West Visayas State University
COLLEGE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY
La Paz, Iloilo City, Philippines

MangoSort: Mitigating Texture Bias in Lightweight Models for Automated Post-Harvest Grading of Carabao Mangoes via Attention-Based Knowledge Distillation

An Undergraduate Thesis

Presented to the Faculty of the

College of Information and Communications Technology

West Visayas State University

La Paz, Iloilo City

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Science in Computer Science

by

Dallas A. Aquino

Frederick Jibril L. Buñag

Ethan Jed V. Carbonell

Vincent L. Corpes, Jr.

October 2026

West Visayas State University
COLLEGE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY
La Paz, Iloilo City, Philippines

Approval Sheet

MangoSort: Mitigating Texture Bias in Lightweight Models for Automated Post-Harvest Grading of Carabao Mangoes via Attention-Based Knowledge Distillation

An Undergraduate Thesis for the Degree

Bachelor of Science in Computer Science

by

Dallas A. Aquino

Frederick Jibril L. Buñag

Ethan Jed V. Carbonell

Vincent L. Corpes, Jr.

_____	_____
Panel	Panel
_____	_____
Panel	Panel
_____	_____
Panel	Adviser

Concurred:

Dr. Ma. Beth S. Concepcion
College of ICT Dean

October 2026

Acknowledgment

The researchers would like to express their deepest appreciation to the following individuals and groups who have contributed in various ways to the completion of this study:

To Dr. Ma. Luche Sabayle, thesis adviser, for her invaluable academic supervision, continuous encouragement, and insightful technical guidance throughout the development of this study.

To the members of the defense panel, for their constructive criticism, rigorous evaluation, and expertise, which significantly contributed to refining the methodology and architecture of the MangoSort system.

To AMCen Western Visayas, for their valuable assistance in the development and fabrication of 3D-printed joints for the MangoSort device, contributing to the assembly, structural integration, and overall development of the system's hardware.

To the Carabao mango suppliers, for providing the physical fruit specimens essential to developing the morphological defect dataset.

To our friends and peers, for their camaraderie, assistance with dataset annotation and hardware assembly, and unwavering moral support throughout this journey.

Above all, the researchers offer their deepest gratitude to Almighty God for granting them the wisdom, strength, health, and perseverance needed to complete this study.

Dallas A. Aquino

Frederick Jibril L. Buñag

Ethan Jed V. Carbonell

Vincent L. Corpes, Jr.

October 2026

D. Aquino; F.J. Buñag; E.J. Carbonell; V. Corpes Jr., "MangoSort: Mitigating Texture Bias in Lightweight Models for Automated Post-Harvest Grading of Carabao Mangoes via Attention-Based Knowledge Distillation," Unpublished Undergraduate Thesis, Computer Science, College of Information and Communications Technology, West Visayas State University, Iloilo City, Philippines, 2026.

Abstract

The Philippine Carabao mango industry struggles with strict export standards and subjective manual grading. While edge AI offers scalable solutions, lightweight models such as YOLOv8-Nano were hypothesized to suffer from texture bias, misclassifying cosmetic blemishes as critical defects. This study introduces MangoSort, an automated grading framework deployed on a Raspberry Pi 5 that couples YOLOv8-Nano instance segmentation with a dual-parameter algorithm combining virtual weighing and defect surface area (in cm^2) to categorize fruit under the Philippine National Standards (PNS/BAFS 13:2024) [1]. An Attention-Based Knowledge Distillation (AT-KD) framework was developed to transfer spatial reasoning from a YOLOv8-Large teacher to the Nano student. On the held-out test split the teacher did not outperform the Nano baseline (mask mAP_{50-95} of 0.470 versus 0.498 and 0.482 for two baseline seeds), the AT-KD student (0.481) was 0.001 below the lower baseline seed and indistinguishable from the seed-to-seed spread, and the hypothesized

confusion of lenticel spots with critical defects did not occur in any model; the dominant error was missed detection of small cosmetic and lenticel regions. The baseline Nano model was selected for deployment. Virtual weighing from the segmentation masks estimated the weight of 15 fruit with a mean absolute error of 16.82 g (MAPE 5.83%). On a Raspberry Pi 5 the model ran at 4.1 frames per second (about 2.4 in the full control loop) for 60 minutes, with a peak CPU temperature of 65.0°C and no thermal throttling.

Keywords: Attention-Based Knowledge Distillation, Texture Bias, Instance Segmentation, Edge Computing, Post-Harvest Grading, Carabao Mango, YOLOv8

Table of Contents

Approval Sheet i

Acknowledgment ii

Abstract iv

List of Figures x

List of Tables xii

List of Appendices xiv

CHAPTER 1 INTRODUCTION TO THE STUDY 1

Background of the Study and Conceptual Framework 1

Objectives of the Study 3

Significance of the Study 5

Delimitation of the Study 10

Definition of Terms 13

CHAPTER 2 REVIEW OF RELATED STUDIES 18

Review of Existing and Related Studies 18

Current Systems 18

Related Systems or Solutions 23

Related Studies 26

CHAPTER 3 RESEARCH DESIGN AND METHODOLOGY 30

Description of the Study 30

Methods and Enhancements 31

Morphological Defect Dataset Construction 31
Attention-Based Knowledge Distillation (AT-KD) Pipeline 34
Edge Hardware Deployment and Benchmarking 36
Automated Grading Logic and Defect Severity Matrix . . 38
Components and Design 43
System Architecture 43
Software Architecture 44
Procedural Design 45
Object-Oriented Design 46
Methodology 47
System Development Life Cycle 47
Statistical Treatment and Evaluation Method 49
Evaluation of AI Vision 49
Robustness and Perturbation Evaluation 50
CHAPTER 4 RESULTS AND DISCUSSION 52
Implementation 52
Development Tools 52
Hardware Requirements 53
Hardware Enclosure and Fabrication 56
Software Requirements 60
Tkinter GUI Control Center 60
Localized Data Auditing via CSV 61

Input and Outputs of the Study 62
Implemented Output Interfaces 63
Results Interpretation and Analysis 65
Dataset Composition and Training Protocol 65
Model-Level Accuracy: Overall and Per-Class Performance 70
Effect of Attention-Based Knowledge Distillation . . . 75
False-Positive Behavior, Texture Bias, and Dominant Failure Modes 79
Robustness to Color, Texture, and Noise 83
Model Selection for Deployment 84
System Evaluation Results 86
Hardware-Level Efficiency on the Raspberry Pi 5 . . . 86
Calculated Capture-Window and Throughput Ceilings . . 88
Virtual Weighing Accuracy (Test A) 89
Limitations of the Study 92
CHAPTER 5 SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS 94
Summary of the Proposed Study, Design, and Implementation 94
Summary of Findings 96
Conclusions 99
Recommendations and Future Work 101
Engineering and Deployment Recommendations 101
Future Work 102
References 104

Appendices 114

List of Figures

1 MangoSort System Architecture 43

2 MangoSort Software Stack 44

3 MangoSort Grading Logic Module 45

4 MangoSort Sequence Diagram 46

5 MangoSort Hardware Components 56

6 MangoSort Enclosure CAD Model and Orthographic Dimensions (mm) 57

7 Custom 3D Printed Brackets 58

8 Actual Assembled MangoSort Prototype Based on the Custom CAD Design 59

9 MangoSort GUI - Camera & Controls Interface 63

10 MangoSort GUI - Live Sorting Logs Interface 64

11 Training and Validation Fitness Convergence Curves for the Baseline YOLOv8-Nano (Seed 0) Model 68

12 Training and Validation Fitness Convergence Curves for the Teacher YOLOv8-Large Model 69

13 Training and Validation Fitness Convergence Curves for the AT-KD Model 69

14 Class-Wise Accuracy Comparison Using the Diagonal Values of the Normalized Confusion Matrices 74

15 Qualitative Image Segmentation Predictions Comparing a Single Ground-Truth Annotation with the Baseline Nano, Teacher Large, and AT-KD Student Models 75

16 Validation Fitness Trajectory Highlighting Early-Epoch Baseline Variance and Transition Boundary 78

17 Diagnostic Visualization of Typical Lenticel Instance Detection Failures 82

18 Diagnostic Visualization of Cosmetic Blemish Detection Failures Correlated with Pixel Volume 82

19 Sample Visualizations of the Perturbed Held-Out Test Set used for Robustness Verification 84

20 CPU Load and Temperature During the 60-Minute Raspberry Pi 5 Benchmark (blue: CPU load; red: CPU temperature) 87

21 Correlation Analysis of Algorithmic Virtual Weighing vs. Physical Ground Truth Mass 91

22 Examples of Critical/Phytosanitary Defects 114

23 Examples of Cosmetic and Superficial Blemishes 114

24 Fully Assembled MangoSort Physical Prototype 115

25 Internal Darkroom Tunnel, Camera Mount, and Raspberry Pi 5 Wiring Assembly 115

List of Tables

I Mango Defect Annotation Dictionary 32

II Phase 1 - Minimum Requirements Verification 39

III Phase 2 - Size Classification for Carabao Mangoes (PNS/BAFS 13:2024) 41

IV Superficial Skin Defects 41

V Phase 3 - Mango Market Grade Classification 42

VI Texture Bias Failure Modes and Label Shifts 50

VII Robustness and Perturbation Evaluation Metrics . . . 51

VIII Development Tools: Hardware 52

IX Development Tools: Software 53

X Edge Deployment Hardware Requirements 54

XI Deployment Software Requirements 60

XII MangoSort System Inputs 62

XIII MangoSort System Outputs 62

XIV Morphological Defect Class Instance Distribution across Partitions 66

XV Experimental Training Protocol and Computational Convergence 67

XVI Model-Level Test Split Performance across Architectural Candidates 71

XVII Per-Class Mask Segmentation Accuracy (AP_{50-95}) on the Test Split 72

XVIII Per-Class Precision, Recall, and Optimal F_1-Score Evaluation on Test Split 73

XIX Matched-Epoch Validation Fitness Comparison (KD Student vs. Baseline Seed 0) 76

XX Threshold Sensitivity of Mask mAP_{50-95} on the Test Split 77

XXI Class-Confusion Rates (Fraction of True Class Predicted as Alternate Class at 0.25 Conf.) 79

XXII Absolute Miss Rates (True Instances Undetected at 0.25 Confidence) 80

XXIII Diagnostic of Validation Misses for Lenticel and Cosmetic Classes (Conf. 0.05, IoU 0.45) 81

XXIV Model Robustness against Perturbed Test Splits (mAP_{50-95}) 83

XXV Real-Time Hardware Inference and Resource Utilization on Raspberry Pi 5 (60-Minute Run) 87

XXVI Pixel-to-Metric Virtual Weighing Performance Evaluation 90

XXVII Raspberry Pi 5 Benchmark Environment 117

List of Appendices

Appendix A Morphological Defect Visual Dictionary 114

Appendix B Physical Prototype and Hardware Assembly 115

Appendix C Software Environment and Reproducibility 116

C.1 Training and Evaluation Environment 116

C.2 Edge Deployment Environment 117

Appendix D Disclaimer 118

CHAPTER 1 INTRODUCTION TO THE STUDY

Background of the Study and Conceptual Framework

The Philippine Carabao mango is a critical agricultural export, but the industry faces a severe market access crisis driven by stringent international Sanitary and Phytosanitary (SPS) standards [2]. High-value destination markets enforce zero-tolerance quarantine policies and rigid cosmetic requirements, resulting in packinghouse rejection rates between 30% and 50% [3]. Currently, compliance relies on manual grading, which is inherently subjective, deeply labor-intensive, and highly susceptible to psychological and visual fatigue, leading to significant inconsistencies across different batches, different packinghouses, and even different shifts within the same facility [4].

Automating this bottleneck via computer vision is hindered by deep morphological ambiguity, most notably the visual convergence between benign Lenticel Spotting and severe Fruit Fly damage [5]. To standard RGB sensors, both defects present identically as localized dark pixel clusters. Standard Convolutional Neural Networks (CNNs) designed for edge devices, such as YOLOv8-Nano, are reported to exhibit a texture bias, misclassifying these defects because they share identical textural profiles but ignore

global spatial distribution. Deep models like YOLOv8-Large were expected to overcome this via global shape-awareness, but they require high-end, power-hungry GPUs that are economically unviable for decentralized, rural packinghouses [6].

To bridge this accuracy-efficiency gap, this study proposes the MangoSort framework, deploying Attention-Based Knowledge Distillation (AT-KD) [7]. By applying attention transfer, with an optional Convolutional Block Attention Module (CBAM), during the training phase, MangoSort is designed to push the lightweight YOLOv8-Nano "Student" network to mathematically replicate the intermediate feature maps and shape-aware spatial reasoning of the massive YOLOv8-Large "Teacher" [8]. This pedagogical mechanism was hypothesized to counter the student network's texture bias, so that the resulting distilled model would reach the detection accuracy required for international compliance while retaining the minimal computational footprint necessary for real-time edge deployment on cost-effective hardware.

Objectives of the Study

The primary objective of this research is to engineer "MangoSort," an automated post-harvest grading system that leverages Attention-Based Knowledge Distillation (AT-KD) to equip a lightweight YOLOv8-Nano model with the capacity to accurately classify complex surface defect morphologies on a resource-constrained Raspberry Pi 5 edge device.

Specifically, it aims to:

1. Curate and annotate a diverse dataset of Carabao mango imagery using polygon-based instance segmentation. The dataset will specifically target surface defect morphology, classifying blemishes and grading the fruit in strict alignment with the Philippine National Standards on Mangoes (PNS/BAFS 13:2024) [1].
2. Develop and train a lightweight Student (YOLOv8-Nano-seg) through a custom knowledge distillation framework. This pipeline implements attention transfer on the neck feature maps, with a Convolutional Block Attention Module (CBAM) and feature-mimicry and logit-distillation terms available for ablation.
3. Formulate a dual-parameter grading mechanism that integrates

pixel-to-metric virtual weighing to estimate fruit volume and size codes, alongside surface area calculations to categorize mangoes according to PNS/BAFS 13:2024 standards [1].

4. Deploy the selected edge-optimal YOLOv8-Nano model onto a Raspberry Pi 5 edge environment and benchmark its viability across model-level segmentation accuracy, hardware runtime efficiency, and virtual weighing accuracy.

Significance of the Study

+

The development of **MangoSort** represents a critical intersection between advanced computer vision theory and the pragmatic needs of the Philippine agricultural sector. By investigating whether **Attention-Based Knowledge Distillation (AT-KD)** can address the texture bias attributed to lightweight neural networks, this research provides an empirical reference point for deploying accurate AI on low-cost hardware. The significance of this study is multifaceted, extending to the following stakeholders and domains:

1. **The Philippine Mango Industry and National Economy.** The Carabao mango is a cornerstone of Philippine agricultural identity and a primary export commodity. However, the industry is currently at a crossroads, constrained by documented rejection rates of 30–50% at export packinghouses [3]. This study is significant as it provides a direct technological intervention to stabilize the value chain. By automating the grading process, MangoSort aims to help ensure that only fruits meeting stringent International Phytosanitary Standards are shipped. This reduces the risk of interception at

foreign ports, thereby restoring the global reputation of Philippine exports and narrowing the yield gap between the Philippines and its regional competitors like Thailand [9].

2. **Export Packinghouses and Large-Scale Traders.** Currently, entry-level and localized packinghouse lines operate at throughputs of 60 to 100 fruits per minute, a volume at which human inspectors are inherently prone to error and inconsistency. This environment is highly susceptible to human error, fatigue-driven oversight, and subjective inconsistency. The significance of this research for exporters lies in throughput reliability. MangoSort introduces an objective and consistent grading mechanism that can distinguish between manageable pathological defects and strict quarantine barriers. For the exporter, this means fewer wasted logistics costs and a significant reduction in the financial losses associated with the rejection of entire shipping containers due to a single misidentified Fruit Fly infested mango.
3. **Small-Scale Farmers and Rural Decentralization.** High-end AI solutions often require server-grade GPUs, which centralizes the grading process in expensive, urban-based facilities. This study is significant because it democratizes access to precision agriculture. By optimizing a **YOLOv8-Nano**

model to run on a **Raspberry Pi 5**, this research shows that segmentation-based grading can be performed at the edge—in decentralized, rural packing stations closer to the farms. This empowers local farmer cooperatives to pre-sort their yields before they reach the major exporters, ensuring they receive a fair price based on an objective, AI-verified grade rather than a subjective middleman's assessment.

4. **Technical Advancement in Computer Vision and AI.** From a scholarly perspective, this study examines a limitation commonly attributed to shallow convolutional neural networks: **Texture Bias.** Lightweight models are often described as texture-greedy, failing to perceive distinct morphological characteristics that define specific biological defects. Moreover, this research contributes to the field of **Knowledge Distillation** by reporting an empirical test of spatial-attention distillation on a morphological task; on this dataset it did not outperform a non-distilled baseline, and its comparison with logit mimicry remains untested.
5. **Government and Regulatory Bodies (DA and DOST).** For agencies such as the Department of Agriculture (DA) and the Department of Science and Technology (DOST), this study serves as a successful use case for the Philippine Mango Industry

Roadmap 2021-2025 [10]. It provides a data-driven framework for modernizing post-harvest protocols. The Morphological Defect Dataset created during this study will serve as a high-quality annotated digital asset that can be used by other researchers to further the development of Philippine AgTech, fostering a culture of innovation and digital transformation in the local agricultural sector.

6. **Environmental Sustainability and Food Security.** Post-harvest loss is a significant contributor to food insecurity. When mangoes are rejected due to misidentification of defects, perfectly edible fruit often ends up in landfills or is left to rot, contributing to greenhouse gas emissions. MangoSort allows for a more granular sorting process where fruits with minor cosmetic Anthracnose can be redirected to local processing (for purees or dried mangoes) rather than being discarded along with Fruit Fly infested fruits. This contributes to a circular economy by maximizing the utility of every harvested fruit and reducing biological waste.
7. **Future Researchers.** Finally, this study serves as a foundational reference for future researchers exploring Attention-Based Knowledge Distillation (AT-KD) and Model Compression for resource-constrained edge devices. The methodologies es-

tablished here—specifically the attention-transfer distillation pipeline (with CBAM and 1×1 adapter components implemented for future ablation), as well as the three-tier benchmarking framework for Raspberry Pi deployment—can be adapted to other high-value crops (e.g., bananas, pineapples) or even different industries requiring precise, shape-aware morphological defect detection.

Delimitation of the Study

This study focuses on the development of "MangoSort," an automated post-harvest grading system utilizing Attention-Based Knowledge Distillation (AT-KD). To ensure the feasibility and precise execution of the research, the following delimitations have been established:

1. **Scope of the Specimen:** The system is strictly limited to the assessment of **Carabao mangoes** (*Mangifera indica L. cv. Carabao*). Other mango varieties (e.g., Pico, Indian, or Apple mangoes) exhibit different morphological baselines and defect manifestations and are therefore excluded from the dataset and model training. Furthermore, the system evaluates the fruit exclusively at the **post-harvest stage**; on-tree pre-harvest detection is beyond the scope of this study.
2. **Defect Detection Parameters:** The grading framework evaluates fruit quality based on external, surface-level morphological defects (such as Mango Scab, Latex Burn, Anthracnose) and 2D pixel-based size estimation. The study excludes the assessment of phenological maturity (ripeness stages based on colorimetric skin progression) and overall mango shape analysis. The physical MangoSort prototype does not incor-

porate mechanical load cells during real-time sorting. Instead, a calibrated digital scale is used exclusively during the benchmarking phase to independently validate the virtual weight estimation algorithm. The system estimates the weight-based size codes (Sizes 1 to 5) through virtual weighing using bounding polygon pixel-to-metric conversion. Furthermore, the detection of internal physiological disorders or pest infestations hidden within the flesh is beyond the scope of this study, as the system relies solely on standard optical RGB imaging rather than X-ray, load cells, or hyperspectral sensors.

3. **Grading Standards and Logic:** The automated classification logic is strictly bounded by the visual and physical criteria outlined in the Philippine National Standards for Mangoes (PNS/BAFS 13:2024) [1]. The system calculates the mango's specific Size Code alongside the cumulative cm^2 surface area of its defects to categorize the fruit into Extra Class, Class I, Class II, For Processing, or Reject tiers based on the standard's exact size-dependent threshold matrices. Destructive chemical evaluations are not included in the grading parameters.
4. **Hardware, Environmental, and System Constraints:** The de-

ployment environment is restricted to a Raspberry Pi 5 device to test the viability of the YOLOv8-Nano student model in resource-constrained settings. For cost-efficiency, the image acquisition setup utilizes standard RGB cameras rather than industrial global shutter cameras. Consequently, the physical prototype's conveyor speed is bounded to mitigate motion blur, and the study does not attempt to match high-end industrial sorting throughputs. Additionally, the system uses a localized CSV logging system and a Tkinter GUI Control Center. This ensures that the Raspberry Pi's memory and SD card I/O bandwidth are exclusively dedicated to real-time AI inference.

