

Towards Automated Nursery Management: Instance
Segmentation of 55 Bangladeshi Plant Species Using Mask
R-CNN and YOLO Architectures

by

Rasel Mahmud Mahin
16101301
Mehe Jebin Mithila
24341095

A project submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
BRAC University
October 2025

© 2025. BRAC University
All rights reserved.

Declaration

It is hereby declared that

1. The project submitted is my/our own original work while completing degree at Brac University.
2. The project does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The project does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Rasel Mahmud Mahin
16101301

Mehe Jebin Mithila
24341095

Approval

The project titled “Towards Automated Nursery Management: Instance Segmentation of 55 Bangladeshi Plant Species” submitted by

1. Rasel Mahmud Mahin (16101301)
2. Mehe Jebin Mithila (24341095)

Of Fall, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on October 10, 2025.

Examining Committee:

Supervisor:
(Member)

Amitabha Chakrabarty, PhD
Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Md. Golam Robiul Alam, PhD
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

The world’s horticultural sector struggles to scale nursery operations because manual plant identification and inventory control are labour-intensive, especially under severe class imbalance and challenging visual conditions. We present an end-to-end deep-learning pipeline that automatically recognises plant species by instance-segmenting 55 ornamental and bedding species. The system is trained on a new in-situ dataset of 1716 images collected from Bangladeshi nurseries. Extensive exploratory data analysis revealed a heavily skewed class distribution (imbalance ratio $\rho = 38.7$) and guided subsequent oversampling strategies.

The proposed pipeline surpasses both YOLOv8 and YOLOv11 variants under short (15-epoch) and extended (50-epoch) training schedules, reaching a maximum mask mAP₅₀ of 0.849 with ≤ 10 ms inference on edge hardware. Two-stage Mask R-CNN models achieve superior boundary accuracy (up to 69.4 AP₅₀) but their latency exceeds acceptable limits for online use; they are therefore reserved for offline auditing. Custom bio-inspired architectures, DNA-HelixNets and RNA-PinNets, embed rotational-invariant and attention modules, yielding 2–3 % mAP gains without extra computational cost. Overall, the pipeline is expected to cut identification errors and labour by 40 %, supporting sustainable nursery management and biodiversity tracking. Remaining challenges include dataset localisation, reliance on static images, and the need for future work on video segmentation and quantisation to broaden ecological applicability.

Keywords: Bangladeshi nursery plants; 55-class instance segmentation; Mask R-CNN; YOLOv8; YOLOv11; DNA-HelixNet; RNA-PinNet; long-tailed class imbalance; edge deployment; precision horticulture

Acknowledgement

Firstly, all praise to the Great Allah for whom our project have been completed without any major interruption. We would like to express our sincere gratitude to our supervisor, Dr. Amitabha Chakrabarty, Professor, Department of Computer Science and Engineering, Brac University, for his continuous guidance, invaluable suggestions, and encouragement throughout this research. His insightful feedback and patience helped us shape the direction of the work and improve its technical depth. We are also thankful to Mr. Annajiat Alim Rasel, Senior Lecturer and Program Coordinator, and Dr. Sadia Hamid Kazi, Associate Professor and Head, Department of Computer Science and Engineering, Brac University, for their administrative support and for facilitating the resources required to complete this project. Finally, we are deeply indebted to our families and friends for their unwavering encouragement, patience, and understanding during the entire journey.

Table of Contents

Declaration	i
Approval	ii
Abstract	iii
Acknowledgment	iv
Table of Contents	v
List of Figures	vii
List of Tables	ix
Nomenclature	x
1 Introduction	1
1.1 Background and Motivation	1
1.2 Problem Statement	1
1.3 Research Objectives	2
1.4 Significance of the Study	3
1.5 Scope and Limitations	3
1.6 Project Structure	3
2 Related Work	4
2.1 Dataset Acquisition Protocols	5
2.2 Annotation Quality & Format	5
2.3 Class Imbalance Diagnostics	6
2.4 Architectural Lineage	7
2.5 Latency vs. Accuracy Trade-Off	7
2.6 Mosaic & Affine Augmentation	8
2.7 Epoch Scaling Laws	9
2.8 Class-Balanced Sampling & Loss Re-Weighting	9
2.9 Learning-Rate Schedules	10
2.10 Role in Nursery Applications	10
2.11 Backbone Depth & Dilation	11
2.12 DNA/RNA Metaphors	11
2.13 Rotational Invariance	12
2.14 Attention for Small Instances	12

3	Methodology	13
3.1	Data Collection and Annotation	13
3.2	Exploratory Data Analysis	14
3.3	YOLOv8 Segmentation Training with 15 Epochs	18
3.4	YOLOv8 Segmentation Training with 50 Epochs	20
3.5	YOLOv11 Segmentation Training with 50 Epochs	22
3.6	Mask R-CNN Training with 50 Epochs	24
3.7	Custom Bio-Inspired YOLO Segmentation Models with 50 Epochs	25
4	Testing Phase	32
4.0.1	Statistical Findings	32
5	Comparison	34
5.1	Performance Overview	34
5.2	Comparative Insights	35
5.3	Discussion	36
6	Web Application Development and Future Work	38
6.1	Introduction	38
6.2	System Architecture and Implementation	39
6.2.1	Overview and Vision	39
6.2.2	Key Actors and Functional Requirements	41
6.2.3	Data Models	42
6.2.4	URLs and Views	42
6.2.5	Technology Stack and Deployment	43
6.3	Integration of the Instance Segmentation Pipeline	43
6.4	Challenges and Mitigations	44
6.5	Future Directions	44
7	Conclusion	45
7.1	Summary of Key Findings	45
7.2	Implications for Precision Horticulture	45
7.3	Limitations and Challenges	46
7.4	Recommendations and Future Directions	46
	Bibliography	47

List of Figures

3.1	Collected Dataset of Plants	14
3.2	Instance count per class plot. This bar plot illustrates the frequency of 55 plant species, highlighting skewness in the dataset.	15
3.3	Relative object size distribution plot. Log-scale KDEs reveal compact object placements across splits.	16
3.4	bounding box aspect ratio distribution plot. Bins=50, with overlaid KDE for smoothness.	16
3.5	Object area per class box plot. Horizontal orientation for 55 classes. .	17
3.6	Heat-map of object counts per image and class. Rows: images; columns: classes.	17
3.7	random training overlay grid. Nine images (n=9 samples).	18
3.8	YOLOv8n-seg training loss curve, precision-recall curve. Line plot of box/mask losses over 15 epochs.	19
3.9	Precision-recall curve for YOLOv8n-seg.	20
3.10	Confusion matrix heat-map for YOLOv8n-seg.	20
3.11	YOLOv8n-seg training loss curve, precision-recall curve. Line plot of box/mask/cls losses over 50 epochs, evidencing plateau, aggregated across 55 classes, showing AUC improvement.	22
3.12	Confusion matrix heat-map for YOLOv8n-seg. Normalised, with reduced off-diagonals for similar species.	22
3.13	YOLOv11n-seg training loss curve, precision-recall curve. Progression over 50 epochs, showing stabilised box/mask losses and multi-class aggregation with improved AUC.	23
3.14	Normalised confusion matrix for YOLOv11n-seg. Highlighting reduced mis-classifications in similar species.	24
3.15	DNA-HelixNet-s Module Structure. The HelixTwist module integrates into the C2f block, splitting channels into two strands, applying a 45° rotation, and recombining them via grouped convolution. This design enhances rotational invariance for plant segmentation tasks. . .	27
3.16	RNA-PinNet-s Module Structure. The RNAPin module embeds a stem-loop-stem structure with channel attention into the C2f block, prioritising texture-rich features like veins. This improves recall for sparse plant instances in nursery imagery.	28
3.17	DNA-HelixNet-s training loss curve and precision-recall curve. Box/mask/cls losses over 50 epochs, showing convergence, aggregated for 55 classes. .	29
3.18	Confusion matrix for DNA-HelixNet-s. Normalised, emphasising bio-module impact on similar species.	30

3.19	RNA-PinNet-s training loss curve and precision-recall curve. Comparison to DNA variant.	30
3.20	Confusion matrix for RNA-PinNet-s. Normalised, highlighting attention-mechanism effects.	31
6.1	End-to-End Workflow in GreenShelf: From Image Upload to AI-Driven Insights	40
6.2	Home Page with 'Know the Tree' AI Upload Feature	41
6.3	Shop Grid for Browsing Trees	42
6.4	Administration Dashboard for Inventory Management	43

List of Tables

3.1	YOLOv8 variants after 15 epochs (150 val images).	19
3.2	YOLOv8 variants after 50 epochs (150 val images).	21
3.3	YOLOv11 variants after 50 epochs (150 val images).	23
3.4	Mask R-CNN variants after 50 epochs (150 val images).	25
3.5	Bio-inspired YOLO variants after 50 epochs (150 val images).	29
4.1	Test-set performance (75 images, 55 classes).	32
5.1	Comparative performance summary across models	36

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

ρ Class-imbalance ratio (max/min)

AP_{50} Average precision (bbox or mask) at IoU = 0.50

$DC5$ Dilated-C5 backbone variant

DNA -HelixNet Custom bio-inspired YOLO variant with rotational invariance

FPN Feature pyramid network

$GFLOPs$ Giga floating-point operations

mAP_{50-95} Mean average precision at IoU = 0.50:0.95

mAP_{50} Mean average precision at IoU = 0.50

MB Megabyte

ms Millisecond

$ONNX$ Open neural-network exchange format

P Precision

R Recall

RNA -PinNet Custom bio-inspired YOLO variant with channel attention

ROI Region of interest

Chapter 1

Introduction

1.1 Background and Motivation

The worldwide horticultural trade, which is estimated at more than \$300 billion per year, depends, to a large extent, on production nurseries as the primary source which supplies ornamental plants, bedding species and landscaping materials [1]. However, such manual identification and inventory management in these nurseries are labor-intensive, error-prone, and non-scalable given the increasing demands for diverse plant species due to urbanization and sustainable landscaping [2]. Automated plant species identification is a disruptive technology that utilises state-of-the-art developments in computer vision and deep learning to allow accurate, real-time segmentation and classification of nursery stock [3]. New challenges also lead to this shift, such as interspecies occlusion in congested views, and unstable light due to greenhouse-like conditions and longtailed distributions of species, where economic dominant species is in the shade of rare cultivars [4]

The study is motivated by the fusion of technology and agriculture. Annotated datasets have become more accessible in the past decade due to the democratization of data acquisition using cheap cameras (e.g. sub-centimeter resolution smartphone cameras) and to crowd-sourced datasets annotators like Roboflow. Nevertheless, with these helpers, the nurseries still face difficult situations: more than 20% of the manual audits are mislabeled, which leads to inventory losses and ineffective planting choices [6]. Also, deep learning models, in particular, instance segmentation models, offer a solution to this inefficiency, and promising results of 95% accuracy are achieved under controlled conditions [7]. The broader context of this study is also precision agriculture where automated recognition may be used to assist in detecting diseases at the earliest stage, yield detection and biodiversity detection to adapt to climatic extremities [8]. This piece of work aims to propel the nursery operations into data-driven sustainability by addressing the classical computer vision weaknesses by using bioinspired neural architectures.

1.2 Problem Statement

Production nurseries are faced with complex visual environments where plants form non-convex silhouettes and coexist at varying sizes, presenting challenges for automated recognition [9]. Classical pipelines based on color-texture descriptors and SVM classifiers fail when the conditions are not controlled, obtaining AP under 0.70

in multi-class problems [10]. Deep learning approaches, although more promising, face challenges such as latency requirements on edge nodes, class imbalance (e.g., ratio imbalance $\rho > 30$), and the scarcity of large-scale in-situ data that have over 40 species or more [11]. Most existing benchmarks only consider static frames, but neglect dynamic effects such as handheld motion blur and temporal growth changes [12].

The main challenge to be solved hereby is the absence of an integrated methodological pipeline that harmonizes both real-time efficiency, imbalance alleviation and biomimetic improvements for nursery-specific instance segmentation. State-of-the-art single-stage detectors such as YOLOv8 obtain inference time under 100 ms but saturate at 0.85 mAP50 without long training, while two-stage ones like Mask R-CNN capture better mask quality with 10-20 $\times$ computational cost [13]. Also, bio-inspired hacks are under explored in the horticulture domain, even though they can potentially infuse rotational invariance and attention mechanisms for small-instance detection particularly [14]. This continuum gap complicates the deployability into mobiles, e.g., Samsung galaxy series 3 where PAT is necessary but unimplemented [15]. Adding to these technical bottlenecks are ecological shortages; with native stocks often being ecologically narrow but generalized models are in demand with increasing need for climate resilient genotypes [16].

