992–143003. https://doi.org/10.1109/ACCESS.2023.3343701

Vats, S., Singh Khanna, M., Kukreja, V., & Mehta, S. (2024). Federated Learning CNN Empowers Precision In Agriculture: A Case Study in Soybean Leaf Disease. *2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation, IATMSI 2024*, *2*, 1–6. https://doi.org/10.1109/IATMSI60426.2024.10502984

Vu, T. T. H., Vo, T. H., Tran, L. H., Choi, J., Vo, T. T., Ly, C. D., Nguyen, T. P., Mondal, S., & Oh, J. (2024). Segmenting hydrogen-induced cracking defects in steel through scanning acoustic microscopy and deep neural networks. *Engineering Reports*, *May*, 1–18. https://doi.org/10.1002/eng2.12933

Wan, C., Xiong, X., Wen, B., Gao, S., Fang, D., Yang, C., & Xue, S. (2022). Crack detection for concrete bridges with imaged based deep learning. *Science Progress*, *105*(4), 1–21. https://doi.org/10.1177/00368504221128487

Wan Nurazwin Syazwani, R., Muhammad Asraf, H., Megat Syahirul Amin, M. A., & Nur Dalila, K. A. (2022). Automated image identification, detection and fruit counting of top-view pineapple crown using machine learning. *Alexandria Engineering Journal*, *61*(2), 1265–1276. https://doi.org/https://doi.org/10.1016/j.aej.2021.06.053

Wang, S., Xu, D., Liang, H., Bai, Y., Li, X., Zhou, J., Su, C., & Wei, W. (2025). *Advances in Deep Learning Applications for Plant Disease and Pest Detection : A Review*. 1–30.

Warin, K., Vicharueang, S., Jantana, P., Limprasert, W., Thanathornwong, B., & Suebnukarn, S. (2024). Deep Learning for Midfacial Fracture Detection in CT Images. *Studies in Health Technology and Informatics*, *310*, 1497–1498.

https://doi.org/10.3233/SHTI231262

Wu, J., Wu, C., Guo, H., Bai, T., He, Y., & Li, X. (2023). Research on Red Jujubes Recognition Based on a Convolutional Neural Network. *Applied Sciences (Switzerland)*, *13*(11). https://doi.org/10.3390/app13116381

Xiao, B., Nguyen, M., & Yan, W. Q. (2024). Fruit ripeness identification using YOLOv8 model. *Multimedia Tools and Applications*, *83*(9), 28039–28056. https://doi.org/10.1007/s11042-023-16570-9

Xiao, Y., Wang, X., Zhang, P., Meng, F., & Shao, F. (2020). Object detection based on faster r-cnn algorithm with skip pooling and fusion of contextual information. *Sensors (Switzerland)*, *20*(19), 1–20. https://doi.org/10.3390/s20195490

Yamashkina, E. O., Kovalenko, S. M., & Platonova, O. V. (2021). Development of repository of deep neural networks for the analysis of geospatial data. *IOP Conference Series: Materials Science and Engineering*, *1047*(1). https://doi.org/10.1088/1757-899X/1047/1/012124

Zhang, L., Ding, G., Li, C., & Li, D. (2023). DCF-Yolov8: An Improved Algorithm for Aggregating Low-Level Features to Detect Agricultural Pests and Diseases. *Agronomy*, *13*(8), 1–16. https://doi.org/10.3390/agronomy13082012

Zhang, X., Zhang, Y., Gao, T., Fang, Y., & Chen, T. (2023). A Novel SSD-Based Detection Algorithm Suitable for Small Object. *IEICE Transactions on Information and Systems*, *E106.D*(5), 625–634. https://doi.org/10.1587/transinf.2022DLP0037

Zhao, X., Wang, L., Zhang, Y., Han, X., Deveci, M., & Parmar, M. (2024). A review of convolutional neural networks in computer vision. In *Artificial Intelligence Review* (Vol. 57, Issue 4). Springer Netherlands. https://doi.org/10.1007/s10462-024-10721-6

Zhu, R., Wang, X., Yan, Z., Qiao, Y., Tian, H., Hu, Z., Zhang, Z., Li, Y., Zhao, H., Xin, D., & Chen, Q. (2022). Exploring Soybean Flower and Pod Variation Patterns During Reproductive Period Based on Fusion Deep Learning. *Frontiers in Plant Science*, *13*(July), 1–18. https://doi.org/10.3389/fpls.2022.922030

# APPENDICES

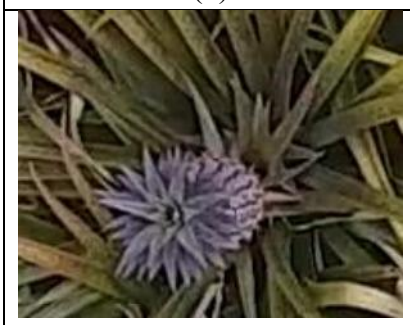
# APPENDIX 1

## Test Images Captured by a UAV at an Altitude of 10m Above the Ground Level

| | |
|---|---|
|  Image 1 |  Image 2 |
|  Image 3 |  Image 4 |
|  Image 5 |  Image 6 |
|  Image 7 |  Image 8 |
|  Image 9 |  Image 10 |

# APPENDIX 2

**Example of Train Images Sample (Variance in Appearance and Shape)**

| | | |
|---|---|---|
|  (1) |  (2) |  (3) |
|  (4) |  (5) |  (6) |
|  (7) |  (8) |  (9) |
|  (10) |  (11) |  (12) |

# APPENDIX 3

## Dataset Distribution for Training, Testing, and Validation

| Dataset Split | Altitude (m) | No. of images | No. of Annotated Pineapple Instances | Purpose |
|---|---|---|---|---|
| Training | 10 | 300 | 1 pineapple in each image | Model training |
| Testing | 10 | 1 | 269 | Performance testing |
| Testing | 8 | 1 | 234 | Performance testing |
| Testing | 6 | 1 | 92 | Performance testing |
| Testing | 4 | 1 | 25 | Performance testing |
| Validation | 9 | 1 | Not specified | Tile-size prediction |
| Validation | 7 | 1 | Not specified | Tile-size prediction |
| Validation | 5 | 1 | Not specified | Tile-size prediction |
| Validation | 3 | 1 | Not specified | Tile-size prediction |

# AUTHOR'S PROFILE



Nurhazirah binti Mohd Rahim received her Bachelor of Biomedical Electronic Engineering (Hons) from Universiti Malaysia Perlis in 2023. She is currently completing her MSc in Electrical Engineering at the College of Electrical Engineering, Universiti Teknologi Mara (UiTM) Cawangan Johor. Her MSc involves pineapple detection, combining a deep spatial slicing network with deep learning. She was a research assistant for RMS/FRGS funded project from 2024-2026.

## LIST OF PUBLICATION

Rahim, N. M., Hairuddin, M. A., Megat Ali, M. S. A., Tahir, N. M., Almisreb, A. A., & Ashar, N. D. K. (2025). Pretrained Convolutional Neural Network for Fruit Classification Analysis of Pineapple Plantation Images. *Engineering, Technology & Applied Science Research*, *15*(2), 20819–20826. https://doi.org/10.48084/etasr.9249