5. **Algorithmic Scope:** The machine learning pipeline is delimited to the **YOLOv8 instance segmentation architecture**. The training pipeline utilizes a **1×1 Convolutional Adapter** to align the high-dimensional feature representations of the teacher with the student's constrained architecture, enforced by a **Composite Loss Function**. Exploring other state-of-the-art architectures (such as Vision Transformers or Mask R-CNN) which are resource intensive for edge applications and falls outside the parameters of this research.

Definition of Terms

For better understanding, the following terms were defined conceptually and operationally:

Attention-Based Knowledge Distillation (AT-KD) -- A machine learning paradigm where a compact, lightweight neural network is trained to replicate the intermediate spatial feature maps and focal areas of a much larger, more complex neural network [11].

Attention-Based Knowledge Distillation (AT-KD) -- Operationally, this refers to the specific algorithmic pipeline used to transfer the ability to recognize nuanced surface defect textures from the heavy YOLOv8-Large-seg model to the lightweight YOLOv8-Nano model.

Carabao Mango -- A globally recognized strain of *Mangifera indica* cultivated in the Philippines, characterized by its distinct sweetness, exotic flavor profile, and specific susceptibility to tropical post-harvest diseases [12].

Carabao Mango -- Operationally, this refers to the sole agricultural specimen evaluated by the grading framework at the post-harvest stage to determine its class.

Composite Loss Function -- A mathematical algorithm utilized during the training of neural networks that combines multiple distinct

error metrics into a single optimizable objective, balancing different learning goals simultaneously [13].

Composite Loss Function -- Operationally, this refers to the customized formula ($L_{total} = L_{YOLO} + \alpha L_{AT-KD}$) that forces the student model to simultaneously learn ground-truth defect segmentation while actively mimicking the teacher model's spatial attention.

Convolutional Adapter (1×1) -- A specific neural network layer utilizing a 1×1 kernel size, primarily designed to project or reduce the channel dimensionality of a feature map without altering its spatial width and height [14].

Convolutional Adapter (1×1) -- Operationally, this refers to the mathematical bridge implemented to align the channel dimensions of the student and teacher feature maps during feature-level distillation.

Convolutional Block Attention Module (CBAM) -- A lightweight, general-purpose attention module for feed-forward convolutional neural networks that sequentially infers attention maps along separate channel and spatial dimensions [15].

Convolutional Block Attention Module (CBAM) -- Operationally, this refers to the dual-attention mechanism available in the AT-KD pipeline to help the student model suppress background conveyor noise (channel attention) and focus accurately on defect depth and

structural boundaries (spatial attention).

Edge Computing -- A distributed computing paradigm that brings computation and data storage closer to the location where it is needed, improving real-time response times and saving bandwidth [16].

Edge Computing -- Operationally, this refers to the localized deployment of the YOLOv8-Nano inference engine directly onto the Raspberry Pi 5 within rural packinghouses, eliminating the need for expensive, cloud-based GPU processing.

Instance Segmentation -- An advanced computer vision task that goes beyond bounding boxes by identifying and delineating individual objects of interest at the pixel level, generating precise, irregular polygon masks [17].

Instance Segmentation -- Operationally, this refers to the primary AI technique used to map the exact boundaries and calculate the cumulative surface area of irregular morphological defects on the mango skin.

Philippine National Standards (PNS/BAFS 13:2024) -- The official regulatory framework established by the Bureau of Agriculture and Fisheries Standards detailing the minimum quality requirements, sizing, and cosmetic defect tolerances for the commercial trade of mangoes [1].

Philippine National Standards (PNS/BAFS 13:2024) -- Operationally, this refers to the strict mathematical thresholds and defect severity matrices (Phase 1 and Phase 2) that the automated Python grading logic uses to assign the final fruit classification (e.g., Extra Class, Class I, Class II).

Student Model -- A highly compact neural network architecture featuring a reduced number of parameters, designed to achieve rapid inference speeds at the cost of some inherent representational capacity [18].

Student Model -- Operationally, this refers to the YOLOv8-Nano-seg model trained under the distillation framework to mimic the spatial attention of its larger counterpart; the non-distilled baseline of the same architecture was ultimately selected for deployment (see Chapter 4).

Surface Defect Morphology -- The biological and physical study of the form, structure, and topological characteristics of exterior abnormalities on a specimen [19].

Surface Defect Morphology -- Operationally, this refers to the distinct physical textures and visual patterns of mango blemishes (e.g., the corky, sunken cracks of Mango Scab versus the flat, streaky marks of Sap Burn) that the system is trained to differentiate.

Teacher Model -- A parameter-heavy, high-capacity neural network capable of extracting complex, high-dimensional features, typically used as a supervisory blueprint for training smaller networks [20].

Teacher Model -- Operationally, this refers to the YOLOv8-Large-seg model, which is trained extensively on the annotated Roboflow dataset to achieve maximum accuracy, serving as the source of attention maps during the distillation phase.

Texture Bias -- A well-documented phenomenon in computer vision where Convolutional Neural Networks (CNNs) disproportionately rely on local, superficial pixel patterns (such as color, grain, or localized contrast) rather than global geometric structures and structural boundaries to identify and classify objects [21].

Texture Bias -- Operationally, this refers to the inherent vulnerability of the lightweight YOLOv8-Nano model to misclassify distinct mango blemishes by focusing only on superficial color intensity, a limitation that the shape-aware Attention-Based Knowledge Distillation pipeline was hypothesized to mitigate.

CHAPTER 2 REVIEW OF RELATED STUDIES

Review of Existing and Related Studies

Current Systems

Philippine Mango Industry Roadmap (2021-2025). The Philippine mango industry, particularly the trade of the ”Carabao” variety, is currently experiencing a critical decline, with national production dropping and average yields falling to 4.1 metric tons per hectare — exactly half of Thailand’s 8.2 metric tons per hectare [10]. The Department of Agriculture’s roadmap emphasizes the urgent need for the strategic rehabilitation of unproductive trees and the implementation of improved post-harvest handling systems to reverse this declining productivity and enhance global export competitiveness [10]. Consequently, there is an urgent need for an accessible, decentralized technological intervention at the post-harvest grading stage to empower local packinghouses to systematically meet these strict international quality metrics.

Export Dynamics and Rejection Rates. The international trade landscape for Philippine Carabao mangoes is severely bottlenecked by a market access crisis, driven by stringent Sanitary and Phytosanitary (SPS) standards imposed by primary importing nations such as Japan and South Korea [2]. Due to systemic

issues, including the overuse of chemical pesticides leading to Maximum Residue Limit (MRL) violations and the presence of quarantine pests, shipments face devastating rejection rates [3]. At the exporter packing house level, inefficiency in manual sorting leads to significant rejection rates due to inconsistent manual grading [4]. Addressing this export bottleneck requires automating the precise detection of critical quarantine pests to prevent fundamentally flawed fruits from advancing into expensive export processing pipelines.

Philippine National Standards (PNS) for Mango Grading. To systematize domestic and international trade, the Philippine government established the PNS/BAFS 13:2024, which dictates the rigorous grading of Carabao mangoes into specific classifications—Extra Class, Class I, and Class II—based on tolerances for size, shape consistency, and the 2D surface area of visual defects [1]. Extra Class fruits must be of superior quality with practically no defects, whereas Class I and Class II permit minor skin blemishes, healed mechanical damage, or slight bruising within strict area-based limits. Specifically, the 2024 standard mandates that Class I defects must not exceed a 2D surface area of 2 cm^2 to 5 cm^2 depending on the fruit's size code, while Class II allows for a range between 4 cm^2 to 7 cm^2. Manual sorting frequently fails to ac-

curately quantify these precise geometric area tolerances, leading to misclassification and inconsistent grading. Bridging this gap necessitates a system that mathematically formalizes these PNS rubrics, allowing for rapid and objective categorization of mangoes into Extra Class, Class I, Class II, or Reject tiers with a precision that human inspectors cannot sustain over long shifts.

Minimum Quality Requirements and Size Classification. According to the PNS/BAFS 13:2024 standard, all mangoes regardless of class must meet minimum requirements: they must be whole, fresh in appearance, sound, clean, and practically free from pests or damage caused by pests [1]. Furthermore, the standard defines a strict weight-based size classification system utilizing size codes, ranging from Size Code 1 (above 350g) down to Size Code 5 (150-200g). In traditional packinghouses, sizing is often performed using mechanical weight sizers or visual estimation, both of which can be inconsistent. An automated system must integrate both 2D surface area analysis of defects and weight estimation to fully comply with the standard's grading criteria.

Tolerances and Uniformity Standards. The PNS/BAFS 13:2024 standard also specifies quality tolerances, allowing for a 5% margin of error for Extra Class and 10% for Classes I and II [1]. Additionally, uniformity is required within each package regard-

ing origin, variety, quality, and size. Current manual systems struggle to maintain this level of precision across thousands of units, often resulting in "over-grading" (rejecting good fruit) or "under-grading" (allowing defective fruit to pass), both of which undermine the credibility of Philippine mango exports in the global market.

Post-Harvest Loss Assessment in the Philippines. Studies have shown that significant post-harvest losses have been reported in the Philippine mango industry due to poor handling, inadequate grading facilities, and the rapid perishability of the fruit [22]. These losses represent a significant economic drain on small-scale farmers and packinghouses, highlighting the need for real-time, automated quality monitoring systems that can identify defects early in the value chain.

Impact of Hot Water Treatment (HWT) on Visual Grading. HWT is a standard quarantine process for Philippine mangoes to control fruit flies, but it can sometimes cause heat injury or "lenticel breakout," which complicates visual grading by introducing non-pathological skin blemishes [23]. Automated systems must be robust enough to distinguish between these non-critical heat-induced marks and pathological defects like Anthracnose to avoid unnecessary fruit rejection.

Morphological Typology of Mango Defects. The core difficulty in automating the mango grading process lies in the visual ambiguity of various surface defects prevalent in the Philippines. These quality-degrading factors are primarily categorized into Mechanical Damage, Biological Damage (Pathological and Entomological), and Physiological Damage. Mechanical damage frequently results from post-harvest mishandling, manifesting as latex sap contamination and severe physical bruising during transport [22]. Biological damage (pathological and entomological) encompasses fungal infections that cause necrotic, sunken tissue collapse [24], as well as insect infestations that induce hyperplastic, raised galls [25]. Physiological damage, such as internal tissue breakdown, occurs due to improper temperature control or storage conditions. Consequently, this creates severe classification challenges at industrial speeds. Standard computer vision systems often fail to differentiate the subtle topological variations of these defects when processing high volumes of fruit on conveyor lines. Therefore, advanced computer vision solutions must be specifically engineered to resolve the shape-from-shading cues necessary to distinguish these critical defect categories efficiently.

Manual vs. Automated Sorting Efficiency. The prevailing

methodology for grading Carabao mangoes relies heavily on manual inspection, a process that is inherently subjective, prone to worker fatigue, and logistically inefficient at the throughputs demanded by modern sorting lines [4]. In contrast, localized automated computer-vision grading systems have demonstrated high accuracy rates in processing mangoes while reducing labor requirements [26]. Similarly, the MANGGA system developed by UP Cebu achieved a 92% accuracy rate in size estimation compared to a manual sorter's 45% [27]. While these hardware systems provide the necessary mechanical throughput, there remains a critical gap for highly optimized, edge-deployable neural networks capable of nuanced defect differentiation to properly complement physical conveyor systems.

Related Systems or Solutions

Lightweight CNN Models for Agricultural Grading. The deployment of deep learning models in rural agricultural settings is severely constrained by hardware limitations, necessitating the use of lightweight architectures such as YOLOv8-Nano. While these models are highly efficient and suitable for edge deployment, they exhibit significant architectural limitations in complex quality grading. Lacking the parameter depth and hierarchical

complexity of large-scale models, standard lightweight Convolutional Neural Networks (CNNs) often fail to resolve fine-grained surface details [28]. Consequently, they tend to categorize distinct surface blemishes—such as latex burn and Anthracnose—merely as generic dark regions on the fruit surface.

Mobile-Based Grading Applications. Several studies have explored the use of smartphone-based applications for on-field fruit quality assessment, utilizing the device's built-in camera and processing power [29]. While these systems offer high accessibility, they often lack the controlled lighting and consistent positioning required for industrial-grade sorting, emphasizing the need for dedicated edge-computing stations in packinghouses.

Real-Time Maturity Detection Systems. Computer vision has also been applied to predict fruit maturity by analyzing skin color and texture patterns [30]. Integrating maturity detection with defect sorting provides a more comprehensive grading solution, allowing packing houses to optimize inventory management based on the projected shelf life of each fruit.

Automated Weight Estimation via Computer Vision. Since physical weighing is slow on high-speed lines, researchers have developed algorithms to estimate fruit weight by calculating volume from 2D images [31]. This "virtual weighing" is a critical

characteristic of modern automated systems, enabling them to meet the weight-based size requirements of the PNS standard without mechanical scales.

IoT-Enabled Sorting Lines. Integration of IoT sensors with computer vision allows for real-time monitoring of environmental conditions (temperature, humidity) during the sorting process, which correlates with the development of physiological defects [32]. These multi-modal systems provide a more holistic view of fruit quality but increase the computational burden on the local processing units.

Texture Bias in ImageNet-Trained CNNs. A fundamental technical hurdle in current computer vision systems is the phenomenon of texture bias. Standard CNNs, particularly those pre-trained on massive datasets like ImageNet, rely heavily on local texture statistics rather than global geometric shape [21]. In the specific context of Carabao mango grading, this bias was hypothesized to be particularly pronounced; because Fruit Fly damage, early stages of anthracnose, and benign lenticel spotting present similar high-frequency visual textures (dark blemishes on otherwise smooth skin), shallow networks were expected to have difficulty differentiating them. This challenge is further explained by Kawaguchi et al. [33], who quantified shape and texture biases,

emphasizing that enhancing shape awareness is critical for robust transfer learning in specialized domains like fruit grading.

Instance Segmentation for Exact Surface Area Calculation. The transition from bounding-box object detection to instance segmentation represents a significant advancement in the precision of agricultural grading systems. Unlike traditional detectors that provide only coarse spatial localizations, instance segmentation algorithms perform pixel-level classification, enabling the exact delineation of defect boundaries. This granularity is essential for calculating the precise 2D surface area of blemishes, such as latex burn or skin abrasions, which must be measured against the strict percentage-based tolerances defined by international quality standards. By isolating the exact pixels belonging to a defect, these systems can objectively determine if a fruit exceeds the allowed surface area threshold for specific grades, thereby eliminating the subjectivity inherent in manual visual estimation [17].

Related Studies

Attention-Based Knowledge Distillation (AT-KD). To bridge the gap between accuracy and edge efficiency, this study incorporates Attention-Based Knowledge Distillation. This technique en-

ables the transfer of spatial attention maps from a high-capacity teacher model to a lightweight student [11]. Recent advancements in 2025 have solidified this approach; Shi et al. [34] introduced the Positive Anchor Area Merge algorithm to optimize distillation for fruit detection, while the PKD-YOLOv8 framework [35] demonstrates that collaborative pruning and knowledge distillation are now the standard for achieving state-of-the-art agricultural object detection on constrained devices.

Convolutional Block Attention Module (CBAM). The use of attention modules like CBAM has been shown to significantly improve the performance of lightweight networks by emphasizing meaningful features and suppressing noise [15]. By integrating spatial and channel attention, models can better resolve the subtle topological differences between defects.

Transfer Learning for Agricultural Defect Detection. Utilizing pre-trained weights from large-scale datasets and fine-tuning them on specific agricultural datasets has been shown to drastically reduce training time and improve accuracy for small-scale datasets [36]. This technique is essential for projects like MangoSort, where the available pool of labeled Carabao mango images is relatively limited.

Model Pruning and Quantization for Edge AI. Beyond knowl-

edge distillation, techniques such as weight pruning and 8-bit quantization have been used to further optimize deep learning models for deployment on ARM-based processors like the Raspberry Pi **efficient_cnn_rpi2024.** These methods reduce the memory footprint and increase inference speed, allowing for more complex analysis within the same hardware constraints.

Knowledge Distillation for Model Compression. Hinton et al. pioneered the concept of Knowledge Distillation, which has since been adapted for compressing large-scale models into mobile-friendly architectures without significant loss in accuracy [37]. This foundational study provides the theoretical basis for transferring complex feature representations into the constrained environment of edge hardware.

Edge Computing for Precision Agriculture. Recent trends in agricultural technology emphasize the shift from cloud-based processing to edge computing to ensure low-latency responses [34]. Deploying models directly on hardware like the Raspberry Pi 5 reduces reliance on stable internet connectivity. This implementation is supported by Nahiduzzaman et al. [38], who achieved real-time classification on Raspberry Pi using YOLOv8, and Okano et al. [39], who validated YOLOv8-based visual conformity detection for industrial inspection on similar edge platforms.

Real-Time Fruit Grading Architectures. Comparative studies of various deep learning architectures have shown that while two-stage detectors offer high accuracy, single-stage detectors like YOLO are more suitable for the high-throughput requirements of industrial fruit grading [40]. While YOLOv8 remains a robust baseline, its application to mango defect detection has been specifically explored in recent literature. For instance, the development of a computer vision application for mango defect detection using YOLOv8 [41] provides the technical validation for using this architecture as the primary detector in the MangoSort system, particularly for addressing the export crisis through automated quality control.

Feature-Based Knowledge Distillation for Efficient Detectors. Recent advancements in feature-based knowledge distillation have demonstrated that transferring complex feature representations from a high-capacity teacher to a compact student significantly enhances the performance of lightweight object detectors [42]. This study provides the technical validation for using distillation techniques to improve the detection of subtle fine-grained surface defects in mangoes on edge-computing hardware like the Raspberry Pi 5.

CHAPTER 3 RESEARCH DESIGN AND METHODOLOGY

Description of the Study

MangoSort investigates the challenge of texture bias in computer vision models applied to the automated post-harvest grading of Philippine Carabao mangoes. As discussed in Chapter 1, the core technical challenge examined by this study is the texture bias attributed to lightweight CNN architectures. To resolve this within the computational constraints of edge hardware, MangoSort employs an Attention-Based Knowledge Distillation (AT-KD) framework structured around three primary methodological components: dataset construction, model distillation, and edge deployment.

The approach trains a high-capacity teacher model (YOLOv8-Large) and then distills its spatial attention maps into a lightweight student model (YOLOv8-Nano), with the aim of encouraging the student to rely on structural morphology rather than textural patterns; whether this aim was achieved is evaluated in Chapter 4. The system integrates a dual-parameter logic that combines this defect classification with 2D virtual weighing, resulting in a PNS/BAFS 13:2024-compliant grading mechanism on a Raspberry Pi 5 device.

Methods and Enhancements

Morphological Defect Dataset Construction

To achieve the first objective of the study, a diverse dataset of Carabao mango imagery was curated and annotated using polygon-based instance segmentation. Polygon masking allows the model to map exact boundaries of blemishes and the entire fruit structure. This granularity is crucial for calculating the cm^2 surface area of defects, enabling the "virtual weighing" of the mango via pixel-to-metric conversion to estimate its Size Code. The dataset is categorized based on tiers mandated by the Philippine National Standards on Mangoes (PNS/BAFS 13:2024) (see Table I and Appendix A for visual examples) [1].

TABLE I Mango Defect Annotation Dictionary

Annotation Label	Description	Notes
whole_mango	The entire fruit (skin area)	Represents the full mango surface.
defect_critical	All rot, deep cuts, and visible pest holes	Flesh-compromising & phytosanitary hazards.
defect_necrotic	Anthracnose, scab, and deep fungal lesions	Progressive or necrotic downgrades.
defect_cosmetic	Latex burns, abrasions, and old insect bites	Superficial marks that affect appearance only.
defect_lenticel	Rusty lenticel spots and mild nor thrips damage	Special tolerance exceptions.

To build this comprehensive dataset, raw imagery was sourced from two publicly available repositories. The first, the Mango Variety and Grading Dataset [43], provided baseline images categorized into Extra Class, Class I, and Class II tiers (200 images each, totaling 600). The second, MangoDHDS [44], contributed 1,500 images covering specific pathological conditions including stem end rot, anthracnose, scab, bacterial canker, and healthy baselines (300 images each). Because these source datasets were originally curated for image-level classification (folder-based sorting), they lacked the pixel-level spatial masks required for our study. Consequently, the researchers manually annotated all 2,100 raw images + 410 additional images of carabao mangoes the

researcher took, delineating the exact polygon boundaries for every defect instance to convert the classification datasets into a high-fidelity instance segmentation dataset.