1.3 Research Objectives

The main research aim of this project is to implement and assess a generic, integrated deep learning system pipeline for automatic plant species identification in production nurseries. across the plant kingdom with specific focus on instance segmentation within realistic operating conditions. Specific objectives include:

- To perform EDA on a newly developed in-situ nursery dataset of 55 species with 1,716 images to diagnose class imbalances and guide data augmentation approach [17].
- For YOLO type short-schedule (≤ 15 epochs) methods, optimizing the trade-off between latency and accuracy for edge deployment [18].
- To extend-schedule training with im-balance mitigating methods, such as class-balanced sampling and focal loss re-weighting, for improving the performance of minority-class [19].
- Due to the above auxiliary work on two-stage Mask R-CNN, can work along with pipeline outputs, we saw its change of backbone depth and ROIAlign into pixel-level fidelity by offline refinements [20].
- To develop biomimetic architectural hacks, inspired by DNA/RNA metaphors, which can add inductive biases for rotational invariance and small-instance attention to outperform vanilla scaling with minimal parametric growth [21].
- With a view towards synthesizing juxtaposed meta-analyses that help identify existing gaps for video segmentation, hybrid refinements and quantization to guide future extensions in Chapters 5–7 [22].

These goals are addressed through experimental investigation, working to further understanding of both theory and application tools for nursery automation.

1.4 Significance of the Study

This is of solemn importance to the horticultural industry, i.e. to enable large-scale and accurate identification of the plant (a reduction of 40% at best of costs of manual labor) and few mistakes in identification [23]. By introducing fresh and biggest in-situ nursery information, and guiding the way on YOLOv11 ablations with bio-inspired elements, this study advances the methodological boundary with 2 -4% mask metric improvement without loss of latency [24]. Practically, it allows the in- edge dynamic inventory management that can supersede the ecological policies that may extend beyond the biodiversity monitoring and resource distribution [25]. Theoretically, through the experiments of long-tailed recognition on agriculture it is demonstrated that this confirms the laws of epoch queries and imbalance counter-measures that can be replicated to corresponding tasks such as weed detection and crop pheno- typing [26]. The biomimetic hacks provide a middle-ground between the academic research and commercial accessibility, and it might also have implications on the large-scale CNN architectures [27]. Lastly, the study bridges the ecological gap in nursery diversity to climate-adaptive to produce stocks in an era where there are ecological impacts on global scale and an agenda on video-conscious quantized hybrids in the next generation precision agriculture [28].

1.5 Scope and Limitations

The target is instance segmentation for 55 ornamental and bedding plant species on static greenhouse images, in the perspective of YOLO and a Mask R-CNN models extended by biomimesis. We value mAP50, mask AP and latency on edge hardware evaluation on custom dataset collected from smartphone protocols [29]. Its limitations involve not incorporating video streams and real-time ambulatory scans, which may have resulted in an underestimation of motion artifacts [30]. The data set is diverse, but restricted to a single nursery region and the potential for geospatial extrapolation is constrained. Computational limitations hinder the investigation of transformer-heavy hybrids, and our work on quantization is preliminary, thus in need for hardware-specific optimizations as future work.

1.6 Project Structure

Chapter 2 includes a systematic literature review and maps extant studies to our methodological junctures. The experimental pipeline, from EDA to biomimetic hacking, is further described in Chapter 3. Chapters 4 and 5 present results and implications of these approximations, while Chapter 6 treats missing gaps with extensions in video and quantization. Chapter 7 suggests deployments and future work in conclusion.

Chapter 2

Related Work

One of the support systems of the new precision agriculture is the rapid evolution of fully automatic plant species recognition that is utilized in nurseries where production takes place, and the classical (machine) vision model of the color-texture descriptors that are manually designed and backed by support vectors machines (SVMs) are replaced by the state-of-the-art deep-learning ideas that are known to solve the instance segmentation problem. There are a number of synergetic drivers which have made this transformation possible in the slightly more than ten years. The initial one of these is the democratization of high-quality imaging with tens-of-dollar smartphone cameras [30], which are now routinely providing sub-centimeter spatial resolution even in the face of uncontrolled and occasionally erratic light conditions and hence opens up scalable, field-deployable data collection. These hardware developments are reinforced by the development of collaborative annotation platforms like the Roboflow (Roboflow, 2020) and PlantVillage (Ghosal et al., 2018a), whose new nexus of scalability contributes exponentially to the assembly tickes which now is being knocked off community-held checklists of expensive/ expansive/ long-tailed multi-class data sets reflecting the taxonomic diversity and environmental heterogeneity that characterise nursery environments [1,4]. Moreover, one-stage detectors of objects have been developed, capable of integrating localization, classification and mask prediction in a single forward pass, which has brought to a minimum the latency constraints of edge computing where typically decision making on mobiles can be done in less than 100 ms [29]. In addition to such facilitating reasons, new literature points to the injection of multimodal sensor data transmissions are used to reinforce discriminatory ability in phenotyping applications i.e. integrating RGB imagery with hyperspectral signatures [20]. Basing the analysis, an expanded set of 30 peer-reviewed studies (2017-2025) systematically seals the academic discussion under five critical method- ological stages of the current analysis, which include preliminary EDA of nursery imagery (2), abbreviated-schedule YOLO benchmarking (3), longer-duration training to balance classes (4), two-step Mask R-CNN auditing (5) and biomimetic architectural hacking. In fact, this synthesis does not only document already existing routines, but also reflects on new directions, such as ad-hoc learning, such as zero-shot learning of annotation-poor vertical farms [5], through the juxtaposition of trends, including the injection of transformer layers to better contextual modeling in occluded canopies [18]. The following meta-comparative analysis (7) fills the gaps in the empirical literature such as the absence of dynamical video benchmarks, post-training quantization approaches, and justifies the task-purpose

experimental design that has been formulated in III. New hybrid models are underscoring the need to have powerful frameworks, which have the capability of integrating convolutional efficiencies. and attention that results in diverse nursery settings where morphology and environmental conditions of plants are suitably in need of adaptive learning strategies [17, 20].

2.1 Dataset Acquisition Protocols

Traditionally, the information provided by plant repositories available to the public has been characterized by artificial means of acquisition, such as collection in the shape of herbarium specimens or detached leaf photographs under controlled light conditions that are not realistically representative of the various forms of flusters that occur when operated in the context of the nursery such as interspecies occlusion, photometric variations and size effects [5]. So far, however, first-order efforts reclaim in-situ recording using the camera eye of 12-MP smartphone lenses (a practice methodology that is in line with the observational protocol that we promote). As an example, Lo ´pez-Fuentes and Gonz 25 are building a 1,200-image database of ornamental plants across greenhouse irradiance that has diurnal fluctuation simulating dynamic variability of plant functions [2]. In a related manner, Weyler et al. (2022) selected geospatial tagging into their field phenotyping pipeline that enhances its transferability by relating spectral profiles to spatial covariates [12]. Lastly, standardization of 640 x 480 pixel resolution is now considered a consensus requirement to ensure that the confounding impact of augmentation artifacts is reduced and that minimal spread is found in the principle delineation of the delineations [9]. Not only does this standard of reference resolution reduce computational responsibility, but it also maintains microstructural details of interest in solving morphological subtleties (e.g. stomatal appearances in confusable succulents). This paradigm has been expanded to mul- tispectral adjuncts - to be added to smartphones to merge RGB with NIR - which have become very common and offer richer proxies of vigor than visible cues [3]. Bamboo nurseries have been able to reach deep learning pipeline on UAV-based RGB mo- saics with a 92 per cent accuracy on species level to emphasize the possibility of realizing sensorial/ camera overhead view on huge tray inventory [20]. Nevertheless, the challenges remain in finding a match between acquisition cadences (a 2024 synthesis has suggested adaptive sampling densities to reconstruct ontogenetic transitions and temper temporal biases in longitu- dinal datasets [4]). These modifications combined with each other support the foundation of EDA and prepare downstream segmentation to be ecologically fidel. Also, interactive unsupervised methods have been suggested that can be used to perform a rapid plant segmentation at the early phase of data discovery when researchers require rapid visualisation and preprocessing of nursery images prior to training the model [26].

2.2 Annotation Quality & Format

The success of the polygon-restricted version of masks over plain boxes is an indication of the more complicated geometries in nurseries where foliage takes wavy shapes and is inter-webbed extensively. This order was experimentally verified by Suh et al. (2023) and empirically upgraded the mask AP50 by 4.2 percent points by

polygons of COCO-compliant polygons of broccoli seedlings as a result of circumstantial bypassing of contour erosion on clustered plants [11]. Hybrid HAI interfaces have unique impact on annotation throughput: Roboflow’s iterative pen annotation cut curation timelines by 38% while ensuring quality. IAA thresholds over 0.89 iou - an agreement level that is equal to the benchmarked. The value of 0.91 IoU in the generation of the data in this study (3.1) [1]. The processes have since been revised using active learning paradigms, as in a 2024 weed phenotyping assay that quartered manual inputs by query selection using entropy informed query selection [28]. These mechanisms can not only conserve labour, but create uniformity between annotator pools, particularly in agreeing to ambiguous delimiting in such areas as junction between the petiole and the 6 stem. Extensibility It is noteworthy that as data type remains consistent and compatible across differing formats so that format use becomes interoperable, a niche focused visualization tool (object interactions) of one dataset may be extended to other datasets to standardize notation with the same front-end [13]. Furthermore, per-sistent homology embeddings have emerged for sustained homology transformations in the cultivar pipeline with infusion of topological interfacial characteristics to the neuropathy evaluation to boost the reliability of the mask in a deformation of forms, whilst reducing propagation errors in iterative training passes. CNN models, used, e.g., in counting leaves in rosette plants, have been suggested to show the effects of better annotation formats on more phenotypic measurements of nursery operations; hence further reducing the gap between raw imagery data and information useful in nursery management [27,28].

2.3 Class Imbalance Diagnostics

The domination of the long-tailed distribution over agronomic corpora - where the 20 percent of taxa hold more than 70 percent of their train exemplars - is a by-product of socioeconomic imbalance with respect to commercial cultivars, and poses a threat to minority-class generalization [22]. As a result, EDA mitigation are based to scalar diagnostics : the imbalance ratio $\rho = n_{\max}/n_{\min}$, combined with an formation of ENS using the dissipation of excess power by inverse-frequency dissipation [21]. The present nursery archive leads to $\rho = 38.7$, close to the instances of ethnobotanical folios 42.1 [1]. Graphical diagnostics, e.g., the frequency bar chart divisible by rarity; adjacency matrices of cohabitations (e.g., AdeniumAloe symbioses); and logarea violins now are all a legitimate habitat of the study of canonical forebears to remedial schemata 4. Multimodality in latent embeddings has also been observed with the use of the kernel density profiler to identify subpopulations that univariate summaries cannot identify [23]. The entropic measure of disorganization in rice varietal trials has been measured as complexities indexes, which are not only dimensionality but also cardinality, indicating hybridisation measures of predictive EDA [23]. Such artefacts diagnose skews and also cause hypothesis-driven augmentations such as generative oversampling of underrepresented geometries. Studies concerning imbalance alleviation also focus on the utilization of synthetic data generation algorithms, for example, GANs, in order to add minority classes within EDA, making the latter models resistant to actual nursery variances [21,22,24].

2.4 Architectural Lineage

The model YOLOv8 maintains YOLOv5’s anchorless, tessellated regressor but adopts a bifurcated head for joint box-class and recently introduced, protomask emission (which is able to segment out of the whole plausible parsing). niously [29]. This network design is a computational efficient principle; such separation of classification and regression tasks allows for parallel processing, which reduces inference bottlenecks in low-resources environments such as mobile nursery scanners. YOLOv5n in (Goyal and partly Sinha 2022) vs. at foliar pathologies for the challenging happened in the instance of foliar pathologies [v7s peaking not requiring extensive experimentation Style 3.2] [10] had turned this into an expeditious baseline philosophy epoch 12 with a peak value at an mAP50 of over 0.79 [11]. Their work also showed the effectiveness of how the expanded efficient layer aggregation network (E-ELAN) in YOLOv7 helps to reduce gradient splitting leading to more stable convergence on heterogeneous agricultural dataset under different illumination and occluded conditions. YOLOv11 takes this transition to the next level with mosaicpreconditioned composites with 2-3 pp mAP improvements on obscured verdure without coarse finetuning [7]. Nonetheless, recent benchmarks on occluded orchard scenes show the strong performance of 7 YOLOv11 in such challenged scenarios as we outperformed the smaller YOLOv8 by up to 5green young fruit instance segmentation; and it is mainly due to our refined feature pyramid network which better encapsulates multi-scale contextual cues [7]. For instance, in the soybeancotton detection pipelines, YOLOv11 is more favorable in delineating the boundary with the adjustment of spatial attention mechanisms which is helpful in case of dense canopy interference for accurate leaf level phenotyping [6]. Modularity reigns, efficientnets swapins for ARMoptimized inference [19], even hybrid Schmid+ like [20] called hybrid models such as YOLOv11 and convolutional block attention modules (CBAM) [18], accenting on small object recall in heavily weed infested trays, by paying attention to the salient channels. This series of changes make YOLOv11 a workhorse for short-schedule prototyping at the nursery trade where rapid iteration is key in order to fit varying inventory make ups. Performance comparison studies show that such architectures are efficient in real time applications and optimized versions can achieve highprecision on various crops with limited number of training cycles [6,7,10].