The curated dataset comprises 2,510 unique photographs which, after 5× offline augmentation of the training partition only, yield **9,538 images** in total. To ensure robust model evaluation and prevent data leakage, the photographs were partitioned by source image into 70% training (1,757), 20% validation (502), and 10% test (251) sets before augmentation; augmenting the training set to 8,785 images gives the post-augmentation proportions of approximately 92%, 5%, and 3%. The class distribution reflects the natural occurrence of these morphological features in real-world post-harvest scenarios, yielding the following pre-augmentation annotation instance counts (12,957 in total): defect_cosmetic (7,510 instances), whole_mango (2,543 instances), defect_necrotic (1,652 instances), defect_lenticel (907 instances), and defect_critical (345 instances).

To enhance the generalization capability of the neural networks and mitigate overfitting within the constraints of edge deployment, a rigorous preprocessing and data augmentation pipeline was executed. All images were auto-oriented and stretched to a standardized 640×640 pixel resolution to match the YOLO architec-

ture's input requirements. To artificially expand the training variance, a multiplier of 5 outputs per training example was applied using the following spatial and pixel-level augmentations:

1. **Geometric Transformations:** Horizontal and vertical flips, random rotations between -14° and +14°, and horizontal/vertical shearing up to ±10°.
2. **Visual Degradation & Noise:** Gaussian blur (up to 1.6px), motion blur (simulating conveyor movement with a length of 20px, 90° angle, across 2 frames), pixel noise injection (up to 0.42% of pixels), and camera gain variance set to 0.03 to simulate lighting fluctuations in rural packinghouses.

Attention-Based Knowledge Distillation (AT-KD) Pipeline

Addressing the second objective, the MangoSort training code implements attention transfer alongside optional feature-mimicry, logit-distillation, CBAM, and adapter components to refine the student model's feature extraction. Only attention transfer was activated in the experiments reported in Chapter 4; the remaining components are described here as part of the framework design and were left for future ablation. To capture the hierarchical shape-awareness of the high-capacity Teacher (YOLOv8-Large-seg), forward hooks are attached specifically to the neck output layers

(P3, P4, P5, corresponding to YOLOv8 architecture indices 15, 18, and 21) which feed directly into the segmentation head.

Where feature-level mimicry is enabled, the pipeline can align the channel dimensions of the Student and Teacher feature maps using a 1×1 Convolutional Adapter followed by a 2D Batch Normalization (BatchNorm2d) layer (inactive in the reported runs). The distillation process is driven by a Composite Loss Function that linearly ramps up via an α multiplier over a 3-epoch warm-up schedule to stabilize early training. The composite loss can combine three terms:

1. **Spatial Attention Transfer (AT Loss):** Acts as a localized supervisory signal based on Zagoruyko & Komodakis' L2-normalized attention maps. By minimizing the squared L2 distance between these flattened spatial maps, the student learns to perceive the structural boundaries of a defect rather than just flat color.
2. **Feature Mimicry Loss:** Enforces feature mimicry on instance-normalized features. By applying instance normalization, the system removes per-channel scale and offset mismatches, forcing the student to replicate the teacher's relative feature distributions.
3. **Logit KD Loss:** Applies output-level distillation using

Soft-target Binary Cross Entropy (BCE). This is weighted by the Teacher's confidence scores to ensure that the massive number of background anchors do not drown out the critical defect signals.

To further suppress irrelevant background conveyor noise and amplify discriminative signals, the framework provides an optional CBAM block that applies sequential channel and spatial attention recalibration to the student's extracted features.

Edge Hardware Deployment and Benchmarking

To achieve the final objective, the selected YOLOv8-Nano model was deployed onto a decentralized Raspberry Pi 5 processing environment. To ensure the system's real-world viability for rural packinghouses, and to prevent compounding variables from skewing the final evaluation, the benchmarking methodology evaluated the system across three distinct performance tiers. Crucially, the system-level tier isolated the evaluation of the virtual weighing mechanism from the defect classification logic.

1. **Model-Level Accuracy:** The AI's pure visual performance was evaluated using Mean Average Precision (mAP_{50} and mAP_{50-95}) to measure polygon mask tightness and localization. Additionally, an ablation study utilizing a Confusion Matrix (measur-

ing Precision, Recall, and F1-Score) was conducted to specifically track changes in False Positive Rates. This quantified whether the AT-KD distilled model reduces texture-bias errors relative to the baseline YOLOv8-Nano.

2. **Hardware-Level Efficiency:** Running continuous AI inference on a Raspberry Pi requires strict resource monitoring. Because the physical prototype's conveyor speed is deliberately bounded, the system was benchmarked to determine its maximum viable throughput. Metrics included Real-Time Inference Speed (measured in Frames Per Second), Operational Latency (milliseconds per frame), and Sustained Thermal/Resource Utilization (CPU load and memory consumption) to ensure the hardware does not suffer from thermal throttling during continuous grading shifts.
3. **System-Level Grading Accuracy:** To ensure that potential errors in size estimation do not artificially distort the evaluation of the defect classification logic, the final automated grading mechanism is evaluated primarily through its sizing logic. The system's pixel-to-metric weight estimation algorithm was evaluated independently. The AI's estimated mathematical weight (g) for each mango was benchmarked against its absolute ground-truth weight, which was measured using a cal-

ibrated physical digital scale. The accuracy of the virtual weighing formulation was quantified using Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE).

Automated Grading Logic and Defect Severity Matrix

To strictly align with the updated Philippine National Standard for Mangoes (PNS/BAFS 13:2024) and ensure algorithmic consistency, MangoSort employs a deterministic, dual-parameter classification logic. The automated grading evaluates both the estimated size code of the mango and the cumulative surface area of detected blemishes [1]. The grading logic is illustrated in Fig. 3.

The system groups surface anomalies into broad morphological categories dictated by the standard and processes each fruit through the following sequential phases:

1. **Phase 1: Minimum Requirements Gating (Clause 4.1 Compliance)**

   In accordance with PNS/BAFS 13:2024 Clause 4.1, all mangoes, regardless of class, must meet baseline physiological and phytosanitary conditions. The system first scans for critical exclusionary defects. If any of the conditions in Table II are detected, the fruit violates minimum trade requirements and bypasses cosmetic area calculations entirely.

TABLE II Phase 1 - Minimum Requirements Verification

Tier	Annotation Label	PNS/BAFS 13:2024
Tier 1	defect_critical (Absolute Reject)	Rotting or Deterioration (Clause 4.1.d) Pest Damage Affecting Flesh (Clause 4.1.f)
Tier 2	defect_necrotic (For Processing)	Necrotic Stains or Trails (Clause 4.1.i) Marked Bruising (Clause 4.1.j)

2. **Phase 2: Classification via Virtual Weighing (Clause 6 Compliance)**

   If the fruit successfully passes Phase 1, the system performs "virtual weighing" to determine its physical size. As described in the delimitations, mechanical load cells are excluded from the real-time inference loop. Instead, MangoSort estimates fruit mass directly from 2D visual data through the following sequence:

   a. OpenCV algorithms extract the minimum bounding rectangle enclosing the whole_mango polygon to determine the major axis (L_{pixels}) and minor axis (W_{pixels}) in pixels.
   b. The pixel measurements are then converted into physical 2D lengths (cm) by applying a fixed camera calibration factor (K), representing the pixels-per-centimeter ra-

tio at the conveyor belt's specific focal depth.

$$MajorAxis(\text{cm}) = \frac{L_{\text{pixels}}}{K}$$

$$MinorAxis(\text{cm}) = \frac{W_{\text{pixels}}}{K}$$

c. Because a mango is fundamentally a 3D object, its volume (V) in cubic centimeters (cm^3) is mathematically modeled as an ellipsoid. Since the overhead camera only captures 2D axes, the minor axis is utilized as a proxy for the mango's physical depth.

$$\text{Volume}(\text{cm}^3) = \frac{4}{3}\pi \left(\frac{\text{Major Axis}}{2}\right)\left(\frac{\text{Minor Axis}}{2}\right)^2$$

d. The estimated volume is then multiplied by the standard specific gravity/density (ρ) of a mature Carabao mango (approximately 1.02 to 1.05 g/cm^3) to calculate the final estimated mass (M) in grams.

$$\text{EstimatedWeight}(\text{g}) = \text{Volume} \times \rho$$

This mathematically derived weight is then immediately categorized into the official Carabao Mango Size Codes as defined in PNS/BAFS 13:2024 (see Table III). By establishing the exact Size Code, the system dynamically determines the strict cm^2 tolerance limits required for the final Phase 3 cosmetic evaluation.

TABLE III Phase 2 - Size Classification for Carabao Mangoes (PNS/BAFS 13:2024)

Size Code	Weight (g)
1	> 350
2	> 300 - 350
3	> 250 - 300
4	> 200 - 250
5	150 - 200

3. **Phase 3: Classification via Aggregate Surface Area (Clause 5 Compliance)**

Once the Size Code is established, the system evaluates the total surface area of purely superficial skin defects (Table IV):

TABLE IV Superficial Skin Defects

Annotation Label	PNS/BAFS 13:2024
defect_lenticel	Scattered suberized rusty lenticels or sunburn (Clause 5.3)
defect_cosmetic	Rubbing, Sunburn, Suberized Stains due to Resin/Latex Exudation, Healed Bruises (Clause 5.2.b)

The 2024 standard dictates that the maximum allowable defect area (in cm^2) dynamically changes based on the size of the fruit. By converting the exact pixel area of segmented defects into cm^2, the system maintains strict alignment with the standard's intent. Based on these parameters, the system's logic outputs the final grade as shown in Table V:

TABLE V Phase 3 – Mango Market Grade Classification

Classification	Condition	Cosmetic Defect Tolerance
Extra Class	Passes Phase 1	Practically free of defects (Clause 5.1) All sizes: $\leq 1cm^2$ defect_cosmetic AND defect_lenticel = 0%
Class I	Passes Phase 1	Total area of defect_cosmetic must not exceed: Size 1: $\leq 5cm^2$ Size 2: $\leq 4cm^2$ Size 3: $\leq 3cm^2$ Size 4 & 5: $\leq 2cm^2$ AND defect_lenticel $\leq$ 40% of total area
Class II	Passes Phase 1	Total area of defect_cosmetic not exceed: Size 1: $\leq 7cm^2$ Size 2: $\leq 6cm^2$ Size 3: $\leq 5cm^2$ Size 4 & 5: $\leq 4cm^2$ AND defect_lenticel $\leq$ 40% of total area
For Processing	Passe Phase 1 but fails defect_cosmetic tolerances	defect_cosmetic exceeds the Class II cm^2 limits OR defect_lenticel $>$ 40% of total area
Reject	Fails Phase 1 (Tier 1)	defect_critical detected — Rotting or Pest Damage Affecting Flesh

Components and Design

System Architecture

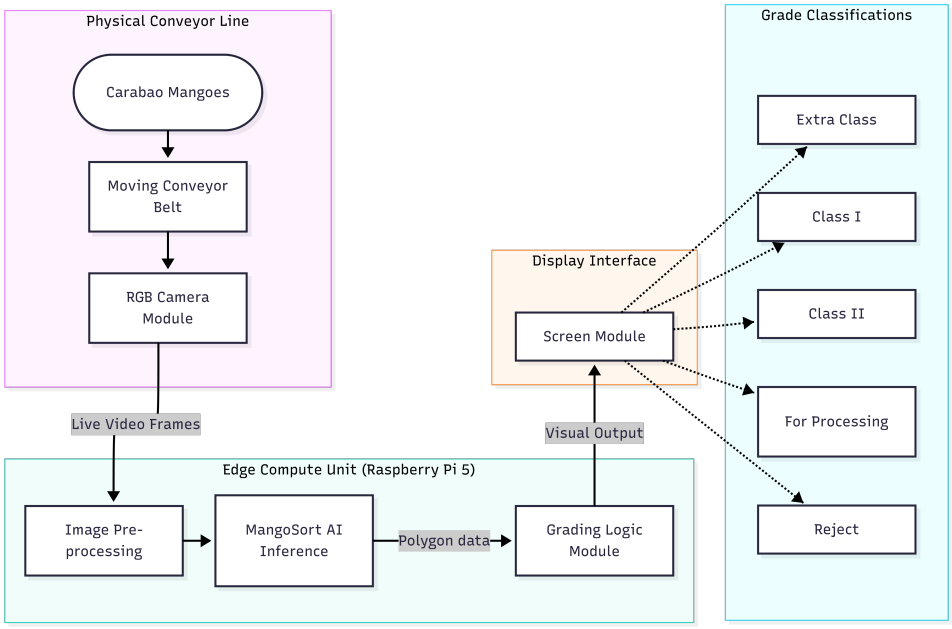


Fig. 1. MangoSort System Architecture

As shown in Fig. 1, the hardware system architecture consists of an RGB image acquisition module mounted over a sorting conveyor line. The camera feeds live video frames to the local Edge Compute Unit (Raspberry Pi 5). The Pi processes the frames locally without cloud reliance, executes the MangoSort AI inference, and outputs the final classification to a connected screen module to visually display the mango's grade (Extra Class, Class I, Class II, For Processing, Reject) for manual sorting or monitoring. The grading logic module is shown in Fig. 3.

Software Architecture

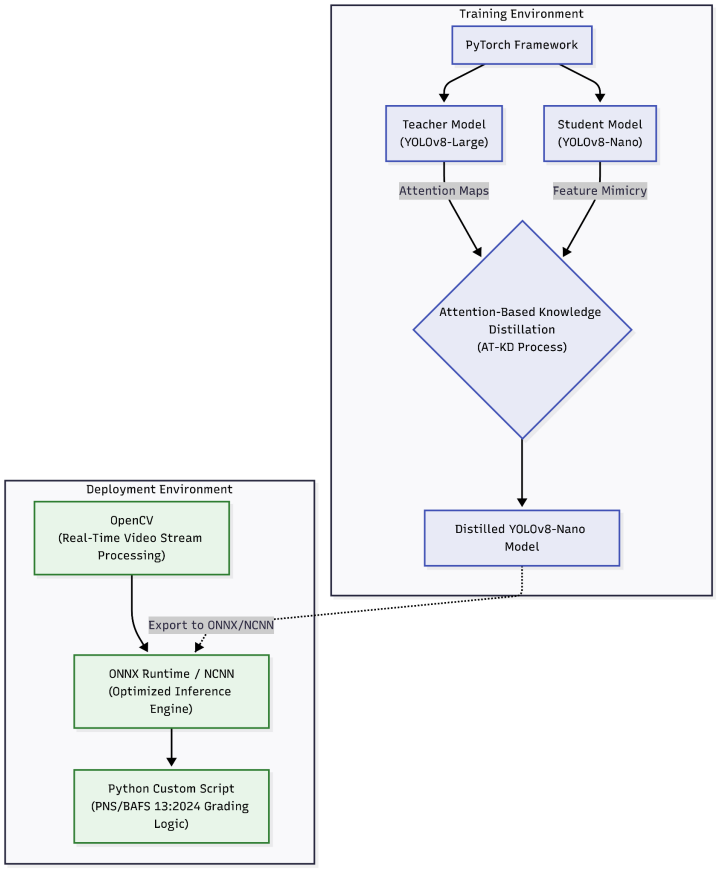


Fig. 2. MangoSort Software Stack

The software stack (Fig. 2) is divided into a Training Environment and a Deployment Environment. The Training Environment utilizes a high-performance GPU executing PyTorch to perform the AT-KD process between YOLOv8-Large and YOLOv8-Nano. The Deployment Environment on the Raspberry Pi utilizes an optimized inference engine (such as ONNX Runtime or NCNN) paired with OpenCV for real-time video stream processing and a Python-based custom script

executing the PNS/BAFS 13:2024 grading logic.

Procedural Design

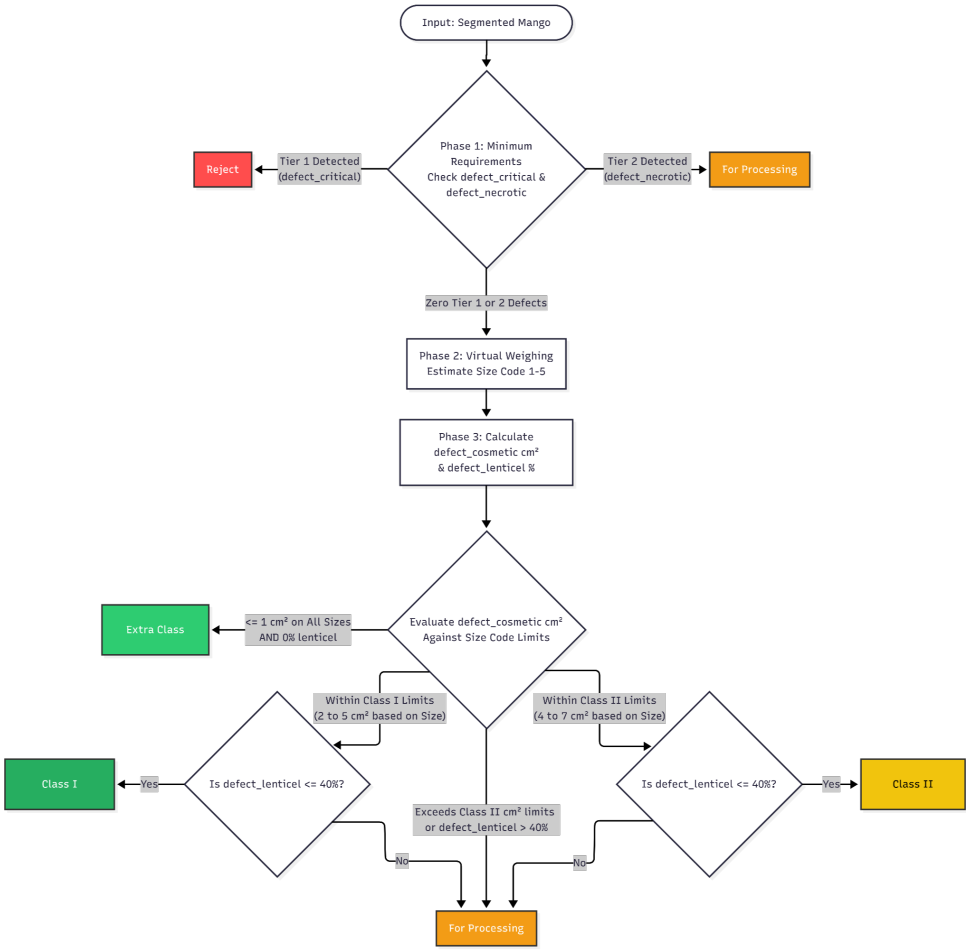


Fig. 3. MangoSort Grading Logic Module

The flowchart illustrates the sequential decision-making process of the MangoSort system. It maps the transition from Phase 1 minimum requirement gating, through Phase 2 virtual sizing, and culminates in Phase 3 size-dependent defect thresholds based on the official PNS/BAFS 13:2024 standards.

Object-Oriented Design

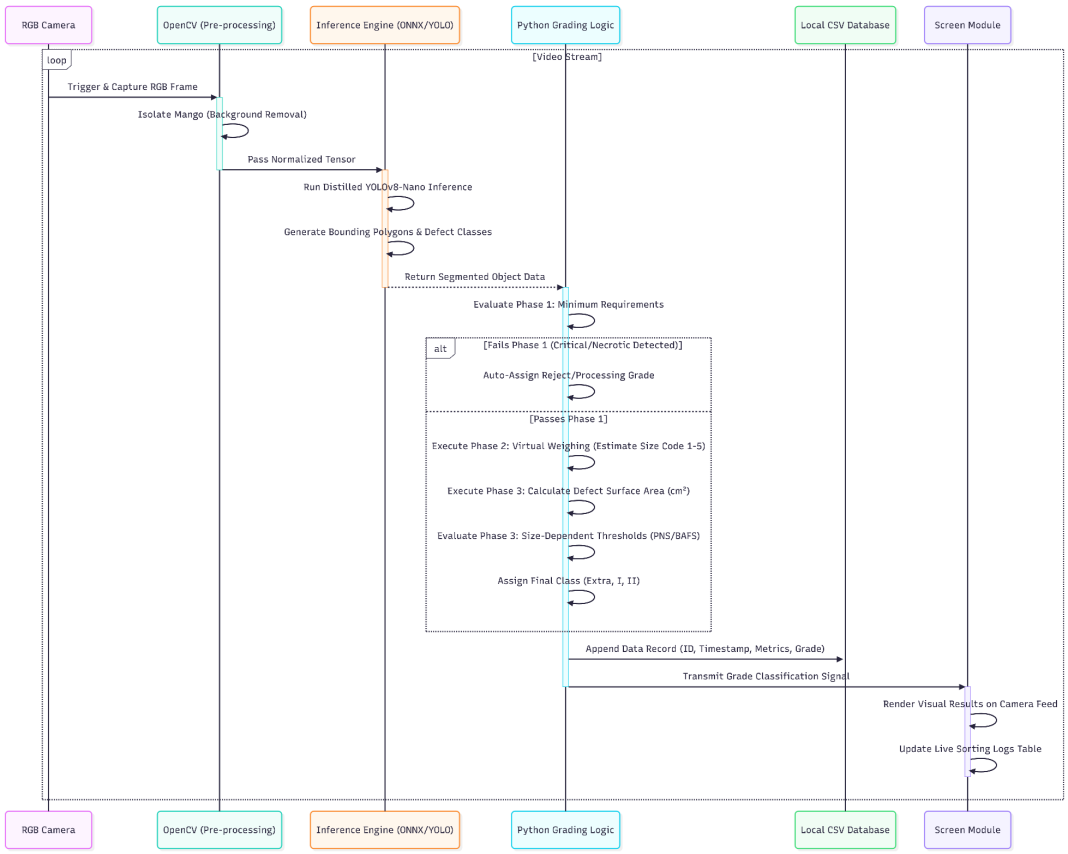


Fig. 4. MangoSort Sequence Diagram

The sequence diagram in Fig. 4 illustrates the chronological API and function calls across the hardware and software layers. It maps the exact sequence of events from the physical camera trigger, through the neural network memory allocation and bounding box generation, down to the final grade rendered on the Raspberry Pi's connected screen module.

Methodology

System Development Life Cycle

This study adopted an Agile Machine Learning framework, blending the principles of the Cross-Industry Standard Process for Data Mining (CRISP-DM) with iterative software development. This ensures continuous refinement of the AI model based on empirical testing. The SDLC phases are:

1. **Data Collection and Preparation:** Acquisition of raw Carabao mango images across varying grades and defect profiles. The data were labeled using polygon instance segmentation according to the established defect dictionary (see Table I Mango Defect Annotation Dictionary).
2. **Model Training and Distillation:** Training the YOLOv8-Large-seg (Teacher) to high convergence. Subsequently, the custom Composite Loss Function (attention transfer, with feature-mimicry, logit-distillation, CBAM, and 1×1 adapter terms available for ablation) was used to distill the spatial attention maps into the YOLOv8-Nano-seg (Student).
3. **Algorithmic Logic Translation:** Developing the Python-based mathematical modules to convert the model's pixel-level outputs into the three-phase PNS/BAFS 13:2024 weight- and

surface-area-based grading logic.

4. **Edge Deployment and Optimization:** Exporting the selected model to ONNX. The ONNX model and grading logic script were deployed onto the Raspberry Pi 5.
5. **Integration and System Testing:** Integrating the physical RGB camera, the Edge Compute Unit, and the external screen module. The end-to-end system is designed to operate on a conveyor line bounded to 50 mm/s to prevent motion blur.
6. **Evaluation and Iteration:** Executing the three-tiered bench-marking protocol. The system was evaluated for Model-Level Accuracy (mAP and Confusion Matrices to verify texture bias reduction), Hardware-Level Efficiency (maximum viable inference speed, thermal throttling), and System-Level Grading Accuracy (validating the virtual weighing mechanism).

Statistical Treatment and Evaluation Method

To validate the effectiveness of the MangoSort system, the following evaluation methods and statistical treatments were utilized across three distinct testing phases:

Evaluation of AI Vision

To evaluate the visual performance of the MangoSort instance segmentation model, the study utilizes a tiered benchmarking framework focused on model-level accuracy. The primary metric for assessing the precision of the generated polygon masks is Mean Average Precision, specifically mAP_{50} and mAP_{50-95}, which measure the tightness and localization accuracy of the segmented defect boundaries.

Furthermore, to evaluate the system's ability to mitigate texture bias, an ablation study is conducted using Confusion Matrices to track specific texture-bias failure modes defined by the study. The evaluation specifically monitors the False Positive/False Negative rates across highly ambiguous morphological pairs. By quantifying these specific misclassifications (detailed in Table VI), the study tests whether the AT-KD pipeline reduces them relative to the non-distilled baseline.

TABLE VI Texture Bias Failure Modes and Label Shifts

Misclassification Type	Technical Label Shift
Benign as Quarantine Hazard	defect_lenticel → defect_critical defect_cosmetic → defect_critical
Superficial as Progressive Decay	defect_cosmetic ↔ defect_necrotic

By quantifying these specific misclassifications, the study tests whether the AT-KD pipeline reduces them relative to the non-distilled baseline.

Robustness and Perturbation Evaluation

To test whether the distilled Student model learned global shape rather than overfitting to superficial local textures, a robustness evaluation is executed. The held-out test split is artificially perturbed into three distinct variants without altering the ground-truth polygon labels. As outlined in Table VII, a student model that successfully internalized the Teacher's shape-awareness would be expected to suffer less mAP degradation across these perturbed datasets compared to a texture-greedy baseline.

TABLE VII Robustness and Perturbation Evaluation Metrics

Perturbation Variant	**Applied Transformation**	**Evaluated Vulnerability**
Gray	Color data entirely removed (grayscale conversion)	Model reliance on chromaticity
Blur	Fine textures removed via Gaussian Blur ($\sigma = 3$)	Model reliance on high-frequency grain
Noise	Gaussian sensor noise injected ($\sigma = 12$)	Model stability under signal degradation

A student model that successfully internalized the Teacher's shape-awareness would be expected to suffer less mAP degradation across these perturbed datasets compared to a texture-greedy baseline.

CHAPTER 4 RESULTS AND DISCUSSION

Implementation

Development Tools

To ensure rapid prototyping and accurate ground-truth establishment during the development phase, a combination of localized physical tools and cloud-based machine learning environments was utilized (Tables VIII and IX).

TABLE VIII Development Tools: Hardware

Tool / Specification	**Purpose**
Local Workstation (Laptop)	Used for writing Python scripts, executing the Tkinter GUI, and running initial baseline OpenCV logic testing.
Google Colab (T4 Tensor Core GPU)	Cloud-based GPU compute environment utilized to bypass localized CPU bottlenecks, allowing for the rapid training of the YOLOv8-Nano baseline model across the 8,785-image augmented training set.

Google Colab was explicitly chosen for the training phase because training a segmentation model on a standard laptop CPU is highly inefficient and prone to memory exhaustion. By offloading the training process to a cloud-based T4 GPU, the researchers achieved rapid iteration cycles.

TABLE IX Development Tools: Software

Tool / Specification	**Purpose**
Roboflow Platform	Utilized for dataset curation, applying morphological polygon annotations, and generating dataset augmentations (e.g., shear, rotation).
Ultralytics YOLOv8 & OpenCV	The core computer vision libraries utilized for deep learning instance segmentation and mathematical contour rendering, respectively.
Visual Studio Code	The primary Integrated Development Environment (IDE) used for scripting the Python backend and Tkinter GUI.

Hardware Requirements

The system's physical architecture is designed for decentralized, rural packinghouses. Therefore, the hardware prioritizes cost-efficiency, low-power consumption, and localized edge-computing capabilities. Table X lists the minimum and recommended specifications.

TABLE X Edge Deployment Hardware Requirements

Component	Minimum Specification	Recommended Specification
Compute Unit	Raspberry Pi 4 Model B (4GB RAM)	Raspberry Pi 5 Single Board Computer (4GB RAM)
Vision Sensor	Standard 1080p USB Webcam	Raspberry Pi Camera Module V2 (8 Megapixels)
Interface	Standard external monitor via HDMI	Waveshare 7-inch LCD Raspberry Pi Touch Screen Display
Power Delivery	Standard 15W USB-C Adapter	Official 27W USB-C PD Power Supply for Raspberry Pi 5
Thermal Mgt.	Passive aluminum heatsinks	Official Raspberry Pi 5 Active Cooler
Connectivity	Standard 15-pin CSI ribbon cable	Official Camera CSI/DSI Flexible FPC Cable (15 to 22 Pin) - 500mm Length

The Raspberry Pi 5 (4GB) was selected as the optimal compute unit due to its upgraded processor, which is capable of running real-time YOLOv8-Nano inference without relying on cloud servers.

The Raspberry Pi Camera Module V2 was strictly chosen over newer autofocus modules because it features a fixed-focus lens. In the automated virtual weighing system, if a camera lens dynamically refocuses as a mango passes under it, it corrupts the fixed pixels-to-centimeter calibration scalar (K-Ratio). A fixed-focus lens guarantees mathematical consistency. Additionally, the 500mm FPC Cable was procured to allow the camera to be securely mounted

high above the conveyor belt while the compute unit remains safely housed at the control station.

A primary challenge encountered during hardware integration was thermal and power bottlenecking. Running continuous neural network inference maximizes CPU utilization, which can cause the Raspberry Pi to quickly reach 85°C and induce thermal throttling. This was mitigated by implementing the Active Cooler to sustain peak performance. Furthermore, under-voltage crashes were avoided by pairing the system exclusively with the Official 27W PD Power Supply. An architectural compatibility issue was also resolved regarding the camera connection; because the Pi 5 utilizes a newer high-density 22-pin MIPI port, a specialized 15-to-22 pin adapter cable had to be sourced to connect the legacy V2 Camera. The assembled components are shown in Fig. 5.

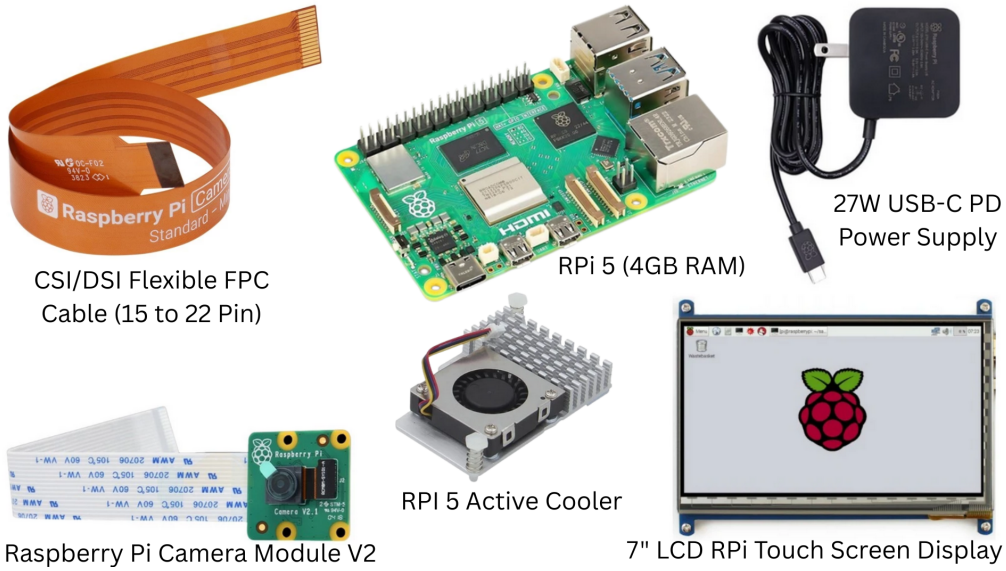


Fig. 5. MangoSort Hardware Components

Hardware Enclosure and Fabrication

To house the edge-computing components and ensure optical consistency for the vision sensor, a custom hardware enclosure was engineered. The structural design and assembly tolerances were modeled using Autodesk Fusion 360, with all dimensional parameters strictly drafted in millimeters (mm) to ensure a compact, desk-deployable footprint suitable for rural packinghouses.

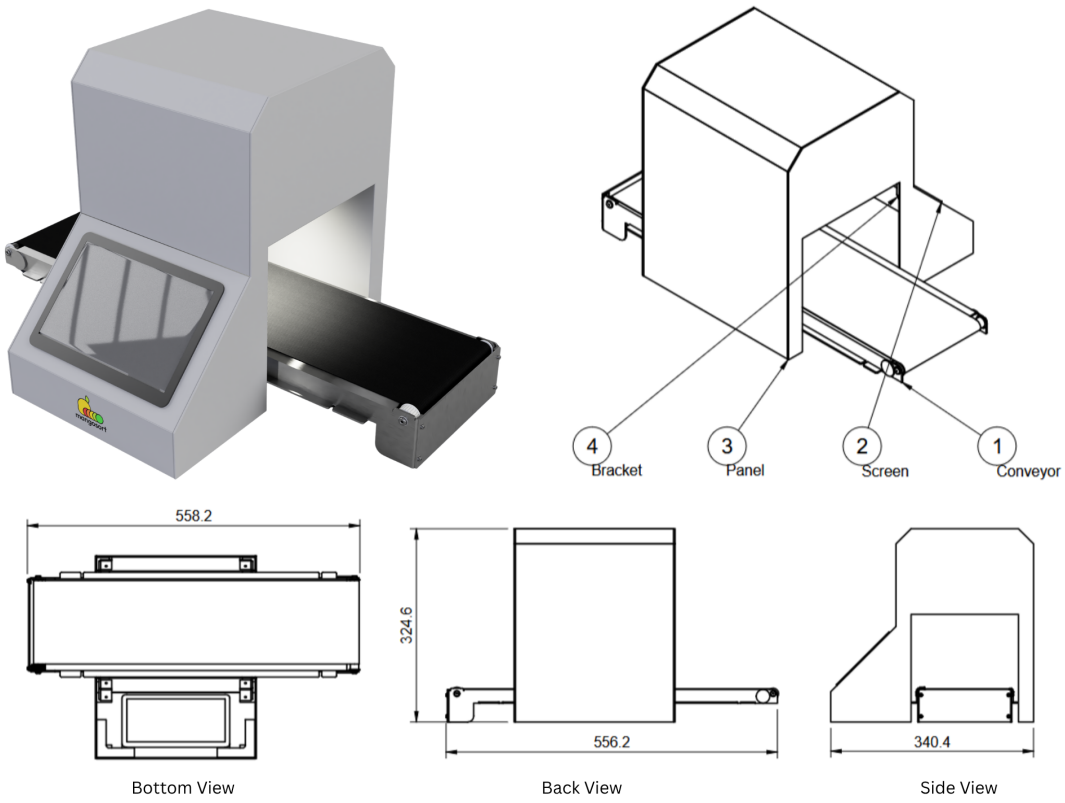


Fig. 6. MangoSort Enclosure CAD Model and Orthographic Dimensions (mm)

As illustrated in Fig. 6, the enclosure acts as a controlled darkroom tunnel over the conveyor belt. This mitigates external light pollution, which is critical for maintaining the accuracy of the YOLOv8-Nano segmentation masks. The primary body panels of the machine were fabricated using 3mm thick Sintra board (PVC foam board). This material was specifically selected because it is lightweight, structurally rigid, cost-effective, and easy to machine, making it highly reproducible for decentralized agricultural deployments.

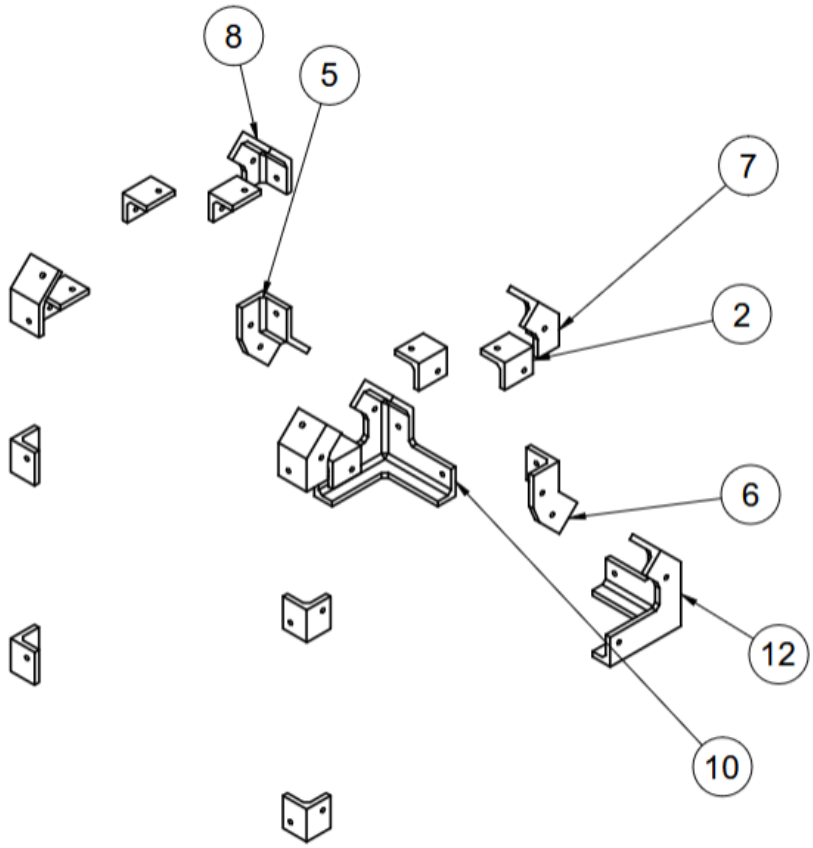


Fig. 7. Custom 3D Printed Brackets

To assemble the 3mm Sintra board panels without compromising the structural integrity of the viewing tunnel, a modular joinery system was developed. As shown in Fig. 7, a series of custom corner and edge brackets were designed in Fusion 360 and manufactured via 3D printing. These brackets allow the panels to be securely bolted together, creating a highly rigid chassis. This rigidity is paramount; any physical vibration from the conveyor motor transferring to the camera mount would alter the fixed focal length, thereby corrupting the pixel-to-metric conversion ratio (K-Ratio)

required for accurate virtual weighing. The assembled prototype is shown in Fig. 8.

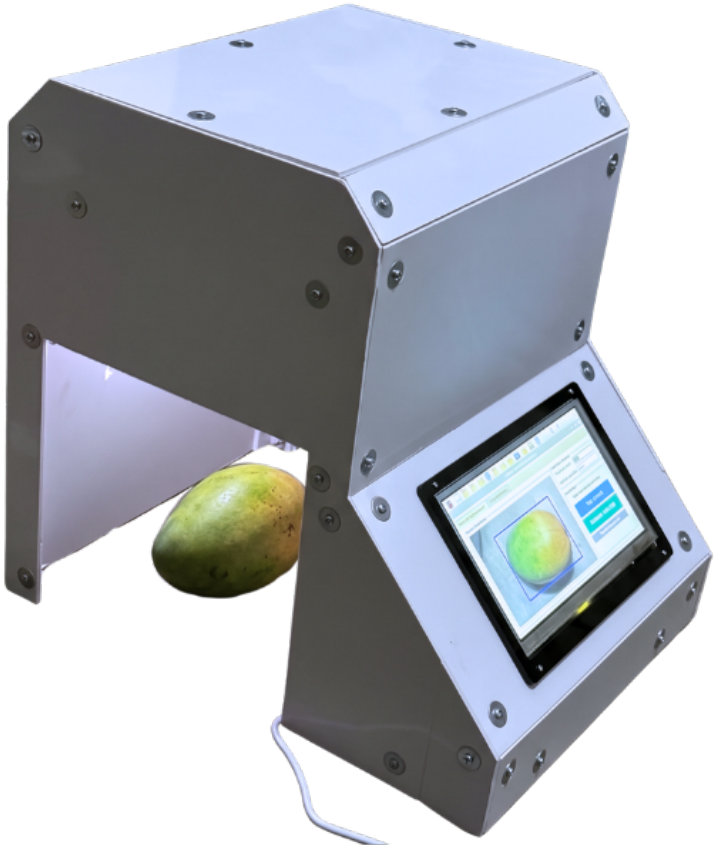


Fig. 8. Actual Assembled MangoSort Prototype Based on the Custom CAD Design

Software Requirements

To ensure maximum efficiency on the constrained edge device, the deployment software stack avoids heavy dependencies and web-browser interfaces, utilizing lightweight native libraries (Table XI).

TABLE XI Deployment Software Requirements

Component	Technology	Purpose
Operating System	Raspberry Pi OS	64-bit, Debian Bookworm with Desktop Environment for the base system
Python Environment	Python 3.13 & venv	Executes code within a strictly isolated Virtual Environment
Model Optimization Engine	Open Neural Network Exchange (ONNX)	To maximize inference speed and throughput
Vision & Processing	opencv-python	Handles real-time frame capturing and morphological contour mathematics
	numpy	Performs rapid array operations
User Interface & Logging	tkinter	Renders the localized Control Center GUI natively on the 7-inch touch screen
	pillow	Bridges OpenCV frames to the Tkinter canvas
	standard csv library	Generates automated, low-overhead post-harvest data logs

Tkinter GUI Control Center

To facilitate real-time system calibration and operational monitoring, the framework incorporates a dedicated Tkinter-based Graphical User Interface (GUI) Control Center. This centralized

dashboard allows operators to dynamically adjust critical system constants, specifically the K-Ratio (the pixel-to-metric conversion factor used for spatial calibration) and the Density coefficients (required for precise volumetric weight estimation). By providing a localized interface for parameter tuning, the system ensures high adaptability to varying hardware configurations—such as changes in camera mounting height or lens focal length—without requiring manual source code modifications or system restarts.