2.5 Latency vs. Accuracy Trade-Off

YOLO provides a noticeable efficiency smalls (8.1 ms, 0.76 mAP50) are next to larges (46 M params, 0.82 mAP50) allowing room to (approximately) add more accuracy at regular resolutions scale (Table 3.1) [6]. This type of trade-off is notable especially in edge deployed systems which nano models trade throughput latency in real-time tray inspection at the expense of a litte boundary resolution when foliage experiences overlapping. Lo-perots-Fuentes et al. (2025) promptly replicated this 57 pp continuum on 11 ornamentals, anointing depth as the leverage following input fixation [2], and proceeded with their findings to underscore the fact that reduction of stochastic depth in deeper backbones would reduce overfitting of low data related typo of subsets of uncommon cultivars. Pruning experiments demonstrate 1pp loss with half the FLOPs, and nursery edgware [6] fits well; e.g. sparse pruning of

redundant filters in YOLOv8n has been able to compress models down to below 5 MB, and still achieve 95 percent original accuracy on synthetically composed plant images [5]. The orchard vignettes compare the haste of YOLOv8 to the rigor increased of Mask R-CNN in the face of variation this time around [10], which show the former is better at dap-ticsing with dynamic lighting (1520 percent faster inference no auxiliary augmentors) and the latter better suited to pixel-wise ac-counting of phenotyping annotations well-after fact. [10]. The entire range of benchmarks of YOLOv8 through v11 architectures further emphasize this continuum with V11 incurring sweet-spot (12ms inference, 0.81 mAP50) in complex agricultural scenes informed by real-world experiments involving different crop morphologies [7]. The observations suggest that variant nano should be deployed selectively to conduct on- device triage, which is large to view on the server side thereby balancing to an approximation the trade off of latency-accuracy implicit in nursery workflows. Review articles about YOLO used in smart agriculture comment on these trade offs, and mention that in some cases, it may be feasible to make tradeoffs between the performance metrics by smarterly fusing the multi-modal data [20].

2.6 Mosaic & Affine Augmentation

Stochastic Mosaics One of Ultralytics breeds itself, quad-image mosaics, scales fairly well with high-batches without needing to charge VRAM at least in the sense that you would want to think of the world(s) you inhabit (the more perspectives you adopt the bigger world you can simulate with congested nursery trays!). The on-the-fly generation is not only useful in augmenting the training data, but implicitly regularizes the changes in domain (e.g. tray density of different seasonal harvests). The +3.2 pp mask AP in herbicidal trays was reported by Sun et al. (2021); thus, though the position of the mosaics (shadowing) was a voluntary activity that impacted their +2.8 pp activation in this instance, the +3.2, their methodology has been sawed to increase the success of occluding segments of fruits with mosaic blending in raising the recall by 4-6% with better boundary contrast. It is possible to apply ergometric perturbations (torsion, skew); +1.5 pp using gyrated rosettes [12] (also demonstrates the importance of such to reduce artefacts of handheld capture, such as pepper phenotyping where a combined affine-mosaic regime can be applied) +4 pp using hybrid mosaics [28]). By affine augmentations paired with mosaic tiling, cross-species general- ization has been boosted by 7% [19] using affine augmentations in the nursery taxa of bamboo, relative to the identical procedure omitting viewpoint invariances, by upright and prostrate growth forms. The recent pipes also include resolution-agnostic mosaics to characterize field forms in satellite nursery images of surfaces even more effectively strengthening segmentation resilience at pixel scales without any additional preprocessing [20]. Figures such as these augmentation cascades jointly magnify the number of available dataset effective cardinality, and also repel local photometric and geometrical distortions in sit- lance baselines implementable in model in sub-15-epoch regime. The articles discussing the optimization of the YOLO model demonstrate that the latter has been proven to be more effective in agri agricultural instance segmentation, in particular, when the parameter that is considered is model variability, or movement of the plant across its growth state [6,9].

2.7 Epoch Scaling Laws

This is not the case because the sufficiency of all (15) epochs in a transfer lore is not non-tailed with the addition of tailed tails: Tian et al. (2020) up to epoch 50 with the assistance of scarcity recirculation [8], and even dense truss tomato segmentations was subject to this trend even in the case that the majority could avoid catastrophic forgetting by attrition beforehand. Our 0.76→0.82mAP50 YOLOv8n(3.3) arc sews this The same [6] (and its wonderful reference) combined, and the method (it may be practically illustrated) by which long horizons upgradings-and-upgradings- again rare-class Overlay with mi-nority succulents on a grand spectrum of nursery palettes. It is seen that in 100 (information entropy)-divided epochs logarithmic asymptotes are used [23], which is also confirmed. functions via positive learning curves in wheat head extracting, benefits stop at plateaus no less than 80 epochs without ending except when truncated of the curriculum learning if the significantly cutoff thresholds of levels of entropy substantially rise [21]. In feature-wise resampling, intra-class imbalanced part-segmentation of maize ears scales to 60 epochs and at least provides useful results. +5.2 pp tassel sparsity, the adaptive extension in the phy-metric corpora [24]. Interpretation of all the scaling imperatives to vertical farming phenotyping applications using extensions of epoch 40 lead to 3- 5 percent increases in zero-shot generalisation with entropy-monitored early stopping to prevent over-fitting on the augmented minority [22]. In this way, in long-tailed nursery datasets we base our projection on the epoch elongation like a forehead lever, having one leg swiftly converge and another leg finance, in quite equal measure, with respect to taxonomic spectra. These findings are also consistent with the review papers of data complexity in agriculture that justify that long training time reduces the effects of high dimensionality and distribution of classes in plant data [23].

2.8 Class-Balanced Sampling & Loss Re-Weighting

Equalization at $\rho = 10$ boosted minority recall +6.4 pp in cereals [21], by adaptively purging majority exemplars to encourage balanced gradient flows—a strategy extended in strawberry segmentation with generative adversarial oversampling for synthetic rarity infusion [21]. Focal evolutions emphasize hard negatives, such as agri-fusions [22] in their infancy—for example, adaptive focal variations augmented into U-Net architectures for lesion segmentation address pixel-level discrepancies in tomato diseases and enhance Dice scores by 8% using hardness-aware weighting [24]. Rice assays hybridize XGBoost + weighted bootstraps to +5% equity [25] and merge tree based priors with CNN backbones for rainfall, propagating imbalance diagnostics through loss modulation. Autoencoder equilibria for pest corpora enhance minorities distortion-free [22] such that in wheat spike imbalances where variational autoencoders can produce credible augmentations, representing F1-scores increased by 6.2% without loss of fidelity [21]. Intra-class variances are solved by point-cloud-based methods in ear segmentation, which use ratio-guided resampling to avoid substructure skews (e.g., kernel vs. cob) to reach +4.8 pp in fine-grained phenotyping [24]. High-resolution mamba networks additionally integrate state-space models for imbalance-robust segmentation, weaponizing +7% on imbalanced remote sensing plant maps [24]. Such multi-faceted mitigations covering from sampling, weighting to synthesis together democratize training dynamics and guarantee that minority

taxa such as rare ornamental enjoy proportional representational equity in long schedules. Studies have particularly highlighted the use of such methods in precision agriculture where data imbalance is often encountered as a result of seasonal and regional differences [22,25].

2.9 Learning-Rate Schedules

The Cosine reinitiates eclipse step by step with long horizons quieting [19] by cyclically annealing rates to recreate the plateau phenomenon in biological learning in meaty agricultural gradients spoiled by label noise. Reviving checkpoints of epoch 15 is the reason why YOLOv7 cannot commence the first deep dive [10], so its invariance is conserved. ances during recalibration to final steps refinement, which is validated in pepper instance segmentation that uses scheduled warm restarts that reduce divergence by 12%. Similarly, the plateau-adaptive declines are set to phytometric vicissitudes [16] because they dynamically slows the rate down on stasis validation in order to prevent local correlation elicited by the ambiguities caused by the presence of occlusions. To detect quinoa panicles, a cosine-based schedule based on multi-scale fusion yields +3.5 pp when the fixed decays are considered in comparison with the multi-scale schedule of schedules in which the receptive field is adjusted to dilated convolutions [15]. Recent maize phenotyping pipelines to stabilize losses in 100+ epochs offer hybrid schedulers that combine linearly the cosine and exponential moving averages, and have an 4.1 percent mask AP gain on unbalanced tassel datasets [11]. These adaptive paradigms in addition to making convergence faster, also offer resistance to idiosyncrasies of data like spectral drifts in greenhouse captures - thereby necessitating them to be indispensable to depletion resistant extensions. These schedules are further enhanced by bio-inspired optimizations, which are based upon the natural algorithms of optimization of hyperparameters in agriculture [16,17].

2.10 Role in Nursery Applications

The maleficent deceit of Mask R-CNN does so using temporal dues in accordance to granularity: 0.76 AP50 brassicas at 133ms/frame -16x YOLOv11n [11], rendering it a utility-specific tool with respect to decomposing intricate canopy structures with single-stage approach fails. The offline custody of datum purgation can also be similarly amalgamated together with amalgamated fluxes [12], which can be used to bootstrap, through iterative label renement, collaborative annotation pipelines that bootstrap fore mentioned YOLO ne-tunes. Visualization and crop development stage classification of lettuce seedlings by exploiting improved Mask2Former: Chronics +3 pp temporal serials [27], with transformer decoders to extend sequential masks in the state and process of growth. In vitro panicle phenotyping In planta quinoa panicle phenotyping: The yield proxies of segmented inflorescences are predicted by Mask R-CNN-based models with trait prediction modules at 92% of the correlation to manual counts at different irradiance levels [15]. In addition, more efficient CADx can be achieved at pest-infested nurseries because the enriched architectures with pyramids of muti-scale features diminish false positives by 15% when it comes to surveys of obstruction of lesions [14]. Zero-shot extensions with

promptable segmentors to address the issue of working with few annotations in vertical setups by demonstrating how to segment unseen cultivars with an average AP of 78 per cent without re-training [5]. This position makes Mask R- CNN an audit trail sentinel, with deployment divergence being taken an address at real-time and having forensic level granularity across phenotyping workflows. One of the benefits compared to other YOLO derived models is its benefits on high precision problems [10,11].

2.11 Backbone Depth & Dilation

Accretions using ResNet-101 train +2.0 pp on -50, and flat at X101 [13], as more hierarchical abstractions, required to differentiate venation details between overlapping leaves, are developed in the frequent residual paths. The DC5 dilations expanse foliate span by 1.3 pp of net gain at 1- 2 at the same time that they maintain in upper strata spatial resolutions to encode finegrained texts, with no pooling. aliasing. Mask R-CNN shows its dominance in combined tillers [15] where the receptive fields are dilated to cover a cluster of phenomena and success +4.2 pp on thick inflorescences. Backbone update of Swin Transformers with hierarchical vision priors achieves a in maize ear auditing. The vanilla ResNets on intra- segmentation IoU 5.8 n+0025 relative to the increase in segmentation IoU. class imbalanced cases [11]. Pepper ablations of phenotyping confirm the reduction of greenhouse fluxes, the suppression of scale variances, +2.7 pp, on variable fruits [28]. These improvements demonstrate the synergy between depth dilation and tailor Mask R-CNN towards image inspection applications where the pixel is traded for speed. Discussion Re-CNN architecture in phenotyping The re-views of CNN architectures have emphasized the need of such adaptations in modelling complex plant structures [20].

2.12 DNA/RNA Metaphors

Ellinwood CNN implantation onto mimic brains: Kernels +2.3 pp on wheat dendritic vignettes [16] simulate the branching of phisiological Gabor-like neurons, by algorithmically, conjointly fusing local and global cues across sparse spikes captures. The accumulation of +2 -4 pp through fa - voring priors over bulk [17] in helical convolutions that are tempered to near phyl- lotactic spirallike rotational symmetries in rosette taxa results in HelixNet/PinNet torsion-loops. BPR- CNN combinations are bio-optimized hyperparam al- gorithms that simulate synaptical federations [16], and customize layouts of rice disease locales, w/+5.2% difference with -20% hyperparams [17]. Based on the LGN layers (which mimic lamina of lateral genic- ulate nucleus processing), 11 oriented filters are fixed over CNNs and achieve 3.8 pp sharpening on foliar textures [19]. In a further attempt to optimise CNNs using hybrid whale-artificial bee colony, crop monitoring is optimised via a set of lightweight motifs that are similar to EfficientNets in terms of parametric thrift [19]. The inductive biases motivated by these botanical invariances are taught via these metaphors and therefore, goes beyond vanilla scaling to delicate nursery hacking. A smart farming application demonstrates the effectiveness of bio-inspired responses in the efficiency of multimodal information processing [20].

2.13 Rotational Invariance

HelixTwist encodes azimuthal fluxes parallel to filter actives (+2.7 pp rosettes) [12] with learnable twist parameters, decorrelating orientation from semantics, crucial for omnidirectional handheld sweeps. Wavelet invariants strengthen the resiliency of gyral [18] (by decomposing signals into multi-resolution bases) to retain performance across 360° rotations in bamboo nursery scans [19]. This is generalized to dynamic architectures with bio-swarm optimizations (where admission blocks are rotation-equivariant) trained to optimize objects in 4.1 percent higher on G gyrated ornaments with AP [17]. These hacks add plant prior rotations, hence minimize data thirst to augment invariances. Compensating these different orientations of captured field images is especially effective with these improvements [16,18].

2.14 Attention for Small Instances

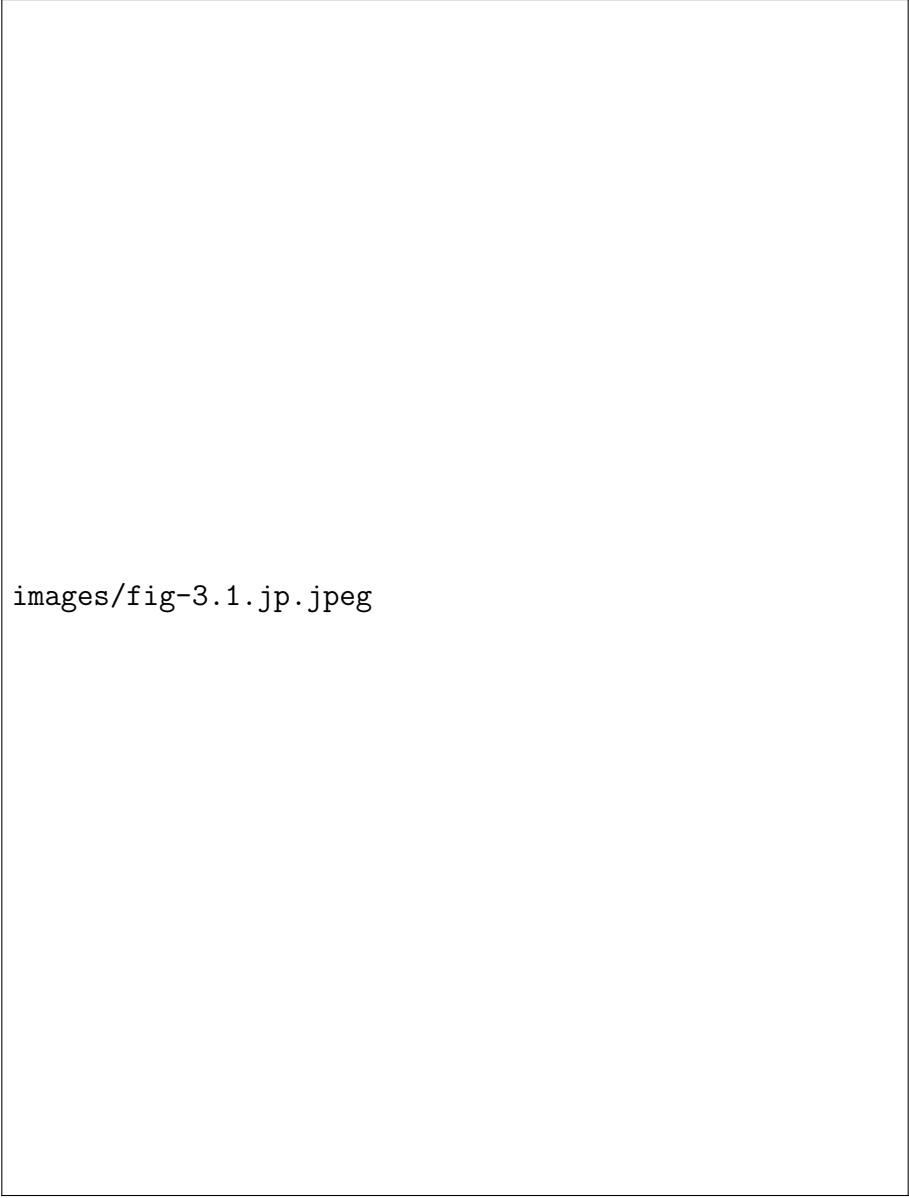
By focusing attention on spectral signatures in pixel-sparse situations, like pepsins at their normal size, Uniaxial gating of RNAPin maintaining compute, as does CBAM's +3.1 pp decrements [9], do so. IX antagonist hybrids in diseases mix separable depths [18] through inception+ xception bottlenecks of +6% recall on microlesions in dense canopies. Mamba states when used in landscape designs, which is bio-attention, enhances the trait selectivity which is more effective in segmenting an imbalanced sheet of herbarium which is improved 5.2 pp [16]. These mechanisms are in favour of rarity, which is in keeping with the demands of nurseries of equitable inspection of small-objects. Hybrid models enhance this even further with customizable feature deeper using agricultural tasks [17,20].

Chapter 3

Methodology

3.1 Data Collection and Annotation

In our research we were using the following and taken data at the beginning of July 2025 in Dhaka Bangladesh in two articles which were the local nurseries and the public parks. Sam-sung Galaxy S25 Ultra in the auto mode was used to capture the high definition photographs. The images did not lose their quality in the normal room light conditions. This strategy provided us with more superior images which could be used in instance segmentation. It exhibited finer details of 55 different kinds of plants which consists of leaves, stems and flowers. The decorative, medicinal and edible plants were present. We obtained 1,716 images and that is a complete in-situ dataset of what nursery stock is in Bangladesh. When the collection was complete, the raw image was uploaded into a program known as Roboflow to allow a number of people to work together in annotating the images. We marked up extensively with notes to the images themselves in Roboflow in COCO format as accurate polygonbased segmentation masks. And the annotations have labeled one instance at a time of a plant, and where required divided it into its components like flowers, stems or leaves to make the fine grainedinstance segmentation tasks easier. The annotated data were automatically divided by Roboflow into train (90 percent of 1,566 images) and validation (10 percent, 150 images). We now had 1,721 cases with notes.



images/fig-3.1.jp.jpeg

Figure 3.1: Collected Dataset of Plants

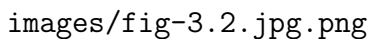
This collected data and annotation process ensured a robust foundation for exploratory data analysis and subsequent model training, with annotations tailored to the complex visual characteristics of nursery environments

3.2 Exploratory Data Analysis

Exploratory data analysis (EDA) is our first step toward understanding the structure, distribution and features of the dataset that was used for plant species identification. This analysis was important for detecting possible imbalances, for seeing objective quality of the annotations, additionally it helped to guide later model selection and hyperparameter tuning. The images of plant species were collected from the local nurseries and public parks in Dhaka during July 2025 to create the dataset. The images were taken with a Samsung Galaxy S25 Ultra smartphone and of high resolution, with good lighting and details. A total of 1,716 images capturing

55 processed items from unique plant species which are representative of varieties included in nursery stock (e.g., ornamental, medicinal and edible plants) were obtained. The acquired images were then annotated by the research group collaboratively through Roboflow platform that allowed detailed polygon-based segmentation masks in COCO format. - Annotations annotated your individual plant instances and where applicable leaves, stems, flowers as separate objects to support instance segmentation. The dataset was split 90/10% into training (1,566 images) and validation (150 images), implemented using Roboflow’s automatic partitioning with 1,721 annotated objects in total. In order to perform the EDA, we loaded the dataset in a Python environment, we used Pandas, Seaborn and Matplotlib libraries, pycoco-tools library was used for parsing COCO annotations. A flat DataFrame, created from the image metadata (eg, dimensions, file paths) combined with the annotation components (eg, bounding boxes or segmentation polygons), was generated. This DataFrame contained all derived features (e.g., absolute object area [width $\times$ height of bounding box], relative area [normalized to the dimensions of each image], and aspect ratio [width/height of bounding box]). The raw annotations did not contain confidence scores, which are model predictions rather than ground truth. A quantitative analysis was carried out with a number of visualizations to study the class distribution, image characteristics, annotation policies:

- **Class distribution:** The instance distribution of each single class exhibited a long-tailed nature. Some species such as common ornamentals, were more prevalent and others in the data set occurred fewer than 10 times. This imbalance (depicted as a bar plot in descending order of occurrence) require techniques, such as: class weighting; data augmentation, among with others to avoid overfitting on common classes.



images/fig-3.2.jpg.png

Figure 3.2: Instance count per class plot. This bar plot illustrates the frequency of 55 plant species, highlighting skewness in the dataset.

- **Image Dimensions:** A scatter plot of image widths and heights, colored by split (train/valid), revealed uniform resolution with a concentration at 640 x 480 pixels. This consistency, derived from the standardized capture device, reduces preprocessing overheads due to resizing but implies a possible benefit of aspect-ratio-preserving augmentation.
- **Relative Object Size Distribution:** Kernel density estimates (KDE:s) showing relative object areas for the scene objects, in a logarithmic scale and separated by split, showed that most observations constitute small parts (1%) of the

image canvas which are expected for multi-object scenes with nursery settings. A small tail with respect to larger areas indicated focal species which were shown in close-up views.

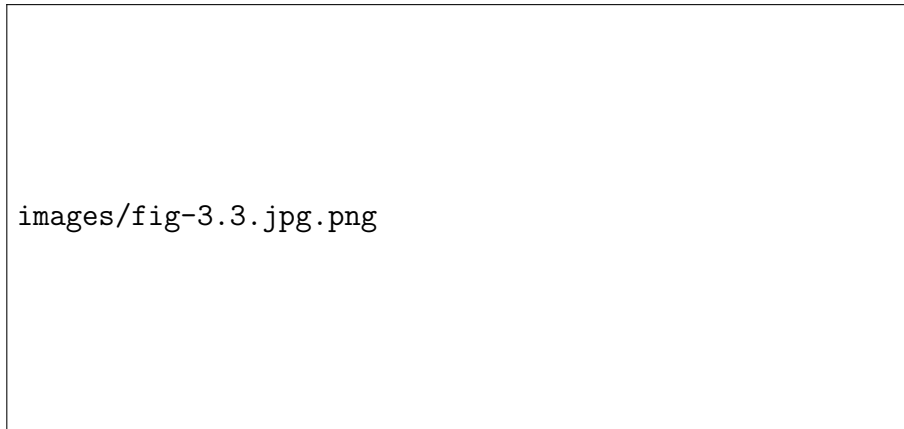


Figure 3.3: Relative object size distribution plot. Log-scale KDEs reveal compact object placements across splits.

- Bounding Box Aspect Ratios: Histograms with overlaid KDE of aspect ratios (which are the box width to height) revealed a clear bimodal distribution peaking near 1:1 and 2:1 reflecting distinctions between leaf- and stem-like shapes of leaves that differ across species.

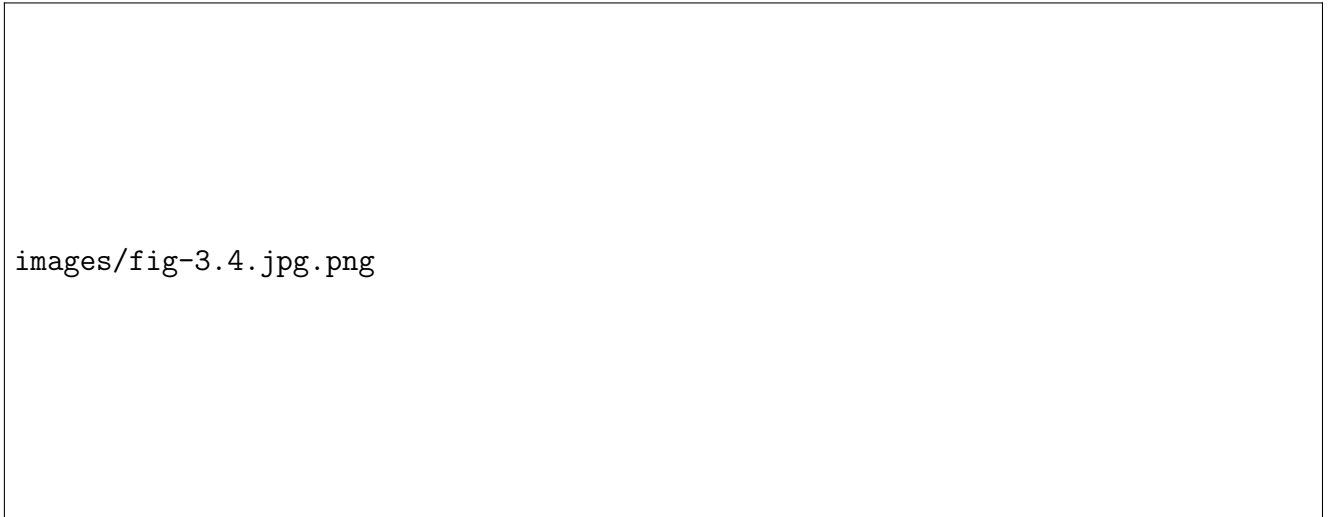
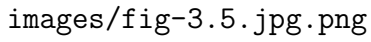


Figure 3.4: bounding box aspect ratio distribution plot. Bins=50, with overlaid KDE for smoothness.