Localized Data Auditing via CSV

Instead of a resource-heavy relational database management system, the system leverages the Python CSV library for localized data auditing and grading logs. This lightweight approach ensures that the Raspberry Pi 5's I/O bandwidth and memory are prioritized for high-speed AI inference while maintaining a comprehensive, non-volatile record of all sorting activities. Each grading event—capturing the fruit's estimated weight, calculated defect surface area, and final classification under the PNS/BAFS 13:2024 standards—is appended to a timestamped CSV file. This provides a human-readable and easily exportable audit trail for yield analysis, allowing packinghouse managers to track throughput and quality distribution with minimal computational overhead.

Input and Outputs of the Study

The MangoSort system operates on a localized edge-computing pipeline that ingests raw optical data, processes it through the ONNX-optimized neural network, and outputs quantified agricultural metrics alongside a final classification based on the Philippine National Standards (PNS/BAFS 13:2024). Tables XII and XIII summarize the inputs and outputs.

TABLE XII MangoSort System Inputs

Input Category	Source & Parameters
Primary Optical Input	Real-time RGB video stream from the overhead Raspberry Pi Camera Module V2.
Operator Constants	**K-Ratio:** 33.0 Px/cm **Specific Gravity:** 1.02 g/cm^3
Operational Triggers	Manual capture commands or "Auto-Grade" targeting ROI.

TABLE XIII MangoSort System Outputs

Output Category	Rendered Data & Format
Intermediate Visual Output	YOLOv8-Nano instance segmentation masks, bounding boxes, and confidence scores projected directly over the live video feed.
Final Quantitative Output	Python grading logic outputs: Estimated Weight (g), Defect Surface Area (cm^2), Lenticel Percentage (%), and final BAFS Market Grade.
Data Logging	Dynamically rendered in the "Live Sorting Logs" interface with grade-specific color coding; simultaneously appended to a localized CSV database (mango_log.csv).

Implemented Output Interfaces

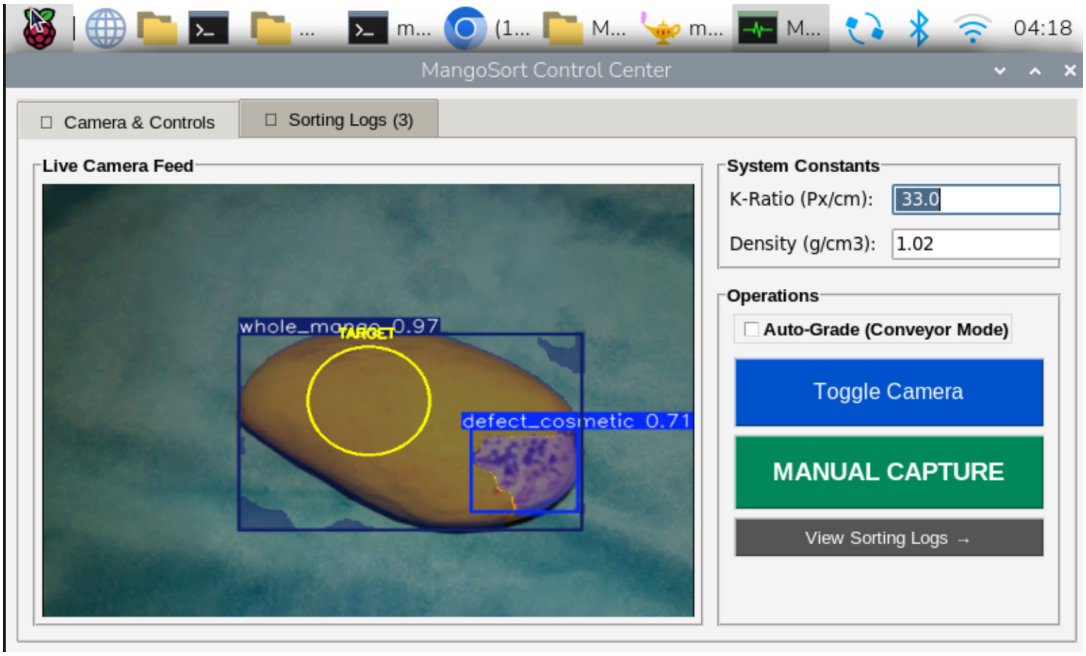


Fig. 9. MangoSort GUI - Camera & Controls Interface

Fig. 9 illustrates the primary input interface of the deployed prototype. The right panel displays the user-defined calibration constants (K-Ratio and Density) and operation toggles. The left panel demonstrates the intermediate visual output, showing the real-time YOLOv8-ONNX inference with instance segmentation masks successfully identifying the whole_mango and a defect_cosmetic anomaly. The yellow centralized target circle represents the automated debounce ROI.

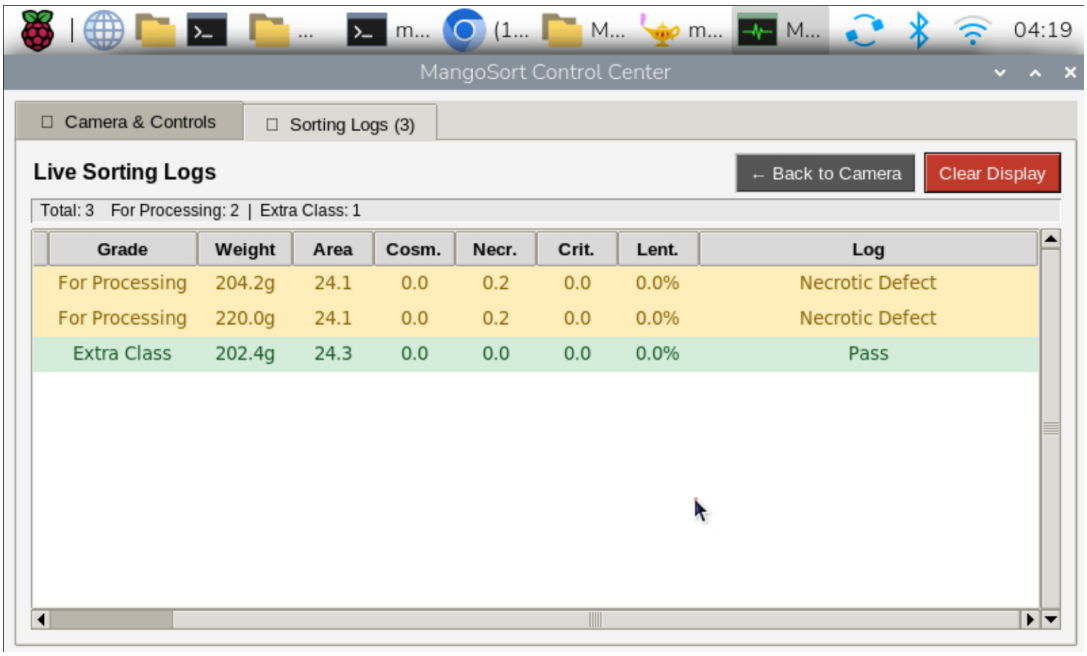


Fig. 10. MangoSort GUI - Live Sorting Logs Interface

Fig. 10 displays the final quantitative outputs of the system. The interface tabulates the mathematical results of the algorithmic pipeline, converting raw pixels into physical metrics (Weight and Surface Area). Based on these metrics, the system autonomously applies the PNS/BAFS 13:2024 tolerances to output the final deterministic classification, dynamically color-coding the results (e.g., green for Extra Class, yellow for For Processing) for efficient packinghouse management.

Results Interpretation and Analysis

Dataset Composition and Training Protocol

To rigorously evaluate the segmentation capability and texture-bias mitigation of the proposed framework, the morphological defect dataset was established under strict version control (Roboflow project mangosort-v2, version 6) adhering to the five primary morphological classes defined in Chapter 3: whole_mango, defect_critical, defect_necrotic, defect_cosmetic, and defect_lenticel.

The underlying image corpus comprises 2,510 unique photographic acquisitions of Carabao mangoes. To avoid leakage of augmented copies across splits, the unaugmented dataset was initially partitioned using a 70/20/10 split by photograph, so that no source photograph (or augmented copy of it) appears in more than one split, yielding 1,757 training images, 502 validation images, and 251 held-out test images. The 5× offline augmentation pipeline (spatial flips, rotations, shearing, Gaussian and motion blur, pixel noise, and simulated illuminance fluctuations) was strictly applied only to the training subset. This augmented the 1,757 baseline training images into 8,785 diverse training samples. When combined with the unaugmented validation (502) and test

(251) splits, the full experimental corpus comprises 9,538 images. This accounts for the post-augmentation distribution of approximately 92% training, 5% validation, and 3% testing cited in the system specifications.

TABLE XIV Morphological Defect Class Instance Distribution across Partitions

Morphological Class	**Validation (502 imgs)**	**Test (251 imgs)**	**Train (Augmented)**	**Pre-Augmentation Total**
whole_mango	503	250	8,819	2,543
defect_critical	62	31	1,260	345
defect_necrotic	294	170	6,017	1,652
defect_cosmetic	1,404	764	26,961	7,510
defect_lenticel	191	73	3,243	907
Total Instances	**2,454**	**1,288**	**46,300**	**12,957**

The instance distribution presented in Table XIV reflects real-world agricultural conditions. Benign cosmetic blemishes (defect_cosmetic) dominate the instances (7,510 pre-augmentation annotations), whereas severe flesh-compromising hazards (defect_critical) occur with significantly lower frequency (345 instances).

All model training and knowledge distillation routines were executed in an isolated Google Colab compute environment utilizing an NVIDIA Tesla T4 GPU (16 GB VRAM) running Ultralytics YOLOv8 (v8.4.175) and PyTorch under CUDA acceleration. All models were

initialized with standard COCO pre-trained weights to facilitate transfer learning across basic geometric primitives, trained at an input resolution of 640×640 pixels with a batch size of 16. The training protocol and execution parameters across the experimental candidates are summarized in Table XV, and Figs. 11, 12 and 13 show the corresponding convergence curves.

TABLE XV Experimental Training Protocol and Computational Convergence

Model Architecture	**Weight Init.**	**Epochs Run**	**Best Epoch**	**Time / Epoch**	**Total Run Time**
Teacher YOLOv8-Large-seg	COCO	28 (pat. 15)	13	∼546 s	4.3 h
Baseline YOLOv8-Nano (Seed 0)	COCO	79 (pat. 20)	59	∼235 s	5.2 h
Baseline YOLOv8-Nano (Seed 1)	COCO	52 (pat. 20)	30	∼242 s	3.5 h
AT-KD Student (Seed 0)	COCO	57 (24+33)	37	351 s / 237 s	4.5 h

The distillation protocol utilized an Attention Transfer loss directed at the output feature pyramids of the neck module (specifically layers P3, P4, and P5 corresponding to architecture indices 15, 18, and 21), assigned a distillation loss weight of 10 and an

attention balancing factor $\alpha = 1.0$ governed by a 3-epoch linear warm-up. Throughout active distillation, the attention transfer loss accounted for approximately 9% to 15% of the total composite backpropagation signal. To establish a transparent, empirical baseline, the primary distillation run was conducted purely with neck attention transfer; the inline CBAM spatial-channel module and 1×1 feature-adapter loss terms remained inactive during this benchmark.

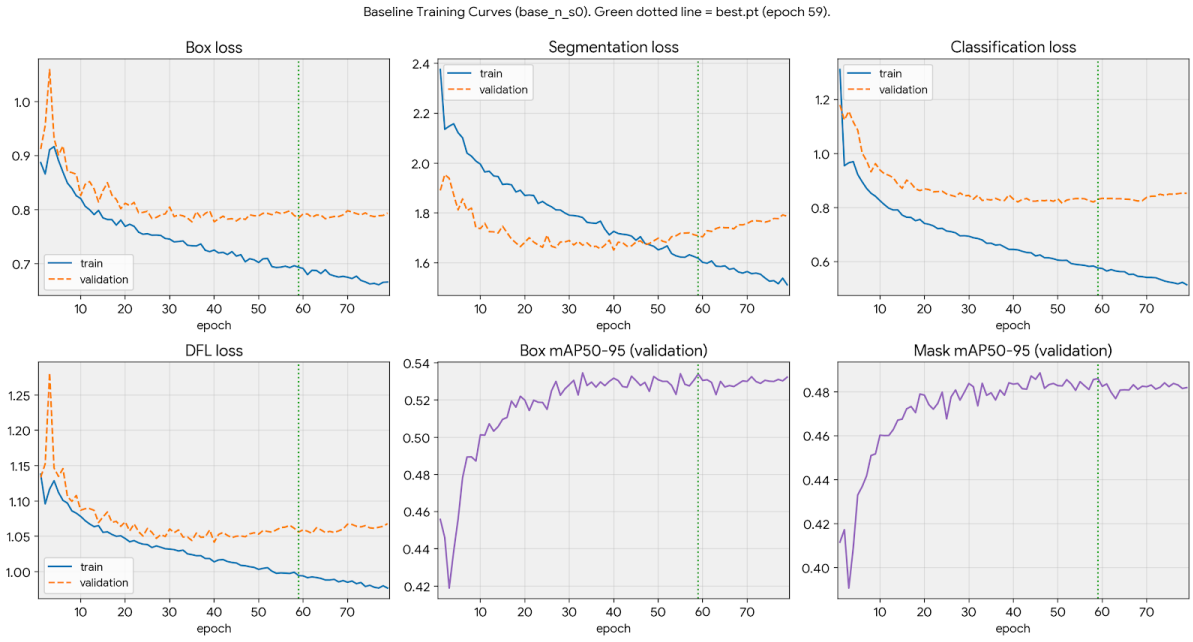


Fig. 11. Training and Validation Fitness Convergence Curves for the Baseline YOLOv8-Nano (Seed 0) Model

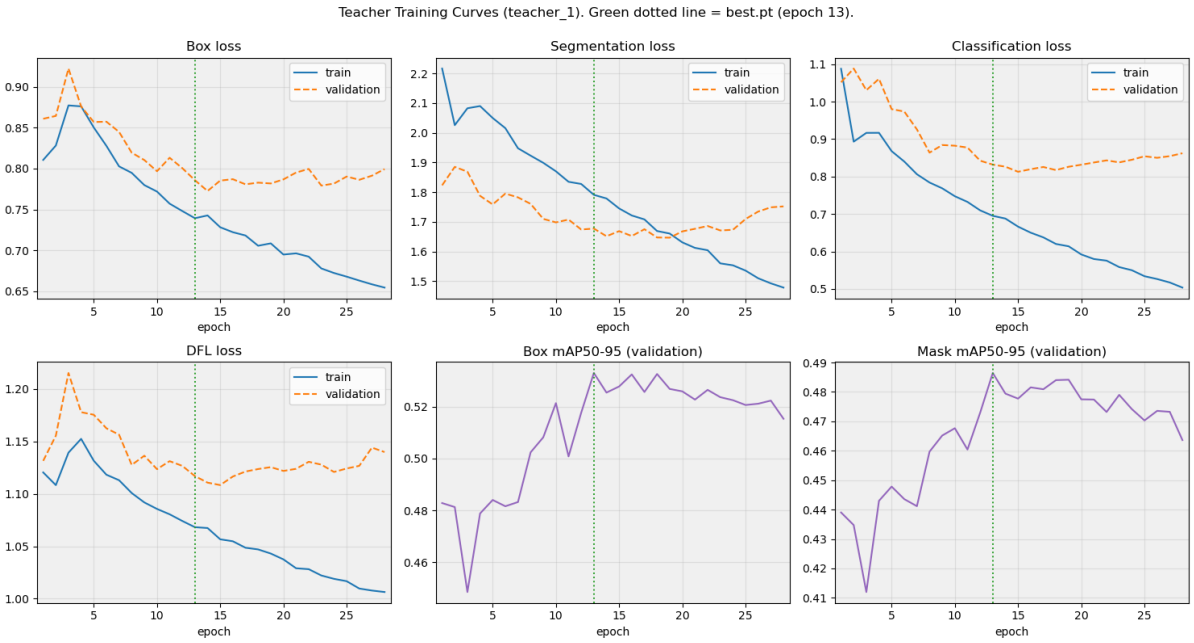


Fig. 12. Training and Validation Fitness Convergence Curves for the Teacher YOLOv8-Large Model

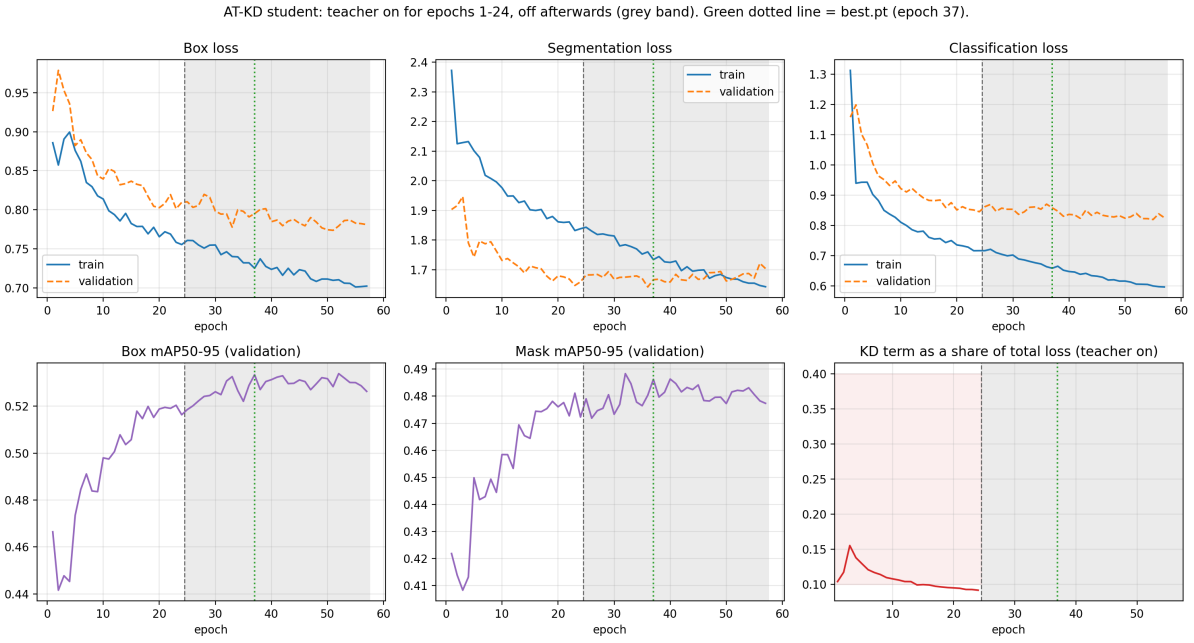


Fig. 13. Training and Validation Fitness Convergence Curves for the AT-KD Model

Model-Level Accuracy: Overall and Per-Class Performance

Because instance segmentation requires simultaneous spatial localization (bounding box coordinates) and pixel-level classification (contour mask delineation), model performance cannot be measured via standard scalar accuracy. In accordance with computer vision standards, visual performance is quantified using Mean Average Precision across intersection-over-union thresholds (mAP_{50} and mAP_{50-95}), alongside Precision, Recall, and the harmonic F_1-measure evaluated at the confidence score that maximizes F_1.

Table XVI outlines the comprehensive evaluation across the held-out test split (251 unaugmented images, 1,288 total ground-truth instances).

TABLE XVI Model-Level Test Split Performance across Architectural Candidates

Model Evaluation Candidate	**Parameters / GFLOPs**	**Box** mAP_{50-95}	**Mask** mAP_{50-95}	**Mask** mAP_{50}
Teacher YOLOv8-Large-seg	45.9 M / 210.1	0.513	0.470	0.642
Baseline YOLOv8-Nano (Seed 0)	3.4 M / 11.9	0.546	0.498	0.663
Baseline YOLOv8-Nano (Seed 1)	3.4 M / 11.9	0.537	0.482	0.650
AT-KD Student YOLOv8-Nano	3.4 M / 11.9	0.524	0.481	0.655

The empirical findings reveal that the compact YOLOv8-Nano baseline achieved parity with, and slightly exceeded, the multi-million parameter YOLOv8-Large teacher on this benchmark. Operating with only 3.4 million parameters and 11.9 GFLOPs (approximately $\frac{1}{13}$ of the parameter mass and $\frac{1}{17}$ of the computational complexity of the Large teacher), the baseline student achieved a Mask mAP_{50-95} of 0.498 (Seed 0) and 0.482 (Seed 1), whereas the YOLOv8-Large teacher achieved 0.470. On the validation split, both architectures converged to an identical Mask mAP_{50-95} of 0.484 and Box mAP_{50-95} of 0.533.

To determine how architectural capacity impacts morphological resolution, Table XVII delineates the granular per-class segmentation performance (AP_{50-95}) across all five targeted categories.

TABLE XVII Per-Class Mask Segmentation Accuracy (AP_{50-95}) on the Test Split

Target Class	Teacher (Large)	Baseline (Seed 0)	AT-KD Student	Baseline (Seed 1)
whole_mango	0.858	0.890	0.856	0.876
defect_critical	0.734	0.795	0.732	0.762
defect_necrotic	0.466	0.473	0.501	0.466
defect_cosmetic	0.241	0.257	0.249	0.253
defect_lenticel	0.048	0.075	0.068	0.051

A detailed analysis of Table XVII reveals consistent trends across all candidate architectures:

1. **Macroscopic and High-Contrast Defect Generalization:** Macro-structural segmentation is highly consistent. The delineation of the complete fruit boundary (whole_mango) achieved high precision (AP_{50-95} between 0.856 and 0.890). Similarly, phytosanitary hazards characterized by marked tissue degradation and deep contrast punctures (defect_critical) were segmented effectively across all models (AP_{50-95} between 0.732 and 0.795).
2. **Moderate Resolution of Pathological Necrosis:** Pathologi-

cal fungal lesions (defect_necrotic), including anthracnose and deep fungal scab, achieved moderate segmentation fidelity (AP_{50-95} ranging from 0.466 to 0.501). Notably, the AT-KD student model yielded its highest relative improvement within this category, achieving an AP_{50-95} of 0.501 compared to the teacher's 0.466 and the Seed 0 baseline's 0.473.

3. **Morphological Ambiguity in Fine Surface Blemishes:** Superficial cosmetic defects (defect_cosmetic) exhibited low segmentation performance across all models ($AP_{50-95} \approx 0.241 - 0.257$). Furthermore, minute lenticel spotting (defect_lenticel) posed a severe classification challenge, yielding low mask precision across all architectures ($AP_{50-95} < 0.08$). Crucially, this limitation was shared by the high-capacity YOLOv8-Large teacher ($AP_{50-95} = 0.048$), demonstrating that the challenge stems from feature representation and label variance rather than the capacity of the lightweight convolutional backbone.

TABLE XVIII Per-Class Precision, Recall, and Optimal F_1-Score Evaluation on Test Split

Target Class	**Precision**	**Recall**	**F_1-Score**
whole_mango	0.9807	0.9880	0.9843
defect_critical	0.7079	0.9355	0.8059
defect_necrotic	0.6097	0.7529	0.6738
defect_cosmetic	0.5807	0.4423	0.5022
defect_lenticel	0.3387	0.1825	0.2372
All Classes (Mean)	**0.6435**	**0.6602**	**0.6407**

Table XVIII reports per-class precision, recall, and F_1 on the test split. On the validation set, the teacher model achieved an aggregate mask precision of 0.716 and recall of 0.626 ($F_1 \approx 0.668$), while the Baseline Seed 0 model attained an aggregate precision of 0.696 and recall of 0.635 ($F_1 \approx 0.664$). These results suggest that, on this dataset, shallow edge networks can match the feature-extraction quality of large models when segmenting post-harvest agricultural defects under consistent imaging conditions. Fig. 14 compares class-wise accuracy from the normalized confusion matrices, and Fig. 15 gives a qualitative example.

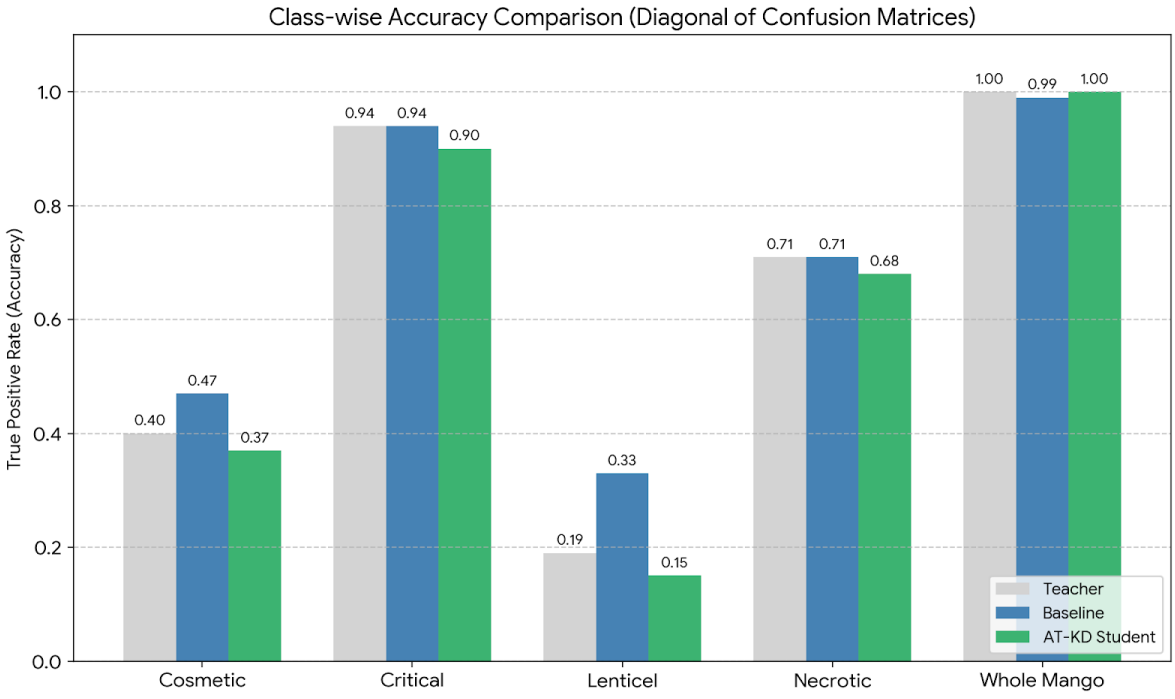


Fig. 14. Class-Wise Accuracy Comparison Using the Diagonal Values of the Normalized Confusion Matrices

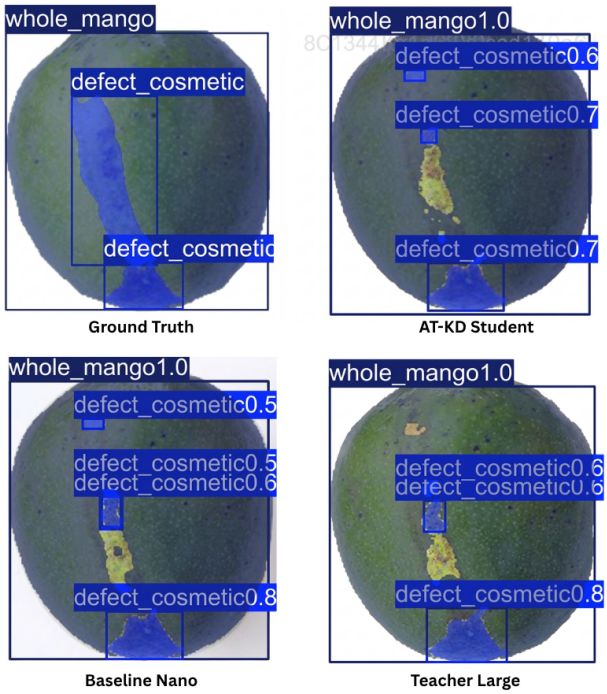


Fig. 15. Qualitative Image Segmentation Predictions Comparing a Single Ground-Truth Annotation with the Baseline Nano, Teacher Large, and AT-KD Student Models

Effect of Attention-Based Knowledge Distillation

A fundamental premise of this study was that the parameter-heavy YOLOv8-Large model possessed complex shape-aware feature representations that could be distilled into the lightweight YOLOv8-Nano student. However, empirical benchmarking revealed a critical capability gate: the teacher model did not exhibit supervisory superiority on this specific dataset.

As previously detailed in Table XVI, the teacher did not exceed the baseline's validation performance (both models converging at Mask $mAP_{50-95} \approx 0.484$) and scored lower on the test split (0.470

vs. 0.498 and 0.482). Despite costing $13.5\times$ the parameters, $17.7\times$ the computational FLOPs, and exhibiting an inference latency of roughly 60 ms per image on the T4 GPU (compared to the Nano's $\sim$7 ms), the YOLOv8-Large model did not harbor significantly superior morphological knowledge to impart to the student network.

To isolate the specific impact of the distillation mechanism from epoch-variance, a matched-epoch validation comparison was conducted using identical learning-rate schedules. As demonstrated in Table XIX, the performance delta between the AT-KD Student and the Baseline Seed 0 model remained indistinguishable from epoch-to-epoch training noise.

TABLE XIX Matched-Epoch Validation Fitness Comparison (KD Student vs. Baseline Seed 0)

Training Phase	**Mean Fitness Difference**	**Epochs KD Ahead**	**Best Fitness (KD vs. Baseline)**
Epochs 1–24 (Teacher Active)	+0.004	11 of 24	1.0016 vs. 1.0011
Epochs 25–57 (Teacher Inactive)	-0.001	14 of 33	1.0197 vs. 1.0201

The observed per-epoch noise within the fitness differential fluctuated between ± 0.008 to 0.015, rendering the performance of both phases virtually identical. Furthermore, when analyzing test set variation, the baseline seeds exhibited an internal spread of

0.016 in Mask mAP_{50-95}. The AT-KD student scored 0.481, which is 0.001 below Seed 1 and 0.017 below Seed 0, and is therefore indistinguishable from the un-distilled baselines given that spread.

An ablation of detection confidence thresholds (Table XX) indicates that the student model's performance gap narrows slightly at lower thresholds, suggesting that distillation marginally impacted bounding-box confidence calibration but did not fundamentally alter boundary accuracy.

TABLE XX Threshold Sensitivity of Mask mAP_{50-95} on the Test Split

Confidence Threshold	**Baseline (Seed 0)**	**Baseline (Seed 1)**	**AT-KD Student**
≈0.001 (Standard mAP)	0.498	0.482	0.481
0.10	0.487	0.472	0.469
0.25 (Analysis Threshold)	0.479	0.461	0.454

Ultimately, while the computational footprint at deployment remained identical for both Nano variants (3.4 M parameters, 11.9 GFLOPs, 13.8 MB ONNX execution file), the computational burden during the training phase increased substantially. A distillation epoch required an average of 351 seconds compared to the standard baseline's 235 seconds, amounting to a $1.5\times$ increase in training cost without yielding a proportionate increase in inference accuracy.

A visual inspection of the validation fitness trajectory

(Fig. 16) reveals that the AT-KD run maintained a mathematical lead strictly during the initial training phase (Epochs 1–6), peaking at a +0.046 differential at Epoch 3. However, an analysis of the absolute fitness curves (left panel) demonstrates that this early gap was driven entirely by a sharp performance dip in the baseline model (dropping to a fitness of roughly 0.81), rather than an accelerated learning curve induced by the distillation process. Because this phenomenon represents a single baseline dip over a single random seed, it cannot be reliably concluded that distillation accelerates early model convergence. Following Epoch 6, the differential effectively neutralizes, hovering around zero for the remainder of the training cycle.

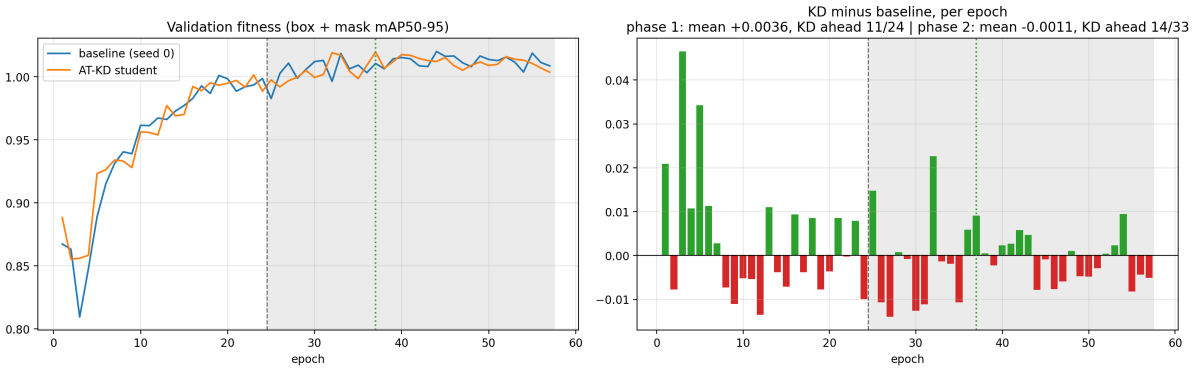


Fig. 16. Validation Fitness Trajectory Highlighting Early-Epoch Baseline Variance and Transition Boundary

False-Positive Behavior, Texture Bias, and Dominant Failure Modes

The primary catalyst for implementing AT-KD was the hypothesis that lightweight networks inherently suffer from texture bias, specifically misclassifying benign spots (such as lenticel or minor cosmetic blemishes) as quarantine hazards (critical rotting or pest damage) due to shared superficial chromaticity. To quantify this, Table XXI cross-examines the normalized class-confusion rates at a 0.25 confidence threshold.

TABLE XXI Class-Confusion Rates (Fraction of True Class Predicted as Alternate Class at 0.25 Conf.)

True Class → Predicted Class	**Teacher (Large)**	**Baseline (Seed 0)**	**AT-KD Student**
defect_lenticel → defect_critical	0.000	0.000	0.000
defect_cosmetic → defect_critical	0.000	0.000	0.000
defect_cosmetic → defect_necrotic	0.024	0.017	0.024
defect_necrotic → defect_cosmetic	0.088	0.129	0.112

Strikingly, the targeted failure mode (benign blemishes misidentified as critical quarantine hazards) recorded a misclas-

sification rate of 0.000 across all architectural candidates. Because the baseline YOLOv8-Nano did not exhibit the hypothesized texture bias on this dataset, there was no false-positive rate for the distillation pipeline to reduce. Instead, an analysis of miss rates (instances entirely undetected by the AI) outlined in Table XXII reveals that the actual dominant failure mode is under-detection, rather than misclassification.

TABLE XXII Absolute Miss Rates (True Instances Undetected at 0.25 Confidence)

Morphological Class	Teacher (Large)	Baseline (Seed 0)	AT-KD Student
defect_lenticel	0.658	0.575	0.767
defect_cosmetic	0.568	0.504	0.594
defect_necrotic	0.176	0.141	0.188
defect_critical	0.032	0.032	0.065
whole_mango	0.004	0.004	0.000

To understand the origin of these high miss rates for superficial blemishes, a spatial diagnostic was performed on the validation split utilizing Intersection-over-Union (IoU) mapping (Table XXIII).

TABLE XXIII Diagnostic of Validation Misses for Lenticel and Cosmetic Classes (Conf. 0.05, IoU 0.45)

Diagnostic Outcome	**Lenticel (191 instances)**	**Cosmetic (1,404 instances)**
Found, confidence ≥ 0.25	38 (20%), median 144 px	611 (44%), median 52 px
Found, low confidence (< 0.25)	27 (14%), median 99 px	250 (18%), median 34 px
Wrong Class Predicted	34 (18%), median 110 px	74 (5%), median 46 px
Box nearby, but IoU too low	32 (17%), median 128 px	205 (15%), median 54 px
Nothing predicted	60 (31%), median 27 px	264 (19%), median 21 px

For the defect_cosmetic class, the likelihood of detection correlates directly with pixel volume; instances that triggered no prediction had a median size of only 21 pixels. Conversely, defect_lenticel misses were not isolated to microscopic clusters (median size ≥ 99 px for 131 out of 191 cases), suggesting morphological ambiguity and inconsistency in ground-truth labeled polygon extents more than algorithmic resolution limits (to be con-

firmed by a label audit). An experimental probe executing the 640×640 trained models at a 960×960 inference resolution strictly degraded global performance (baseline mAP_{50-95} shifted from 0.484 to 0.355) due to spatial scaling mismatch, so higher-resolution inference without retraining is counterproductive; because the models were trained at 640 px, this probe does not test whether training at a higher resolution would help. Typical failures are shown in Figs. 17 and 18.

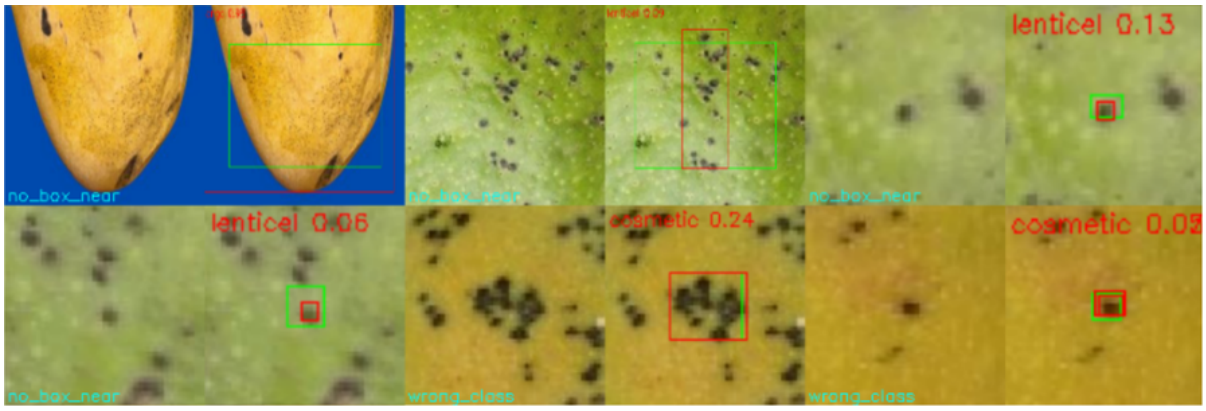


Fig. 17. Diagnostic Visualization of Typical Lenticel Instance Detection Failures

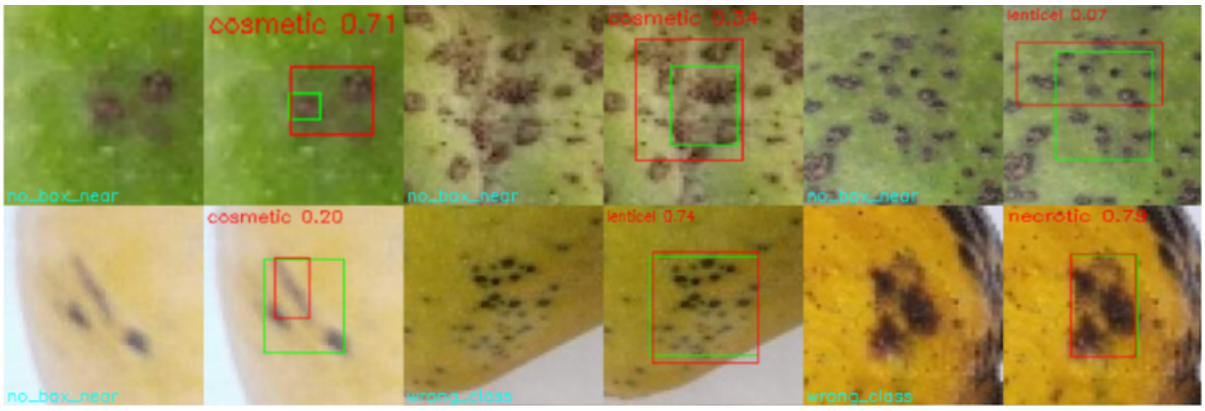


Fig. 18. Diagnostic Visualization of Cosmetic Blemish Detection Failures Correlated with Pixel Volume

Robustness to Color, Texture, and Noise

To objectively evaluate whether the distilled student model internalized depth-aware, structural feature abstractions rather than merely memorizing local RGB pixel intensity, the held-out test split was perturbed to induce visual degradation. Fig. 19 shows samples of each variant.

TABLE XXIV Model Robustness against Perturbed Test Splits (mAP_{50-95})

Test Split	Baseline YOLOv8-Nano (Seed 0)	AT-KD Student YOLOv8-Nano
Clean Baseline	0.498	0.481
Grayscale	0.431	0.406
Blur ($\sigma = 3$)	0.453	0.447
Noise ($\sigma = 12$)	0.491	0.481

As shown in Table XXIV, both models suffered their sharpest performance decay under total chromatic desaturation (Grayscale conversion). The Baseline dropped by an absolute margin of -0.067, while the AT-KD student degraded by -0.075. Under simulated optical convolution (Gaussian Blur $\sigma = 3$), the baseline dropped by -0.045 compared to the student's -0.034. The differences in degradation between the two models are small relative to the run-to-run variation observed between baseline seeds. Consequently, these metrics show no measurable improvement in robustness or resistance to optical degradation from Attention-Based Knowledge Distillation. The

larger drop under grayscale suggests that both models rely on color cues.