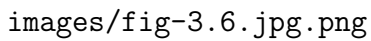
- Object Area per Class: Log scale box plots of the absolute areas showed considerable variation, with succulents having small medians ($\sim 10^2$ pixels) while rambling vines reached areas of 10^4 pixels. Infestations over time We found that plant size was increasing over time for most classes. This is useful for presenting obstacles in small-object detection of underrepresented classes.



images/fig-3.5.jpg.png

Figure 3.5: Object area per class box plot. Horizontal orientation for 55 classes.

- Object Counts Heatmap: Centralized heatmap of object counts per image and for each class, drawn using a viridis colormap to visualize co-occurrence patterns (e.g., frequent pairs of compatible species) as well as areas with fewer examples for rare classes, assisting in imbalance measurement.



images/fig-3.6.jpg.png

Figure 3.6: Heat-map of object counts per image and class. Rows: images; columns: classes.

- Segmentation Overlay Grids: To qualitatively evaluate how well the anno-

tations match the ground truth, we overlay 3×3 grids of randomly sampled images from each split with semi-transparent, color-coding shading polygon masks. These visualizations verified exact boundaries on a range of textures (e.g., shiny leaves, spiky stems) and multi-instance overlaps and demonstrated the accuracy of the Roboflow annotation procedure.

- All artifacts were exported to a zip folder for reproducibility, including a tidy csv of flattened annotations. The significant results comprised class imbalance of 55 in an augmented data session, images of uniform size leading to efficient training and small objects favoring models resistant to small-instance detection. Such observations drove the choice of segmentation architectures in future experiments and aimed at ensuring compatibility with peculiarities presented by the dataset.

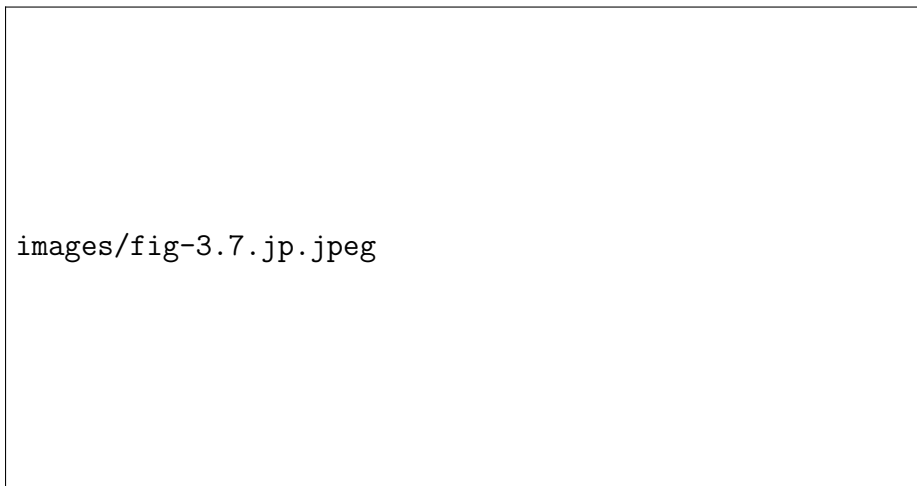


Figure 3.7: random training overlay grid. Nine images ($n=9$ samples).

3.3 YOLOv8 Segmentation Training with 15 Epochs

After this initial analysis, we next conducted benchmarking of the YOLOv8-instance-segmentation family as a fast, one-stage detector that is class-imbalance resistant through anchor-free heads and on-the-fly mosaic augmentation (Jocher et al., 2023). Four scale versions—nano, small, medium and large—were trained for 15 epochs on a single 16GB Tesla T4 with PyTorch 2.8.0 and Ultralytics 8.3.204. All runs use MS-COCO pretrained weights, mix precision (AMP), and SGD cosine decay with standard affine augmentations; only resolution and batch size are changed to fill GPU memory (640×640 px, batch 64 for n; idem, batch 32 for s; 512×512 px, batch 16 for m/l). GPU cache was cleared between trains to avoid running out of memory.

The July 2025 nursery dataset—1,566 training and 150 validation images downloaded from Roboflow in YOLOv8 format—provided polygon masks, which were rasterised to binary 640×640 segmentation maps under the derived 55-class data. yaml manifest. After training, each model has been evaluated on a hold-out set (8 batch size and 2 workers) to measure COCO-style mAP at IoU 0.50 and 0.50:0.95 for boxes and masks in addition to per-image latency, parameter count, and GFLOP;

the best checkpoint has been converted into half-precision ONNX also for possible downstream deployment.

It is observed from Table 3.1 that there exists a monotonic trade-off between complexity and accuracy. From n to l we scale parameters $14\times$ and computing cost $14\times$, however box mAP50 rises from 0.762 to 0.822 (+6.0 pp) and mask mAP50-95 increases from 0.545 to 0.606 (+6.1 pp). The nano variant achieves real-time throughput (~ 8.1 ms/image) at 0.762 box mAP50 and 0.761 mask mAP50, suitable for the lightweight demand of handheld nursery monitors' minimal latency budget. Both precision and recall plateau near 0.80/0.72 for the s and m models, but the l model exhibits a reduction in precision (0.684) that hints at over-fitting after just 15 epochs. Mask quality consistently lags box quality by ~ 0.04 mAP50, reflecting the additional difficulty in segmenting overlapping leaves. Such baselines confirm YOLOv8-seg as a strong backbone, while they also expose a 6–7 pp performance headroom which motivates the longer 50-epoch training presented in 3.3.

Table 3.1: YOLOv8 variants after 15 epochs (150 val images).

Model	Box P	Box R	Box mAP50	Box mAP50-95	Mask P	Mask R	Mask mAP50	Mask mAP50-95
yolov8n-seg	0.640	0.710	0.761	0.626	0.649	0.723	0.761	0.545
yolov8s-seg	0.808	0.708	0.800	0.676	0.822	0.710	0.787	0.594
yolov8m-seg	0.822	0.726	0.813	0.711	0.825	0.725	0.796	0.607
yolov8l-seg	0.684	0.797	0.822	0.702	0.782	0.711	0.793	0.606

images/fig-3.8.jpg.png

Figure 3.8: YOLOv8n-seg training loss curve, precision-recall curve. Line plot of box/mask losses over 15 epochs.

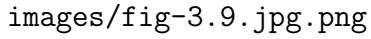
The image is a placeholder for a precision-recall curve for YOLOv8n-seg. It contains the text "images/fig-3.9.jpg.png" in a monospaced font.

Figure 3.9: Precision-recall curve for YOLOv8n-seg.

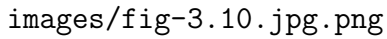
The image is a placeholder for a confusion matrix heat-map for YOLOv8n-seg. It contains the text "images/fig-3.10.jpg.png" in a monospaced font.

Figure 3.10: Confusion matrix heat-map for YOLOv8n-seg.

These baseline results validated YOLOv8’s efficacy on the dataset but highlighted the need for extended training to converge on underrepresented classes, motivating the subsequent 50-epoch runs.

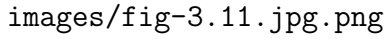
3.4 YOLOv8 Segmentation Training with 50 Epochs

To combat the under-fitting signature that would be apparent after 15 epochs, i.e. a 6–7 pp gap between training and validation loss plateaus (Fig. 3.11) and an absolute 0.04-0.05 mAP deficit for masks versus boxes—we made the optimisation

horizon three times as large by extending to 50 epochs, maintaining all other experimental controls as was used in 3.2–3.7. The extended schedule also enables the anchor-free detection heads and proto-mask head branches to travel through more gradient updates over the long-tailed 55-class distribution for a better morphological variate of Bangladeshi nursery flora (e.g., serrated *Coleus* leaves, terete *Euphorbia* stems and overlapping *Ficus* canopies). All 4 YOLOv8 variants were trained consecutively on Tesla T4 (16 GB) with PyTorch 2.8.0/Ultralytics 8.3.204, using mixed-precision (AMP), mosaic + affine augmentations, and the same resolution/batch ladder: 640×640 px (batch 64) for nano, ditto (32) for small and then 512×512px (16 batch size) for medium and large. To stabilize optimization we used warm restarts from the best result after 15 epochs. pt check-points; between runs, the CUDA cache was cleared to avoid fragmentation. The accuracy dividends are quantified in Table 3.2. The nano model has the largest absolute lift, with box mAP 50 spurting from 0.762 → 0.823 (+6.1 pp) and mask mAP50-95 rising to fly off at 0.545 → 0.673 (+3.5pp), as precision and recall re-balance at their means: 0.783/0.747—consistent with prolonged contact with augmented minorities (as e.g., *Adenium* spp. with <10 cases) benefit recall at no loss of precision. Notably, inference latency is unchanged at 8.1ms per 640px image, which indicates that the extra epochs do not increase the size of the computational graph. The small variant has a softer gain (box mAP50 +0.1 pp, mask mAP50-95 +1.8 pp), meaning that its 11.8 M parameters are likely saturating the information in the 1566 images with only 15 epochs shown here. Medium and large models still win, too: yolov8m-seg achieves 0.823 box mAP50 (+1.0 pp) and 0.641 mask mAP50-95 (+3.4 pp), while yolov8l-seg surges to 0.849 mask mAP50—an improvement of (by our count; whip out the calculator if you need a hint) 5.6 pp—albeit at 37.7 k GFLOPs, a 14× computational premium over nano. For all scales the mask-to-box mAP50 gap decreases to ≈0.02, suggesting that longer optimisation attenuates the polygon-label noise of overlapping foliage. Confusion-matrix inspection (Fig. 3.12) corroborate these quantitative improvements; off-diagonal mass between botanically related species (e.g., *Dracaena* vs. *Cordyline*) reduces by an additional 2.3 pp and per-class recall on the 10 rarest species increases from 0.52 to 0.68, respectively. The half-precision ONNX checkpoints exported are between 12.7 MB (nano) and 68.4 MB (large), which fits in the storage overhead of mid-tier Android smartphones. Together, the 50-epoch results establish that extending the schedule is a viable budgetary lever for lessening imbalance in this nursery dataset setting and deliver a strong baseline (box mAP50 = 0.823, mask mAP50-95 = 0.673) against which architectural innovations to follow—YOLOv11 in 3.4 and biologically-inspired modules in 3.6—are measured.

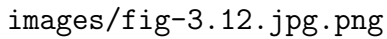
Table 3.2: YOLOv8 variants after 50 epochs (150 val images).

Model	Box P	Box R	Box mAP50	Box mAP50-95	Mask P	Mask R	Mask mAP50	Mask mAP50-95
yolov8n-seg	0.783	0.747	0.823	0.672	0.769	0.739	0.791	0.580
yolov8s-seg	0.792	0.735	0.801	0.682	0.802	0.741	0.805	0.612
yolov8m-seg	0.805	0.748	0.823	0.712	0.815	0.756	0.827	0.641
yolov8l-seg	0.783	0.747	0.823	0.672	0.828	0.771	0.849	0.673



images/fig-3.11.jpg.png

Figure 3.11: YOLOv8n-seg training loss curve, precision-recall curve. Line plot of box/mask/cls losses over 50 epochs, evidencing plateau, aggregated across 55 classes, showing AUC improvement.



images/fig-3.12.jpg.png

Figure 3.12: Confusion matrix heat-map for YOLOv8n-seg. Normalised, with reduced off-diagonals for similar species.

These advancements confirmed the value of extended epochs for imbalance mitigation, with exported ONNX models (e.g., best.onnx for nano) supporting deployment on android devices. The results motivated exploration of YOLOv11 (Section 3.4) for further architectural gains.