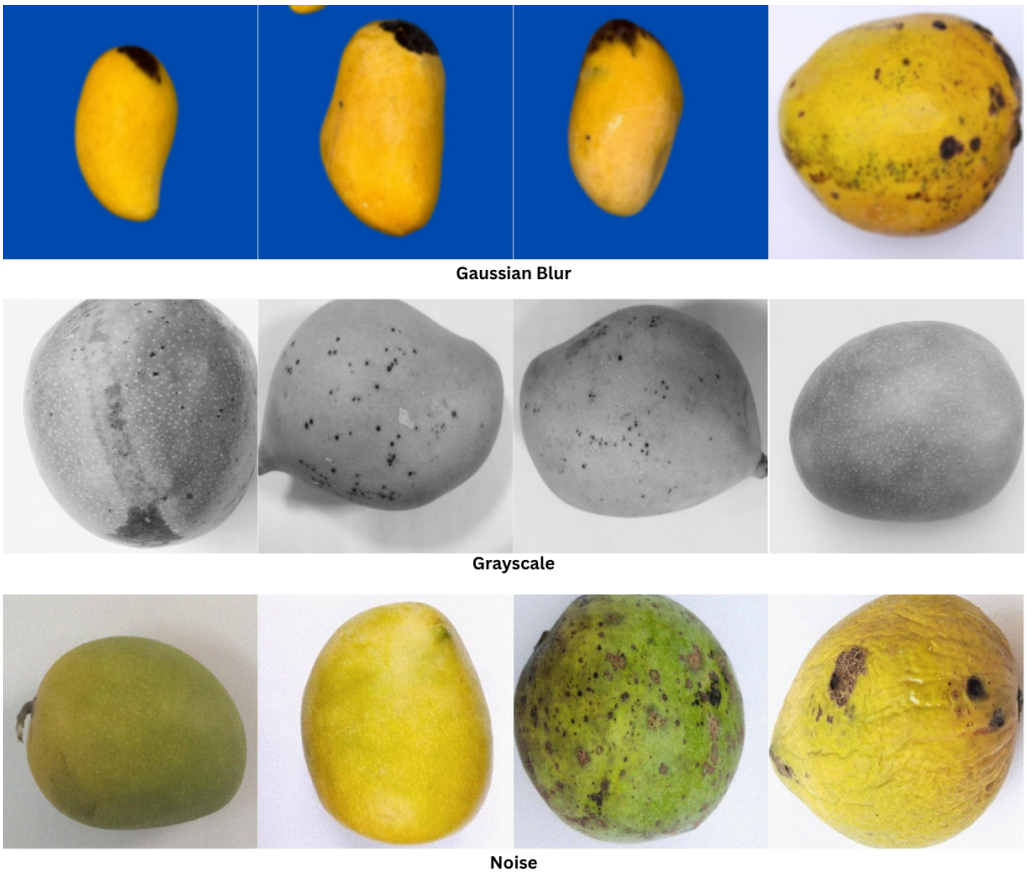


Fig. 19. Sample Visualizations of the Perturbed Held-Out Test Set used for Robustness Verification

Model Selection for Deployment

Based on the comprehensive, multi-tiered algorithmic evaluation, the Baseline YOLOv8-Nano model and the AT-KD Student model are tied within the seed-to-seed variation in segmentation accuracy, false-positive mitigation, and perturbation robustness. Because the YOLOv8-Large teacher failed to supply a superior super-

visory gradient, the distillation pipeline induced a 1.5× computational penalty during training without delivering a quantifiable inference advantage.

Therefore, to directly address Objective 2 while maintaining maximum computational efficiency, the baseline YOLOv8-Nano (seed 0) was selected as the final inference engine, exported via the ONNX framework for hardware deployment on the Raspberry Pi 5. Because the AT-KD student offered no measurable advantage at identical deployment cost, Objective 4 was carried out with the baseline model. The control software applies a confidence threshold of 0.4, which is higher than the 0.25 used for Table XXII, so the miss rates reported there are expected to be lower bounds for the deployed system. While Attention-Based Knowledge Distillation is a theoretically sound mechanism, its application to post-harvest mango classification is unnecessary on this dataset when the foundational architectural capacity of a Nano-tier CNN already meets or exceeds the inherent limits of the dataset's ground-truth variance.

System Evaluation Results

Hardware-Level Efficiency on the Raspberry Pi 5

To evaluate the deployment viability of the MangoSort framework for decentralized rural packinghouses, the selected YOLOv8-Nano model (mangosort_ai.onnx) was executed natively on a Raspberry Pi 5 Model B Rev 1.1 (4GB RAM) utilizing the Ultralytics 8.4.50 runtime. Hardware metrics were recorded using a custom 60-minute continuous benchmarking script that replicates the GUI's exact per-frame operational loop (inference, polygon mask rendering, and display resizing) running against a live 1640×1232 pixel camera stream at a 640 px model input size. The system was cooled by the official Raspberry Pi Active Cooler. The full benchmark environment is listed in Table XXVII (Appendix C). The results of the 60-minute stress test (9,101 processed frames) are detailed in Table XXV.

TABLE XXV Real-Time Hardware Inference and Resource Utilization on Raspberry Pi 5 (60-Minute Run)

Hardware Metric	**Recorded Value**
Inference-only speed (FPS)	4.1
Control-loop speed incl. 30 ms GUI delay (FPS)	2.4
Median latency, predict() (ms per frame)	243.0
90th / 99th percentile inference latency (ms per frame)	260.2 / 290.9
Full loop (inference + drawing + resize), median / p90 (ms)	388.6 / 406.8
Average CPU load, mean (max) (%)	87% (97%)
Process memory (RSS), start / end / peak (MB)	1067 / 1226 / 1264
CPU temperature, start / peak / end (°C)	56.8 / 65.0 / 62.2
Throttling events (firmware flags)	None (0x0 throughout)

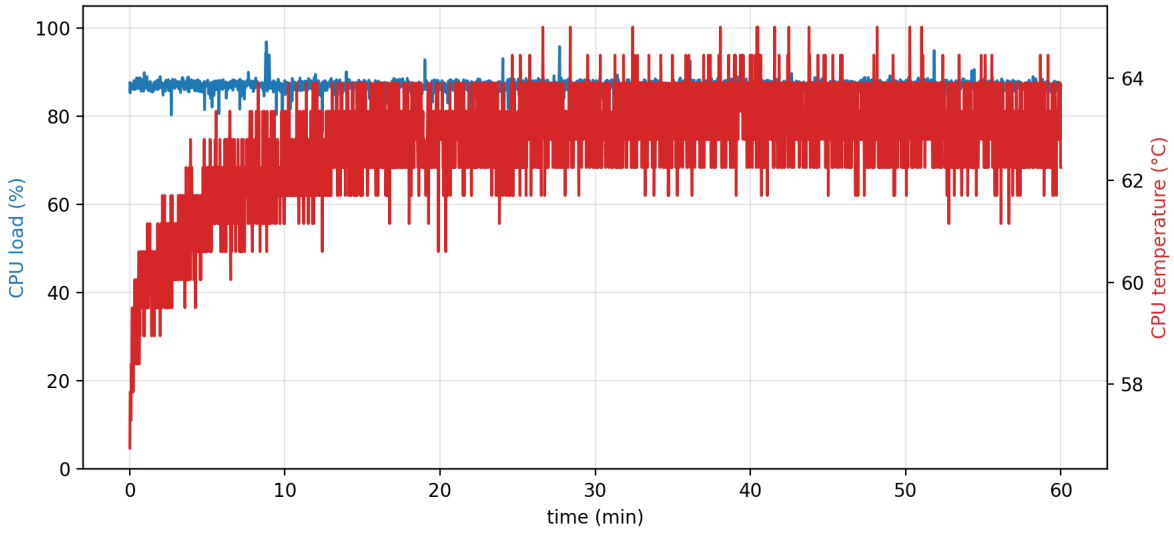


Fig. 20. CPU Load and Temperature During the 60-Minute Raspberry Pi 5 Benchmark (blue: CPU load; red: CPU temperature)

The 60-minute run shows that the prototype sustained continuous inference without thermal limiting. Despite an average CPU load of 87% for a full hour, the CPU temperature peaked at 65.0°C

and the firmware reported no throttling flags (0x0) throughout, which indicates that the Active Cooler was adequate for this test (Fig. 20). Process memory grew from 1,067 MB to 1,226 MB, a linear-fit slope of +2.53 MB/minute that did not level off during the hour. If the growth stayed linear, an 8-hour shift would add roughly 1.2 GB (a calculated extrapolation), which should still fit in the 4 GB of RAM but leaves less margin, so restarting the application at the start of each shift is recommended.

Calculated Capture-Window and Throughput Ceilings

To determine whether the recorded 2.4 FPS control-loop speed is sufficient for automated sorting, the system's operational speed must be contextualized against the physical conveyor parameters. The physical prototype operates at a bounded belt speed of $v = 50$ mm/s. The control software is programmed to grade a fruit when its detected centroid enters a central target circle with a radius of $R = 60$ pixels.

With the calibration constant (K-Ratio) of 33.0 pixels/cm, the physical span of this target circle on the belt is 36.4 mm. Consequently, the fruit dwells inside the target zone for approximately 0.73 seconds. To capture the fruit at least once within this window, the system must process frames at a minimum rate of 1.4 FPS. The measured control-loop period of about 419 ms (the 388.6 ms

full-loop median plus the 30 ms GUI delay, i.e., 2.4 FPS) comfortably satisfies this requirement, yielding a margin of roughly 1.7 frame periods per dwell cycle.

Furthermore, the throughput ceiling is set by the belt rather than by the AI. Assuming an average Carabao mango length of 99 mm along the belt (calculated, not measured, from a published average weight of 230 g [45], a specific gravity of 1.02 g/cm^3 and an assumed length-to-width ratio of 1.5) and a conservative gap of 50 mm between fruits, the belt can carry at most $60 \times 50/(99+50) \approx 20$ fruits per minute. Because the measured control-loop rate meets the single-look requirement, the AI is not expected to be the limiting factor at 50 mm/s.

Virtual Weighing Accuracy (Test A)

The automated assignment of strict cosmetic defect tolerances hinges entirely on the accurate resolution of the mango's physical Size Code (PNS/BAFS 13:2024). To isolate the volumetric algorithm from purely visual classification errors, the virtual weighing mechanism was benchmarked independently. A sample of 15 physical Carabao mangoes, spanning weights from 185.8 g (Size 5) to 380.5 g (Size 1), was processed through the viewing tunnel with the calibration constant strictly locked at a K-Ratio of 33.0 pixels/cm at

a camera focal distance of 32 cm. Table XXVI summarizes the results by size tier, and Fig. 21 plots estimated against measured mass.

TABLE XXVI Pixel-to-Metric Virtual Weighing Performance Evaluation

Sample Tier	**Mean Ground Truth (g)**	**Mean Estimated (g)**	**Mean Abs. Error (g)**	**MAPE (%)**
Size 1 & 2 (Large)	338.7	345.5	16.6	5.16
Size 3 (Medium)	276.0	285.5	19.7	7.08
Size 4 & 5 (Small)	213.8	201.4	12.5	5.90
Aggregate System Average	–	–	**MAE: 16.82 g**	**MAPE: 5.83%**

The ground-truth weights were established utilizing a high-precision commercial digital scale (calibrated to ±1 g precision). The system achieved an overall Mean Absolute Error (MAE) of 16.82 grams and a Mean Absolute Percentage Error (MAPE) of 5.83%.

An analysis of bias directionality indicates that the volume formula tends to marginally overestimate the weight of certain mangoes (recording an average positive bias of roughly +3.83 g across the sample). This behavior is an expected artifact of the mathematical logic, which relies on a 2D prolate ellipsoid model. If a mango exhibits a flatter physical depth than its overhead minor-axis width implies, the geometric assumption slightly over-calculates

the hidden volume. However, a MAPE of under 6% on 15 fruit suggests that virtual weighing is a viable, low-latency alternative to mechanical load cells. The mean absolute error of 16.82 g is about one third of the 50 g width of a size class, so fruit close to a class boundary can still be assigned the wrong size code; a larger sample is needed to quantify how often this occurs.

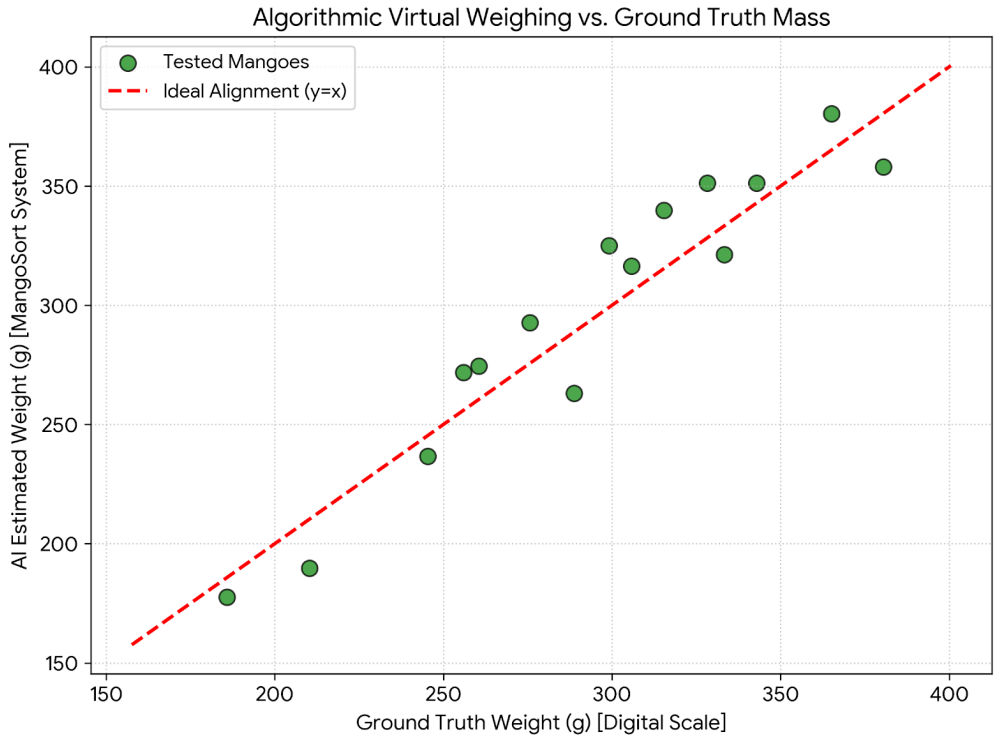


Fig. 21. Correlation Analysis of Algorithmic Virtual Weighing vs. Physical Ground Truth Mass

Limitations of the Study

This study possesses several fundamental limitations that bound how its results should be interpreted. First, the core premise of Attention-Based Knowledge Distillation was not empirically met within this setting: the YOLOv8-Large teacher did not outperform the YOLOv8-Nano baseline on the validation split and scored lower on the held-out test split. Consequently, there was little superior morphological knowledge available for the student network to absorb. Second, the evidence concerning distillation efficacy rests on a single completed distillation run compared against two un-distilled baseline seeds. The two baseline seeds exhibited an internal difference of 0.016 in Mask mAP_{50-95}—roughly equivalent to the performance gap between the baseline and the distilled student—meaning differences of this magnitude cannot be definitively attributed to the distillation mechanism. Additionally, the test split contains only 251 images and a limited number of defect_critical instances, which renders per-class statistical estimates imprecise, particularly for critical and lenticel defects.

Third, the inherent quality of the dataset limited what any neural network could extract. The dataset relies on approximately

1,757 unique baseline photographs expanded five-fold via offline augmentation. Finally, the hardware and system tests used a single prototype at a fixed belt speed of 50 mm/s, whose belt-limited ceiling of about 20 fruits per minute is well below the 60 to 100 fruits per minute of commercial lines, and utilized 15 fruit for virtual weighing. Consequently, results may differ at higher belt speeds, under uncontrolled lighting, for other cultivars, or when subjected to expert human benchmarking.

CHAPTER 5 SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

Summary of the Proposed Study, Design, and Implementation

The manual grading of Carabao mangoes is a labor-intensive, highly subjective bottleneck that frequently leads to inconsistent phytosanitary and cosmetic classification. To address this, the MangoSort system was developed as a decentralized, automated grading framework deployed on a Raspberry Pi 5. The system utilizes a lightweight YOLOv8-Nano instance segmentation model, integrated with a dual-parameter mathematical logic strictly aligned with the Philippine National Standards (PNS/BAFS 13:2024).

The study was driven by four core objectives: (1) to curate a polygon-annotated morphological defect dataset; (2) to train a lightweight YOLOv8-Nano-seg student model via a custom distillation framework to test whether it mitigates hypothesized texture bias; (3) to construct a grading mechanism uniting virtual weighing with exact surface-area calculations; and (4) to deploy and benchmark the model on edge hardware across model-level, hardware-level, and system-level performance tiers.

The system design integrates a controlled optical darkroom tunnel positioned over a conveyor belt, capturing high-resolution RGB imagery. The implementation utilized Roboflow for data ver-

sioning, Ultralytics YOLOv8 and PyTorch running on a cloud-based NVIDIA Tesla T4 GPU for model training, and the ONNX runtime engine for optimized inference on the Raspberry Pi 5. The final output is rendered via a localized Tkinter GUI Control Center, simultaneously appending post-harvest grading data to an offline CSV database.

Summary of Findings

Based on the comprehensive evaluation protocols executed during the study, the following findings were established relative to the primary research objectives:

1. **Dataset Curation (Objective 1):** The study successfully constructed a targeted morphological dataset consisting of 2,510 unique photographs and 12,957 polygon instances, categorized into five specific tiers aligned with PNS/BAFS 13:2024. A 70/20/10 split separated by unique photograph was maintained, with the training partition undergoing a $5\times$ augmentation pipeline, yielding a total corpus of 9,538 images for experimental training.
2. **Attention-Based Knowledge Distillation (Objective 2):** The AT-KD framework (attention transfer, with feature-mimicry, logit-distillation, CBAM, and adapter components implemented in code) was engineered, but only the attention-transfer configuration was evaluated, and that evaluation did not show a benefit. The parameter-heavy YOLOv8-Large teacher did not outperform the Nano baseline (Test Mask mAP_{50-95} of 0.470 vs. baseline's 0.498/0.482). The distilled student scored 0.481, which is 0.001 below Baseline Seed 1 and 0.017 below Seed 0,

a gap that cannot be distinguished from the 0.016 seed-to-seed spread. Crucially, the hypothesized "texture bias" (benign blemishes misidentified as critical quarantine hazards) did not occur in any model; instead, the dominant failure mode was simple under-detection of small cosmetic marks. The distilled student did not demonstrate enhanced robustness to grayscale, blur, or optical noise perturbations.

3. **Dual-Parameter Grading Mechanism (Objective 3):** The deterministic grading logic successfully integrated both sizing and cosmetic surface-area rules under PNS/BAFS 13:2024. The virtual weighing sub-algorithm estimated fruit mass with a Mean Absolute Error of 16.82 grams and a Mean Absolute Percentage Error (MAPE) of 5.83% across 15 physical specimens, suggesting its feasibility as an optical alternative, pending a larger sample, to mechanical load cells. The automated calculation of defect surface area in cm^2 implements the standard's tolerance tiers in software.
4. **Edge Hardware Deployment (Objective 4):** Deployed natively via ONNX runtime on the Raspberry Pi 5, the selected YOLOv8-Nano model fulfilled the hardware-level efficiency tier. The system sustained 4.1 FPS inference-only and 2.4 FPS in the full control loop (including GUI rendering and debounce

logic). During a 60-minute stress test, the system maintained an 87% average CPU load, peaked at 65.0°C, and recorded zero throttling events. The 2.4 FPS rate satisfies the minimum 1.4 FPS capture requirement for a conveyor moving at 50 mm/s, yielding a calculated operational throughput of up to 20 fruits per minute.

Conclusions

Grounded in the empirical data and statistical analyses extracted during the benchmarking phases, the study draws the following conclusions:

1. The curated morphological dataset successfully supports the robust segmentation of distinct mango defects. Macroscopic elements (whole fruit and critical flesh hazards) are delineated with high precision, while superficial anomalies (cosmetic and lenticel spots) remain computationally challenging to resolve at standard input resolutions.
2. A lightweight edge network (YOLOv8-Nano-seg) matched the YOLOv8-Large-seg model trained under the same protocol on this specific agricultural task, using roughly $\frac{1}{13}$ of the parameter mass and $\frac{1}{17}$ of the computational load. This suggests that, on this dataset, the larger model offered no measurable advantage; the comparison is limited by a single teacher configuration (best epoch 13 of 28) and by label variance in the small-defect classes.
3. The hypothesis that Attention-Based Knowledge Distillation significantly improves accuracy and morphological robustness on this dataset is not supported by the data. Because the tar-

geted texture-bias failure mode was inherently absent in the baseline, and the teacher model lacked a superior supervisory gradient, the distillation mechanism incurred a higher training cost without yielding a quantifiable performance advantage.

4. Virtual weighing from the segmented mask and a prolate-ellipsoid model estimated fruit weight with a mean absolute error of 16.82 g and a MAPE of 5.83% on 15 fruit. This is promising as a low-cost alternative to mechanical load cells, but the error is about one third of the 50 g size-class width and the sample is small, so size-code accuracy near class boundaries remains to be quantified.
5. The Raspberry Pi 5 ran the Nano model for 60 minutes at 640 px without thermal throttling (peak 65.0°C), at 4.1 FPS inference-only and about 2.4 FPS in the full control loop. This is adequate for single-look grading at 50 mm/s but leaves little margin, and memory growth of about 2.5 MB per minute calls for periodic restarts.

Recommendations

Based on the operational insights and methodological limitations identified during the development of the MangoSort system, the following recommendations are directed toward agricultural stakeholders and future researchers:

Engineering and Deployment Recommendations

1. **Dynamic Confidence Thresholding:** It is recommended to deploy the Nano model utilizing class-specific confidence thresholds derived from a precision-recall validation sweep, rather than the flat 0.4 threshold used by the current control software, to effectively mitigate the high miss rates of subtle cosmetic blemishes.
2. **Human-in-the-Loop Review:** Given the inherent difficulty in resolving minute lenticel spots, the grading logic should treat these specific anomalies with caution. It is recommended to implement an algorithmic flag that redirects fruit hovering closely on a grade boundary (e.g., between Class I and Class II) for manual human verification.
3. **Frame-Rate Margin:** At 50 mm/s the control loop (about 2.4 FPS) leaves roughly 1.7 frame periods inside the target

circle, enough for one look but not for three. Reducing the per-frame display cost (mask drawing and resizing took about 146 ms of the 389 ms median loop time) or lowering the belt speed would add margin; any change should be re-benchmarked.

4. **Scheduled Restarts:** Because process memory grew by about 2.5 MB per minute during the 60-minute run, the application should be restarted at the start of each shift until a longer run shows that the growth levels off.

Future Work

To build upon the foundational parameters established by this study, future research should pursue the following avenues:

First, future iterations should execute a comprehensive distillation ablation study. This requires applying distillation without runtime interruptions, utilizing multiple architectural seeds, tuning the distillation weight matrix, and executing the full suite of proposed mechanisms (Logit Distillation, Feature Mimicry, and CBAM both inline and as a training branch). A prerequisite for this is training a significantly stronger teacher model (potentially via higher input resolutions or an ensemble approach), as KD can only transfer knowledge the teacher actually possesses.

Second, the agricultural dataset requires systemic refine-

ment prior to algorithmic adjustments. Future work should establish a rigid labeling manual with explicit visual criteria for defect_lenticel, defect_cosmetic, and defect_necrotic classes. Annotations should be independently executed by multiple domain experts utilizing statistical agreement checks, and the dataset must be aggressively expanded to balance the underrepresented minority classes.

Third, to resolve the detection failures surrounding minute physical blemishes, future researchers should investigate the latency-accuracy trade-offs of training the YOLOv8-Nano model at higher resolutions (e.g., 960×960), applying class-weighted loss functions, or utilizing tiled-inference algorithms, balancing these improvements against the thermal threshold of the Raspberry Pi 5. Finally, evaluating the prototype directly within a high-throughput, chaotic sorting facility across multiple seasonal harvest batches is necessary to determine true environmental generalization beyond the dataset utilized in this study.

References

[1] Bureau of Agriculture and Fisheries Standards, "Philippine national standard: Fresh mangoes - grading and classification (pns/bafs 13:2024)," Department of Agriculture, Philippines, Tech. Rep., 2024. [Online]. Available: https://drive.google.com/file/d/14lw81-9V-6LAl-O-cyynUAUaPHHpK-Ap/view.

[2] Philippine Department of Trade and Industry, "The philippines in the mango global value chain," Department of Trade and Industry, Makati, Philippines, Policy Briefs Series 2018-02, 2018. [Online]. Available: https://dtiwebfiles.s3.ap-southeast-1.amazonaws.com/uploads/2019/11/2018-02-The-Philippines-in-the-Mango-Global-Value-Chain.pdf.

[3] E. R. Bayogan and L. Secretaria, "Appendix 8.17: Enhancing mango fruit quality in asian mango chains – study: The philippines," University of the Philippines Mindanao, Davao City, Philippines, Tech. Rep., Oct. 2019, ACIAR Proj. AGB/2016/009. [Online]. Available: https://research.aciar.gov.au/apmangonet/wp-content/uploads/2019/10/8.17-MQ-Quality-Kit-Study_Philippines.pdf.

[4] T. R. Razak, N. A. K. Faizi, M. H. Ismail, S. S. M. Fauzi, and R. A. J. M. Gining, "Towards capturing mango grading

from human experts - a comprehensive user study," in *2021 IEEE International Conference on Robotics, Automation, Artificial-Intelligence and Internet-of-Things (RAAICON)*, vol. 98, 2021, pp. 1–6. DOI: 10.1109/icraie52900.2021.9703830.

[5] T.-N. Doan and D.-N. Le-Thi, "A novel mango grading system based on image processing and machine learning methods," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 5, 2023. DOI: 10.14569/ijacsa.2023.01405115.

[6] E. Kim, *Yolov8 nano vs yolov8 large*, Medium, Nov. 2023. [Online]. Available: https://medium.com/@elvenkim1/yolov8-nano-vs-yolov8-large-4f21324baa38.

[7] G. Yang, S. Yu, Y. Sheng, and H. Yang, "Attention and feature transfer based knowledge distillation," *Scientific Reports*, vol. 13, no. 1, p. 18 369, Oct. 2023. DOI: 10.1038/s41598-023-43986-y.

[8] D.-L. Borza, A. S. Darabant, T. A. Ileni, and A.-I. Marinescu, "Effective online knowledge distillation via attention-based model ensembling," *Mathematics*, vol. 10, no. 22, p. 4285, Nov. 2022. DOI: 10.3390/math10224285.

[9] K. Fernandez-Stark, V. Couto, and G. Gereffi, "The philippines in the mango global value chain," Duke Center on Globalization, Governance & Competitiveness, Durham, NC, USA, Tech. Rep., 2017. [Online]. Available: https://dtiwebfiles.s3.ap-southeast-1.amazonaws.com/uploads/2019/11/2018-02-The-Philippines-in-the-Mango-Global-Value-Chain.pdf.

[10] Department of Agriculture, "Philippine mango industry roadmap 2021-2025," Republic of the Philippines, Quezon City, Philippines, High Value Crops Development Program, 2021. [Online]. Available: https://pcaf.da.gov.ph/index.php/cir-mango/.

[11] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, Toulon, France, 2017. [Online]. Available: https://arxiv.org/abs/1612.03928.

[12] PCAARRD DOST, *Mango*, PCAARRD's Industry Strategic Science & Technology Program Information System (ISP-IS), n.d. [Online]. Available: https://ispweb.pcaarrd.dost.gov.ph/isp-commodities/mango/.

[13] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 7482–7491. DOI: 10.1109/CVPR.2018.00781.

[14] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, "Efficient parametrization of multi-domain deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, Jun. 2018, pp. 8119–8127. DOI: 10.1109/CVPR.2018.00847.

[15] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. [Online]. Available: https://arxiv.org/abs/1807.06521.

[16] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, Oct. 2016. DOI: 10.1109/JIOT.2016.2579198.

[17] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference*

on Computer Vision (ICCV), Venice, Italy, 2017, pp. 2961–2969. DOI: 10.1109/ICCV.2017.322.

[18] L. Wang and K. J. Yoon, "Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 6, pp. 3048–3068, Jun. 2022. DOI: 10.1109/TPAMI.2021.3055564.

[19] T. Czimmermann *et al.*, "Visual-based defect detection and classification approaches for industrial applications—a survey," *Sensors*, vol. 20, no. 5, p. 1459, Mar. 2020. DOI: 10.3390/s20051459.

[20] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, Jun. 2021. DOI: 10.1007/s11263-021-01453-z.

[21] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=Bygh9j09KX.

[22] Asian Development Bank (ADB) and Southeast Asian Regional Center for Graduate Study and Research in Agriculture (SEARCA), "Analysis of fruit and vegetable value chains in the philippines," Asian Development Bank, Manila, Philippines, Tech. Rep., Jun. 2022. [Online]. Available: https://www.adb.org/sites/default/files/project-documents/52239/52239-001-tacr-en_3.pdf.

[23] Bureau of Plant Industry (BPI), "Hot water treatment protocol for mango export," Department of Agriculture, Philippines, Tech. Rep., n.d. [Online]. Available: https://www.da.gov.ph.

[24] L. F. Arauz, "Mango anthracnose: Economic impact and current options for integrated management," *Plant Disease*, vol. 84, no. 6, p. 600, 2000. DOI: 10.1094/PDIS.2000.84.6.600.

[25] J. Baculo, C. Ruiz, and O. Aran, "Cecid fly defect detection in mangoes using object detection frameworks," in *Proceedings of the IEEE Region 10 Conference (TENCON)*, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9687884.

[26] E. C. G. Canieda, B. G. J. Belonio, M. A. A. Dizon, *et al.*, "Development of a computer vision-based mango grading system

for the philippine mango industry," *Agronomy*, vol. 14, no. 4, p. 831, 2024. DOI: 10.3390/agronomy14040831.

[27] DOST-PCAARRD, *Up cebu's mangga system automates mango grading for domestic and export markets*, Aug. 2023. [Online]. Available: https://www.pcaarrd.dost.gov.ph/index.php/quick-information-dispatch-qid-articles/up-cebus-mangga-system-automates-mango-grading-for-domestic-and-export-markets.

[28] R. Cohort, "Lfa-yolo: Detection of lychee fruit anthracnose based on uav images and deep learning," *Multimedia Tools and Applications*, 2026. [Online]. Available: https://www.researchgate.net/publication/384923080.

[29] R. E. Wibowo, Suharjito, M. Asrol, and D. N. Utama, "Real-time oil palm fruit grading system using smartphone and modified yolov4," *IEEE Access*, vol. 11, pp. 59758–59773, Jun. 2023. DOI: 10.1109/ACCESS.2023.3283597.

[30] H. K. M. S. Kumar *et al.*, "Non-destructive maturity determination of mango fruit using deep learning," *Journal of Food Measurement and Characterization*, vol. 15, pp. 2271–2282, 2021. DOI: 10.1007/s11694-021-00825-y.

[31] M. O. Okinda *et al.*, "A machine vision system for combined weight and size grading of citrus fruit," *Biosystems Engineering*, vol. 194, pp. 194–207, 2020. DOI: 10.1016/j.biosystemseng.2020.03.012.

[32] M. S. Mahmud *et al.*, "Iot based fruit quality inspection and lifespan detection system," in *Proceedings of the International Conference on Next-Generation Computing, IoT and Machine Learning (NCIM)*, Gazipur, Bangladesh, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10392254.

[33] Y. Kawaguchi, M. Hara, and K. Yamanishi, "Quantifying shape and texture bias for enhancing transfer learning in convolutional neural networks," *Signals*, vol. 5, no. 4, p. 40, 2024. DOI: 10.3390/signals5040040.

[34] X. Shi, X. Zhang, Y. Su, and X. Zhang, "Positive anchor area merge algorithm: A knowledge distillation algorithm for fruit detection tasks based on yolov8," *IEEE Access*, vol. 13, pp. 34 954–34 968, 2025. DOI: 10.1109/ACCESS.2025.3544361.

[35] H. Yu *et al.*, "Pkd-yolov8: A collaborative pruning and knowledge distillation framework for lightweight rapeseed pest

detection," *Sensors*, vol. 25, no. 16, p. 5004, 2025. DOI: 10.3390/s25165004.

[36] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018. DOI: 10.1016/j.compag.2018.02.016.

[37] G. Hinton, O. Vinyals, and J. Dean, *Distilling the knowledge in a neural network*, 2015. arXiv: 1503.02531. [Online]. Available: https://arxiv.org/abs/1503.02531.

[38] M. Nahiduzzaman *et al.*, "Deep learning-based real-time detection and classification of tomato ripeness stages using yolov8 on raspberry pi," *Engineering Research Express*, vol. 7, no. 1, 2025. DOI: 10.1088/2631-8695/ada720.

[39] M. T. Okano *et al.*, "Edge ai for industrial visual inspection: Yolov8-based visual conformity detection using raspberry pi," *Algorithms*, vol. 18, no. 8, Aug. 2025. DOI: 10.3390/a18080510.

[40] J. G. A. Barbedo, "A review on the use of deep learning in plant disease recognition," *Computers and Electronics in Agriculture*, vol. 144, pp. 369–379, 2018. DOI: 10.1016/j.compag.2017.12.025.

[41] "Development of a computer vision application for mango (mangifera indica l.) fruit defect detection using yolov8 architecture," in *2024 IEEE International Conference*, Oct. 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10730444.

[42] L. Zhang and K. Ma, "Improve object detection with feature-based knowledge distillation: Towards accurate and efficient detectors," in *9th International Conference on Learning Representations (ICLR)*, 2021. [Online]. Available: https://openreview.net/pdf?id=uKhGRvM8QNH.

[43] H. M. R. Iqbal and A. Hakim, *Mango variety and grading dataset*, version V1, Mendeley Data, 2021. DOI: 10.17632/5mc3s86982.1.

[44] P. Barhate, B. Barhate, and M. Joshi, *Mangodhds: A dataset of diseased and healthy mangoes of 1.5 k images*, version V1, Mendeley Data, 2025. DOI: 10.17632/b4nrw5hyyc.1.

[45] R. Avila and A. Joaquin, "Weight estimation of mangoes using machine learning," Aug. 2026.

Appendices

Appendix A Morphological Defect Visual Dictionary

This appendix provides visual examples of the specific mango defects annotated within the dataset, categorized according to the PNS/BAFS 13:2024 standards.



Fig. 22. Examples of Critical/Phytosanitary Defects

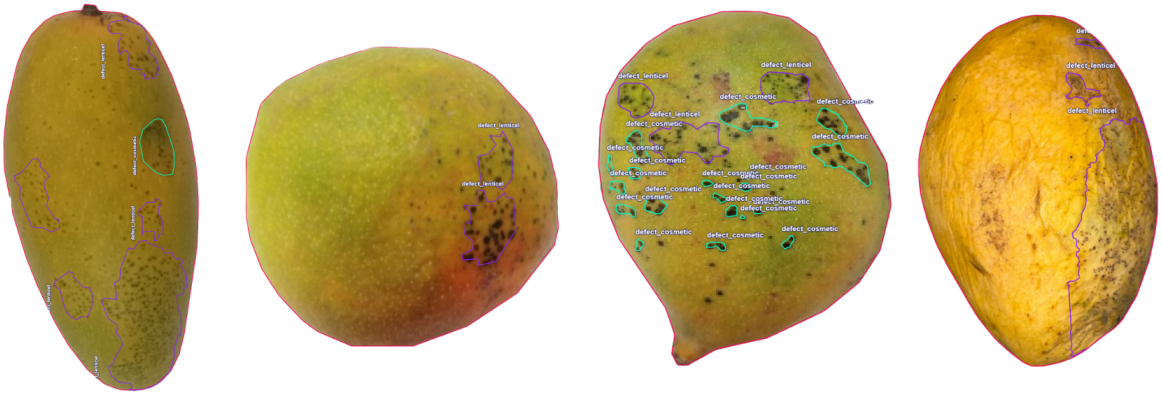


Fig. 23. Examples of Cosmetic and Superficial Blemishes

Appendix B Physical Prototype and Hardware Assembly

The following figures detail the physical construction of the MangoSort edge-computing enclosure, internal component wiring, and integration with the conveyor system.

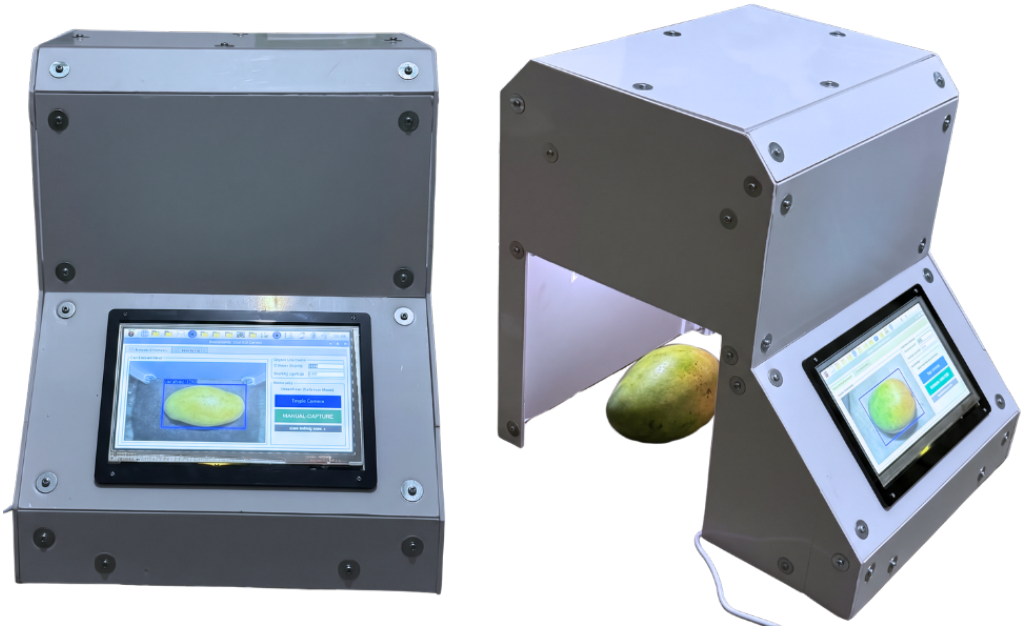


Fig. 24. Fully Assembled MangoSort Physical Prototype

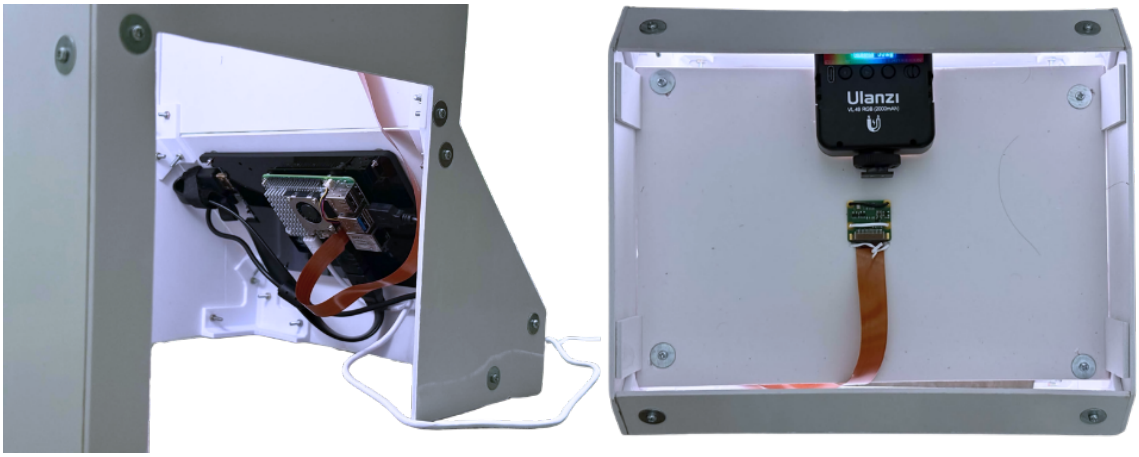


Fig. 25. Internal Darkroom Tunnel, Camera Mount, and Raspberry Pi 5 Wiring Assembly

Appendix C Software Environment and Reproducibility

C.1 Training and Evaluation Environment

Model training and accuracy evaluation ran on Google Colab with an NVIDIA Tesla T4 GPU and Ultralytics 8.4.175, pinned for all runs. The distillation framework (attention transfer, feature mimicry, channel-wise and logit distillation, and CBAM) is implemented in mangosort_kd.py (only attention transfer was activated in the reported runs) and driven by a notebook that trains the teacher, the baselines and the student and evaluates each on the test split. The missed-detection diagnostics were produced with miss_gallery.py.

C.2 Edge Deployment Environment

TABLE XXVII Raspberry Pi 5 Benchmark Environment

Item	**Setting**
Device	Raspberry Pi 5 Model B Rev 1.1 (4GB RAM), four CPU cores, Official Active Cooler
Software	Python 3.13.5, Ultralytics 8.4.50
Model	mangosort_ai.onnx, input size 640 px, confidence threshold 0.4, five classes
Camera stream	Raspberry Pi Camera Module V2, 1640×1232 pixel frames
Benchmark script	pi_benchmark.py: reproduces the control software's per-frame loop and samples CPU load, memory, temperature, clock speed and throttling flags once per second
Control software	mango_gui.py (Tkinter GUI, CSV audit log)

Appendix D Disclaimer

This research project and its corresponding documentation entitled "MangoSort: Mitigating Texture Bias in Lightweight Models for Automated Post-Harvest Grading of Carabao Mangoes via Attention-Based Knowledge Distillation" is submitted to the College of Information and Communications Technology, West Visayas State University, in partial fulfillment of the requirements for the degree, Bachelor of Science in Computer Science. It is the product of our own work, except where indicated text.

We hereby grant the College of Information and Communications Technology permission to freely use, publish in local or international journal/conferences, reproduce, or distribute publicly the paper and electronic copies of this software project and its corresponding documentation in whole or in part, provided that we are acknowledged.

Dallas A. Aquino

Frederick Jibril L. Buñag

Ethan Jed V. Carbonell

Vincent L. Corpes, Jr.

October 2026