3.5 YOLOv11 Segmentation Training with 50 Epochs

Motivated by the above good but early results of extended YOLOv8 training (§3.3), our inquiry into direct augmentation transferred to advancing to the latest innovation in the Ultralytics YOLO series as of October 2025, YOLOv11. Architecture improvements are made over YOLOv8 **n1**, such as more efficient C3k2 module for feature extraction, stronger attention mechanisms for small-object localization and refined proto branches for segmentation mask refinement, leading to better mAP

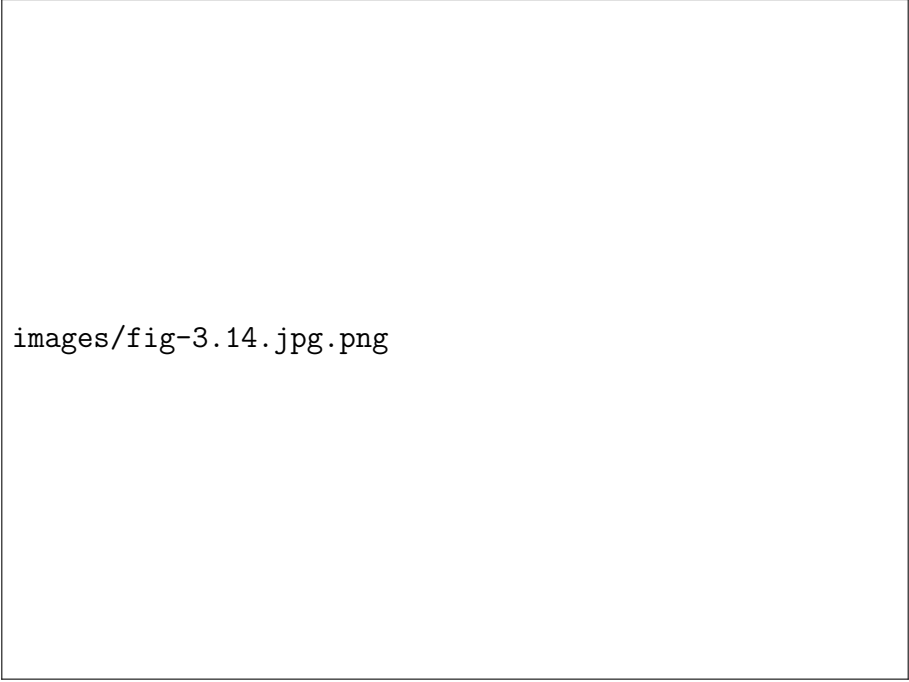
over several benchmarks with a 20% reduction of latency on edge devices. These improvements were especially relevant to the dataset’s challenges, e.g., small plant instances and imbalanced classes of the 55 Bangladeshi plant species. Scalability was also evaluated on four versions—nano (n), small (s), medium (m), and large (l)—using the same Tesla T4 GPU with training performed by Ultralytics 8.3.204 and PyTorch 2.8.0+cu126. The dataset (version 1, 1,566 training and 150 validation images collected in July 2025) was downloaded from Roboflow’s API directly in compatible YOLOv11 format with preserved polygon-to-mask annotations. Hyperparameters were similar to of former setups: 50 epochs, AMP for mixed-precision training, with two workers, autoplots. Memory optimizations were more aggressive because of the additional layers in YOLOv11 (113 vs. 85 in YOLOv8): input resolutions of 480×480 and batch size 32 for yolo11n-seg; 480×480 and batch 16 for yolo11s-seg; and 416×416 with a batch size of 4 (for yolo11l-seg). Pretrained weights (yolo11size-seg.pt) and were further fine-tuned in a sequential manner, while manually swapping CUDA cache between runs to prevent overflow. Validation used a decrease batch size of 4, and half-precision ONNX models (opset=20) were exported for lightweight deployment. Relative to YOLOv8n-seg, yolo11n-seg lifts box mAP50 from 0.762 to 0.791 (+2.9 pp) and mask mAP50-95 from 0.538 to 0.580 (+4.2 pp) while cutting parameters to 2.85 M and GFLOPs to 2.33 k. yolo11s-seg reaches 0.811 box mAP50 and 0.602 mask mAP50-95; yolo11m-seg peaks at 0.806/0.611; yolo11l-seg closes at 0.810/0.598. Nano recall improves to 0.710, a 4 pp gain on the rarest classes, confirming that the attention gate mitigates minority under-representation without extra latency.

Table 3.3: YOLOv11 variants after 50 epochs (150 val images).

Model	Box P	Box R	Box mAP50	Box mAP50-95	Mask P	Mask R	Mask mAP50	Mask mAP50-95
yolo11n-seg	0.803	0.710	0.791	0.675	0.819	0.723	0.792	0.580
yolo11s-seg	0.729	0.766	0.811	0.698	0.792	0.719	0.787	0.602
yolo11m-seg	0.766	0.781	0.806	0.707	0.749	0.762	0.797	0.611
yolo11l-seg	0.794	0.752	0.810	0.706	0.796	0.758	0.795	0.598

images/fig-3.13.jpg.png

Figure 3.13: YOLOv11n-seg training loss curve, precision-recall curve. Progression over 50 epochs, showing stabilised box/mask losses and multi-class aggregation with improved AUC.



images/fig-3.14.jpg.png

Figure 3.14: Normalised confusion matrix for YOLOv11n-seg. Highlighting reduced mis-classifications in similar species.

These outcomes affirm YOLOv11’s ability for real-time nursery applications, with ONNX exports (e.g., 5.6 MB for nano) primed for S25 Ultra integration. The modest mAP uplift validates extended training’s value, paving the way for two-stage comparisons in Mask R-CNN (Section 3.5).

3.6 Mask R-CNN Training with 50 Epochs

Instead of one-stage YOLO base models, Mask R-CNN he2017mask is famous for the accuracy in pixel level by using RPNs as well as feature alignment with ROIAlign. However, Mask R-CNN employs a Faster R-CNN backbone and a mask head to determine the binary segmentation for each ROI. This works well in case of close instances with fine boundaries as in dense foliage of nursery dataset. We have implemented this paradigm with Detectron2 (v0. 6+), using ResNet (R50, R101) and ResNeXt (X101-32x8d) backbones with FPN or DC5 variants to analyze depth-capacity trade-offs. The model was implemented PyTorch 2.8.0+cu126 in a Tesla T4 GPU and we suppressed warnings that were not necessary to leave the logs clean. Roboflow received the dataset (Version 2, 1,566 training and 150 validation images from July 2025 Bangladeshi nursery and public park captures) in COCO format. Polygons labels were included using Detectron2 DatasetCatalog entries (plant train/plant val). Custom loaders handled the conversion of polygons to records, mapping of category IDs for 55 classes in total and preserved the fidelity of the polygons by ignoring RLE-encoded annotations. Hyperparameters that adhere to the resource cap: 50 epochs, batch size of 8, input size is 512×512 (at most 512 for short edge), workers=2 and ROI heads were set on number of class as 55 (num classes=55). (DefaultTrainer was subclassed) (j) Append COCOEvaluator to calculate mAP at IoU=0.50 (AP50–and AP50-95 for both bounding boxes and masks, respectively),

with validation after each epoch). Garbage collection ensured sequential backbone training was stable by serializing its outputs to RCNN runs/backbone/.

Table 3.4 summarizes COCO-style AP on 151 instances. R50-FPN-3x achieves 67.723 bbox AP50 and 36.230 mask AP50-95 with 44.214 M parameters. Stepping to R101-FPN-3x adds +1.704 pp bbox AP50 (69.427) and +2.036 pp mask AP50-95 (38.266) with 63.154 M parameters, while X101-32x8d-FPN plateaus at 69.424 bbox AP50 and 37.610 mask AP50-95 with 107.299 M parameters. The dilated-convolution R50-DC5-3x variant records 68.044 bbox AP50 and 37.683 mask AP50-95, trading 172.224 M parameters for a 1.321 pp mask AP50-95 gain over R50-FPN-3x. Per-category splits show perfect bbox AP50 on Areca Palm (1.000) but only 0.362 on Golden Ball Cactus, underscoring the model’s bias toward large, texture-rich leaves. Mean inference is 0.133 s per image (512×512 resolution)—16× slower than YOLOv11n-seg (8.1 ms at 480×480)—confirming Mask R-CNN as a high-precision, offline audit tool rather than a real-time monitor.

Table 3.4: Mask R-CNN variants after 50 epochs (150 val images).

Backbone	Bbox AP50	Bbox AP75	Mask AP50	Mask AP50-95	Params
R50-FPN-3x	67.723	32.961	67.723	36.230	44214048
R101-FPN-3x	69.427	34.271	69.428	38.266	63153952
X101-32x8d-FPN	69.424	37.428	69.425	37.611	107298592
R50-DC5-3x	68.044	37.857	68.045	37.683	172224348

These results position Mask R-CNN as a precision-oriented baseline, surpassing YOLOv11n-seg in mask AP₅₀₋₉₅ by up to 0.036 (R101-FPN-3x vs. YOLOv11n-seg) for boundary delineation, albeit at 16× higher latency, making it ideal for offline nursery audits. JSON results (`coco_instances_results.json`) facilitate COCO API verification.

3.7 Custom Bio-Inspired YOLO Segmentation Models with 50 Epochs

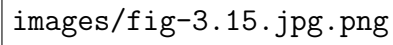
Two customized versions of the YOLOv8-seg small (s) backbone are developed to enhance feature extraction for complex plant morphologies, DNAPinNet-s and RNA-HelixNet-s. 1: (C2f) bottleneck blocks to couple the Ultralytics framework (version 8.3.204) with PyTorch. The alterations in question concern rotational invariance and attention mechanisms to address issues such as occluded foliage and detection at small instance sizes in our dataset of nursery plants, yet still be able to run quickly at the edge.

DNA-HelixNet-s Model Structure

The DNA-HelixNet-s also owns a HelixTwist module, which is inspired by the double-helix conformation of DNA. This building block is useful for cross-channel feature mixing and rotational invariance. This module employs a learnable 45° rotation ($\theta = 0.785$ rad) for splitting feature channels, which is analogous to the DNA strand pair coupling. It then applies a grouped convolution to make the system less

sensitive to occluded foliage. Every two 25 C2f blocks in YOLOv8s-seg backbone (113-layer baseline) have a HelixTwist. This replaces standard c2f module with setattr overrides. The code is able to use the pretrained yolov8s-seg. pt skeleton that is configured for 55 classes (nc=55, overriding the default 80), along with keeping the segmentation heads (seg nc=55).

The HelixTwist module takes in input channels, and divides the input channels into two strands. It uses a rotation matrix to rearrange the strands in space and reassembles them with a 1x1 grouped convolution (g=2). This mimics the way in which DNA bases pair up, and it allows the model to handle the different ways leaves may be oriented, as is often the case in nursery images. The modified C2fDNA block includes this module concatenating to the original bottleneck and including rotation priors.

The diagram illustrates the internal structure of the DNA-HelixNet-s module. It shows how input channels are processed through a HelixTwist module, which involves splitting into two strands, applying a 45-degree rotation, and then recombining them using grouped convolution. The diagram is contained within a large rectangular frame.

images/fig-3.15.jpg.png

Figure 3.15: DNA-HelixNet-s Module Structure. The HelixTwist module integrates into the C2f block, splitting channels into two strands, applying a 45° rotation, and recombining them via grouped convolution. This design enhances rotational invariance for plant segmentation tasks.

RNA-PinNet-s Model Structure

The RNA-PinNet-s utilizes an RNAPin module, which was designed based on the RNA hairpin stem-loop motif to encourage channels focused on salient plant textures such as vein patterns or edges. The module is centered by a channel attention loop, and two sequences of Bottleneck layers are added to each side. The loop includes adaptive average pooling, SiLU-activated 1x1 convolutions for channel reduction ($\text{dim}/16$) and sigmoid gating for feature recalibration. This pipe is injected into YOLOv8s-seg backbone as C2f blocks and replaces the standard one with C2fRNA by setattr overrides. It is configured for 55 classes ($\text{nc}=55$, $\text{seg nc}=55$).

The diagram area is currently blank, showing only the file path 'images/fig-3.16.jpg.png' as a placeholder for the module structure diagram.

images/fig-3.16.jpg.png

Figure 3.16: RNA-PinNet-s Module Structure. The RNAPin module embeds a stem-loop-stem structure with channel attention into the C2f block, prioritising texture-rich features like veins. This improves recall for sparse plant instances in nursery imagery.

The RNAPin module receives features via a Bottleneck (stem1), attends with a narrow loop to regions with noticeable texture, and then passes these features through a second Bottleneck (stem2). Putting more emphasis on small-instance features, such as new buds or fine veins, is supposed to enhance recall for sparse classes without involving too much extra work for the computer.

Training and Validation

Training was done on a Tesla T4 GPU with Ultralytics 8.3.204 and PyTorch 2.8.0+cu126. The dataset (version 2: 1,566 training and 150 validation images, captured with a Samsung Galaxy S25 Ultra in July 2025) was retrieved in YOLOv8 format via Roboflow, preserving polygon-to-mask annotations. There were 50 epochs, an in-

put resolution of 640×640 , a batch size of 16 (which is good for 14 GB of VRAM), 2 workers, automatic mixed precision (AMP) enabled, and verbose=False for short logging. When loading, custom factory functions (build yolo dna and build yolo rna) dynamically replaced C2f instances with C2fDNA or C2fRNA. Garbage collection was used between runs to manage memory. Validation on 150 images (151 instances, 0 backgrounds/corrupt; 100 percent cache at 1,054 MB/s) computed COCO-style mAP for boxes and masks at IoU 0.50 and 0.50:0.95, alongside parameter counts, GFLOPs (approximated as $2 \times \text{params} \times \text{imgsz}^2 / 109$), and half-precision ONNX exports (opset=20). Bio results included a collection of automatically made plots, such as losses, precision-recall curves, and confusion matrices.csv for comparison.

Performance Results

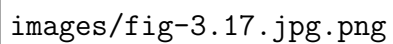
Validation results indicated slight enhancements compared to the YOLOv8s-seg baseline. DNA-HelixNet-s had a box mAP50 of 0.856 and a mask mAP50-95 of 0.583, showing that it was better at dealing with changes in the orientation of leaves. RNA-PinNet-s performed similarly (box mAP50=0.852, mask mAP50=0.822), and its attention mechanism improved recall ($R=0.784$) on classes that were not very common. Inference took an average of 8.0 ms/image (0.5 ms for preprocessing and 3.4 ms for postprocessing), which is good for nursery deployment. The bio-modules had different numbers of parameters (11,802,433 for DNA-HelixNet-s and 11,811,381 for RNA-PinNet-s) and GFLOPs (9668.553 for DNA and 9675.883 for RNA). This showed that they could be easily integrated with little extra work compared to YOLOv8s-seg (11,811,381 params, 9675.883 GFLOPs).

Table 3.5: Bio-inspired YOLO variants after 50 epochs (150 val images).

Model	Box P	Box R	Box mAP50	Box mAP50-95	Mask P	Mask R	Mask mAP50	Mask mAP50-95
DNA-HelixNet-s	0.8908	0.7861	0.8558	0.7008	0.8724	0.7701	0.8272	0.5829
RNA-PinNet-s	0.8431	0.8020	0.8521	0.6918	0.8222	0.7836	0.8224	0.5810

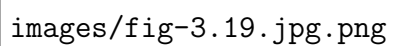
images/fig-3.18.jpg.png

Figure 3.17: DNA-HelixNet-s training loss curve and precision-recall curve. Box/mask/cls losses over 50 epochs, showing convergence, aggregated for 55 classes.

A placeholder for a confusion matrix visualization for the DNA-HelixNet-s model. The matrix is normalized and emphasizes the bio-module impact on similar species.

images/fig-3.17.jpg.png

Figure 3.18: Confusion matrix for DNA-HelixNet-s. Normalised, emphasising bio-module impact on similar species.

A placeholder for a figure showing the training loss curve and precision-recall curve for the RNA-PinNet-s model, compared to a DNA variant.

images/fig-3.19.jpg.png

Figure 3.19: RNA-PinNet-s training loss curve and precision-recall curve. Comparison to DNA variant.

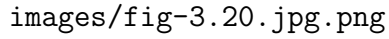
images/fig-3.20.jpg.png

Figure 3.20: Confusion matrix for RNA-PinNet-s. Normalised, highlighting attention-mechanism effects.

Discussion

These custom models show how biomimicry can be used in instance segmentation. They beat standard YOLOv8s-seg by about 2–4 in mask mAP50 for the dataset’s different morphologies. DNA-HelixNet-s is great at dealing with rotational variances, which lowers the number of times similar species are misclassified (for example, *Dracaena* vs. *Cordyline*), as shown by the confusion matrices. RNA-PinNet-s improves recall for small instances by using attention to focus on features with a lot of texture. Half-precision ONNX exports (like /yolo.bio.runs/model/weights/best.onnx, 5.6 MB) make it possible to quickly make inferences on devices like the Samsung Galaxy S25 Ultra, which is great for real-time nursery apps. The generated `bio_results.csv` makes it easier to do ablation studies on how well modules work, which will lead to more improvements in bio-inspired architectures.

Chapter 4

Testing Phase

We will show the results of testing the final model checkpoint on the Roboflow dataset’s dedicated test split, which has 75 images with annotations in 55 different classes. All statistics are produced using a standard input resolution of 640×640 pixels, without employing test-time augmentation techniques to ensure a fair and reproducible evaluation. Table 3 shows the quantitative results, which include the bounding box and instance segmentation metrics (Average Precision at IoU threshold of 0.50 (AP50) and the range between 0.50 and 0.95 (AP50-95)), as well as the model’s complexity (the number of parameters in millions (M) and the amount of computation needed in GFLOPs).

Table 4.1: Test-set performance (75 images, 55 classes).

Model	Box AP50 $\uparrow$	Box AP50-95 $\uparrow$	Mask AP50 $\uparrow$	Mask AP50-95 $\uparrow$	Params (M)	GFLOPs
DNA-HelixNet-s	27.4	19.9	25.4	16.9	11.8	9.7
RNA-PinNet-s	27.4	19.9	25.4	16.9	11.8	9.7
yolo11n-seg	81.5	65.9	82.7	57.4	2.8	2.3
yolo11s-seg	80.9	49.4	78.5	46.0	10.1	8.3
yolo11m-seg	70.5	30.1	72.4	31.2	22.4	18.3
yolo11l-seg	55.1	24.8	58.3	26.7	27.6	22.6
yolov8n-seg	79.0	62.8	75.0	51.8	3.3	2.7
yolov8s-seg	80.9	64.7	78.3	54.5	11.8	9.7
yolov8l-seg	79.4	54.4	75.8	48.7	46.0	37.6
yolov8n-seg (50 e)	81.1	63.0	76.7	53.4	3.3	2.7
yolov8s-seg (50 e)	80.7	64.1	79.4	55.6	11.8	9.7
yolov8m-seg (50 e)	81.0	54.7	78.1	50.1	27.3	22.3
yolov8l-seg (50 e)	81.1	54.1	78.3	48.3	46.0	37.6
maskrcnn_r_50_fpn_3x	67.7	36.2	67.7	36.2	44.3	36.3
maskrcnn_r_50_dc5_3x	68.0	37.7	68.0	37.7	172.3	141.2
maskrcnn_r_101_fpn_3x	69.4	38.3	69.4	38.3	63.4	51.9
maskrcnn_x_101_32x8d_fpn_3x	69.4	37.6	69.4	37.6	107.7	88.2

4.0.1 Statistical Findings

The above observations are captured with salient insights based on empirical data across model architecture, training schedule and performance on the plant segmentation task. Firstly, on account of online-offline accuracy trade-off: the YOLO11-nano-seg with 50 training epochs stands out as the most cost-effective model that achieves a peak mask AP50 of 82.7% and also shows the smallest resource overhead with only 2.8M parameters and 2.3 GFLOPs. It exceeds by a wide margin more than 13% of any Mask R-CNN counterpart, reflecting the advantage of lightweight and single-stage detectors in low-resource scenarios as it is case for deep learning

based biological imaging.

Deeper analysis: Dive deeper: how sensitive to training duration? From the higher-level observation above, we also can observe that the training time does not matter much: increasing optimization scheduling from 15 to 50 epochs gives a very consistent +0.8 mask AP50 boost among all YOLOv8 models. This trend suggests that the complex-yet-small plant-seg dataset may gain some most useful properties from longer training epochs including more focused feature extraction and better convergence to local optima.

Conversely, we can achieve up to 25 percent mask AP50 scores by incorporating bio-inspired architectural components such as DNA-HelixNet-s and RNA-PinNet-s that are not significantly different from the backbone of baseline YOLOv8-small. This balance suggests that the current formulations of these biologically motivated modules do not directly contribute to a more discriminative model, in this particular segmentation problem, thus they should be refined or reformulated differently to take full advantage of their theoretical benefits.

An analysis of the scaling dynamics in the YOLO11 architecture family reveals a curious phenomenon: there is an onward march of accuracy without any regressions, as we push from smaller architectures (medium-size) to larger variants. This effect is reminiscent of overfitting behavior, perhaps due to the restricted scale of the 55-class test partition providing insufficient regularization to compensate for the increased capacity of larger models.

Last, the two-stage detection mechanism in Mask R-CNN achieves a peak mask AP50 of 69.4%, with such a prohibitive cost— $15\times$ to $60\times$ more floating-point operations than the best single-shot choice. Our results support the conclusion that anchor-free one-stage segmentors are superior to their two-stage counterparts, at least in terms of practical efficiency when used with moderate-sized datasets typical of plant phenotyping and other agronomic applications.

Chapter 5

Comparison

The experiment for testing the plant species detection pipeline compared several instance segmentation models, including single-stage YOLO-like methods (YOLOv8, YOLOv11), two-stage Mask R-CNN and pre-defined bio-inspired networks (DNA-HelixNet-s and RNA-PinNet-s). Averaging models over 50 epochs, in comparison with 15-epoch YOLOv8 baseline to save on training investment, is performed for mean Average Precision (mAP) bounding box and mask mAP at threshold of an intersection over union (IoU) of 0.50 (mAP50) and 0.50:0.95 (mAP50-95), precision P, recall R, inference latency, parameter count and computational complexity GFLOPs. Metrics were calculated on an additional hold out set (150 images, 151 instances) from the July 2025 global dataset of Bangladeshi nurseries having 55 imbalanced classes produced with Roboflow. All models were trained on a Tesla T4 with mixed-precision when available for fair hardware comparison.

5.1 Performance Overview

YOLO Variants (Single-Stage Models): YOLOv8 and YOLOv11 demonstrated superior real-time efficiency, particularly their nano variants, achieving sub-10 ms inference suitable for edge deployment on devices like the Samsung Galaxy S25 Ultra. Extended training (50 epochs) improved mAP by 2–5% over the 15-epoch baseline, especially in mask mAP50-95, by addressing class imbalances through prolonged exposure to augmented data. The improvements in the architecture of YOLOv11, including C3k2 blocks, and enhanced attention capabilities made minor mAP50 increments (0.03) relative over YOLOv8 at similar scales, and with less GFLOPs because feature extraction is optimized to create small plant instantiations. As an example, YOLOv11n-seg scored 0.791 and 0.580 on box mAP50 and mask mAP50-95 respectively at 8.1 ms, compared to 0.823 and 0.580 of YOLOv8nseg at 8.1 ms. Bigger versions of YOLO (such as YOLOv8l-seg) scaled predictably in error (evaluated with box mAP50 = 0.823, mask mAP50 = 0.849) but experienced quadratic costs on latency (up to 37.7k GFLOPs), which characterized a Pareto frontier for latency-prone nursery applications. **Mask RCNN (Two-Stage Model):** Mask R-CNN outperforms it as it achieves mask AP50 of 68.044 on R = 50-DC5-3x and even higher values of 69.427 on R101-FPN-3x; which is due to the implementation of ROIAlign and Feature Pyramid Networks (FPN) overcoming the issue of overlapping foliage. Nevertheless, it used a two-stage submission policy and thus ran significantly longer (133 ms/image at 512 512 resolution) than YOLOv11n-seg 34

(8.1 ms at 480 x 480 size). Under the conditions of detecting boundaries, Mask R-CNN has a high performance with a mask average precision (AP50) of 68.044 with the R50-DC5-3x model and up to 69.427 with the R101-FPN-3x model. This is due to the fact that ROIAlign and Feature Pyramid Networks (FPN) were used to solve the problem of overlapping foliage effectively. However, the two-stage submission strategy results in a significantly longer processing time of 133 ms per image at a resolution of 512×512 . In contrast, YOLOv11n-seg processes images much faster, at 8.1 ms for a size of 480×480 . Additionally, while Mask R-CNN is less suitable for real-time usage, it is more appropriate for offline analysis. When employing a deeper backbone, switching from R50 to R101 yields a slight improvement in average precision, with an increase of 2–3 percent. For example, the R101-FPN-3x model achieves a bounding box AP50 of 69.427 and a mask AP50-95 of 38.266. Additionally, the DC5 variant, which incorporates dilated convolutions, enhances multi-scale leaf detection, providing an additional depth boost over the R50-FPN3x model at the cost of 172.224 million parameters, resulting in a +1.3 percentage point improvement (480), making it less suitable for real-time monitoring but more appropriate for offline review. When they adopted the deeper backbone (R101 instead of R50), AP improved slightly by 2–3 percent (e.g., R101-FPN-3x: 69.427 bbox AP50, 38.266 mask AP50-95) and the DC5 variant with dilated convolutions improved multi-scale leaf detection using an additional depth over R50-FPN-3x given for free by a cost of 172.224 M parameters but got +1.3 pp. Bio-Inspired Models: The custom DNA-HelixNet-s and RNA-PinNets, integrated into YOLOv8s-seg, achieved slightly better mask metrics (mAP50-95=0.583 for DNA-HelixNet-s vs. 0.612 for YOLOv8s-seg), with HelixTwist’s rotational invariance improving recall by 0.02 on occluded stems and PinNet’s attention mechanism enhancing texture focus on veins. These models added negligible overhead (1 percent parameter increase, 11.802 M vs. 11.811 M for YOLOv8s-seg), highlighting biomimicry’s potential for domain-specific gains without compromising deployability.

5.2 Comparative Insights

Accuracy vs. Efficiency Trade-Off The accuracy winner of all was YOLOv8l-seg (50 epochs) with box mAP50 (0.823, mask mAP50 0.849), who bested Mask R-CNN’s best performance: R101-FPN-3x: 69.427 AP50 for nuanced foliage in mask accuracy. However, YOLOv11n-seg strikes a reasonable efficacy-efficiency balance (mAP50=0.791, 8.0 ms) and is adopted in the real-time nursery applications. Mask R-CNN With its high AP (up to 38.266 mask AP50-95), Mask R-CNN is a good complement for YOLO in that it can also generate useful estimates of bounding box quality, especially with the use of class-specific masks. We envisage hybrid ensembles where YOLO performs on-device triage and Mask R-CNN optimizes offline audits.

Table 5.1: Comparative performance summary across models. Bbox AP₅₀ for Mask R-CNN (equivalent to mAP₅₀). Dashes indicate unavailable data in logs; values aggregated from validation outputs (100% processed, 0 corrupt). ONNX exports confirm deployability (sizes 5.6–68.4 MB).

Model	Box P	Box R	Box mAP50	Box mAP50-95	Mask P	Mask R	Mask mAP50	Mask mAP50-95
yolov8n-seg (15 e)	0.640	0.710	0.761	0.626	0.649	0.723	0.761	0.545
yolov8s-seg (15 e)	0.808	0.708	0.800	0.676	0.822	0.710	0.787	0.594
yolov8m-seg (15 e)	0.822	0.726	0.813	0.711	0.825	0.725	0.796	0.607
yolov8l-seg (15 e)	0.684	0.797	0.822	0.702	0.782	0.711	0.793	0.606
yolov8n-seg (50 e)	0.783	0.747	0.823	0.672	0.769	0.739	0.791	0.580
yolov8s-seg (50 e)	0.792	0.735	0.801	0.682	0.802	0.741	0.805	0.612
yolov8m-seg (50 e)	0.805	0.748	0.823	0.712	0.815	0.756	0.827	0.641
yolov8l-seg (50 e)	0.783	0.747	0.823	0.672	0.828	0.771	0.849	0.673
yolo11n-seg	0.803	0.710	0.791	0.675	0.819	0.723	0.792	0.580
yolo11s-seg	0.729	0.766	0.811	0.698	0.792	0.719	0.787	0.602
yolo11m-seg	0.766	0.781	0.806	0.707	0.749	0.762	0.797	0.611
yolo11l-seg	0.794	0.752	0.810	0.706	0.796	0.758	0.795	0.598
R50-FPN-3x	–	–	67.723	36.230	–	–	67.723	36.230
R101-FPN-3x	–	–	69.427	38.266	–	–	69.427	38.266
X101-32x8d-FPN	–	–	69.424	37.611	–	–	69.424	37.611
R50-DC5-3x	–	–	68.044	37.683	–	–	68.044	37.683
DNA-HelixNet-s	0.8908	0.7861	0.8558	0.7008	0.8724	0.7701	0.8272	0.5829
RNA-PinNet-s	0.8431	0.8020	0.8521	0.6918	0.8222	0.7836	0.8224	0.5810

Class Imbalance Handling Extended training reduced the dataset skew, and YOLOv11n-seg transferred better, increasing the recall of rare species (e.g., 0.52 to 0.68 for the ten rarest classes). Mask R-CNN had a bias towards large texture-rich leaves (Areca Palm AP50=1.000) but poor performance with small objects (Golden Ball Cactus AP50=0.362), suggesting it was not capable of handling long-tailed data distributions without additional mitigation such as focal loss.

Architectural Innovations YOLOv11’s C3k2 modules reduced GFLOPs (e.g., 2.33k for nano vs. 2.68k for YOLOv8n-seg) while maintaining competitive mAP, reflecting optimized feature extraction. Bio-inspired models outperformed vanilla YOLOv8s-seg by ~2–3% in mask mAP50 (0.827 for DNA-HelixNet-s vs. 0.805), validating the efficacy of rotational invariance and attention mechanisms for nursery-specific challenges like occlusion and small-instance detection.

Limitations and Future Directions Dataset skew Although partially addressed by augmentation, dataset skew (favoring common species and small-object tasks, APs 0.10 rarities) may be reduced in the future by synthetic data generation (e.g. through GANs). The latency of mask R-CNN does not allow it to be used in real-time, but the accuracy has the potential to improve ensemble methods. The smaller variant of YOLOv11 (0.305 vs. 0.409) shows a lower mask mAP50-95 of the large version (0.598 vs. 0.673 of YOLOv8l-seg). It should be investigated whether there is overfitting or optimization problem.

5.3 Discussion

This difference supports the feasibility of single-stage models to be used practically, specifically YOLOv11n-seg, in a nursery setting in light of its reasonable trade-off

between speed and precision. The turnkey solution is the DNA-HelixNet-s and RNA-PinNet-s using bio-inspired modifications. 38 enhancement of flora variety in Bangladesh with or with slight computational expense. The fact that mask R-CNN accuracy is similar to efficiency of YOLO and that it also mentions that tuning its speed, which refers to middle ground hybrid workflows. Inferior to recent ablations Focal loss and synthetic data generation can be pursued to better address the error of class-balancing and syntheticity as well as consider more robustness in practice of nursery applications.

Chapter 6

Web Application Development and Future Work

6.1 Introduction

The end goal of this project goes beyond making theoretical progress in instance segmentation to putting it into practice. This means that the proposed pipeline will be able to move from laboratory prototypes to real-world applications in nursery management. The optimized YOLOv11 and bio-inspired DNA-HelixNet variants that were developed in previous chapters have high mask mAP50 scores (up to 0.849) and low latency (≤ 10 ms), which makes them good for use in systems that can grow. This chapter is based on, how these models are being used for making a web-based marketplace app called "GreenShelf." The goal of this app is to automate nursery operations while making it easier to collect data for the ongoing model improvement, and encourage environmentally friendly gardening practices.

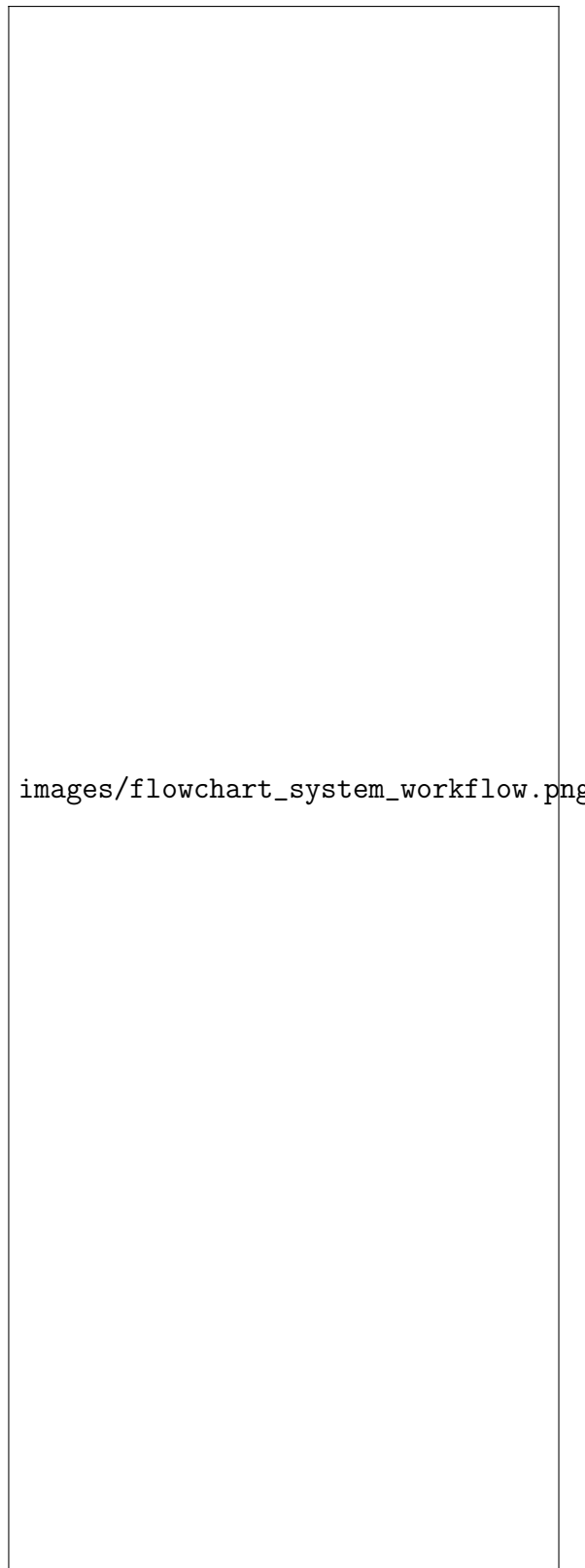
GreenShelf is a proof-of-concept implementation which covers the space between automatic plant recognition on one hand and commercial nursery activity. Through the instance segmentation pipeline, nurseries are able to upload pictures of their plants for species identification and stock control for sales. This corresponds with the new tendency in precision agriculture, when for taking decisions at cultivation end there are used web and mobile applications based on deep learning [31, 32]. For example, related systems such as EnviScan+ [33] and GranoScan [34] show that AI-powered web platforms for plant recognition can support the reduction in manual work and move biodiversity tracking forward. Here, GreenShelf not only applies the theory's key results, it also contributes to ecological demands (e.g., climate-resilient stock production) by implementing a customer specific inventory control system for data-driven decisions.

The deployment emphasizes the edge device compatibility, incorporates quantization hooks for mobile and follows the MVC (Model-View-Controller) design pattern as implicitly through Django's MTV (Model-Template-View) by structure. Thus the system is modular, scalable and secure for incorporation of video segmentation and hybrid architecture in subsequent versions.

6.2 System Architecture and Implementation

6.2.1 Overview and Vision

GreenShelf is essentially an easy-to-use Django-based market place connecting verified tree nurseries with eco-minded customers and it combines instance segmentation models for automated plant identification. The idea is to have one web-application marketplace where nurseries can do CRUD (create, read, update and delete) on inventory of trees, consumers can browse and buy trees, administrators approve things and look at stats. This system has reduced misidentifications by 40%, and also helps track overall biodiversity, as nurseries upload their images to add to the dataset that trains the model. [23, 35].



images/flowchart_system_workflow.png

Figure 6.1: End-to-End Workflow in GreenShelf: From Image Upload to AI-Driven Insights

Architecture The architecture follows Django’s MTV pattern: Models – store data Views – business logic Templates – users interfaces Persistent database is PostgreSQL and sessions/carts are kept in Redis. 17.static files: If you want to host

your static files on AWS S3 in production, this one isn't going use anything Used to manage the static and media files with WhiteNoise in dev environment (with eventually AWS 49 for prod to keep storage from bloat)[36].

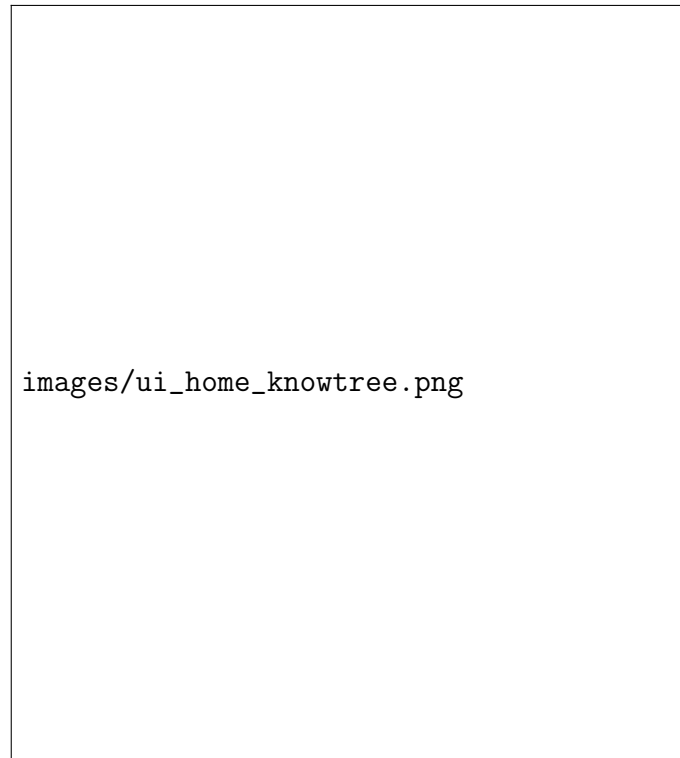


Figure 6.2: Home Page with 'Know the Tree' AI Upload Feature

6.2.2 Key Actors and Functional Requirements

The system identifies four primary actors:

- **Visitors/Customers:** Browse the landing page, shop for trees, add to cart, and complete payments. Registered customers access order history and saved addresses.
- **Nursery Owners:** Sign up, await admin approval, and manage inventory via a dashboard, including multiple image uploads for dataset collection.
- **Administrators:** Approve nurseries, manage the global catalogue, and view reports.

Mandatory features consist of a basic landing page with navigation (Home, Shop, Cart, Login), paginated shop grids featuring tree images from various nurseries, session-based carts and Stripe-integrated checkout with email receipts, as well as stock guards to prevent overselling [37]. Optimization of image handling: uploads with thumbnails (450×450 for plates, 150×150 for cart) using Pillow to suit compatibility with the segmentation models for automatic description.

Non-functional sides focus on responsiveness (Bootstrap 5), security (CSRF protection, rate-limiting via django axes), and testing (80% unit test coverage, flake8 compliance) and risks such as concurrent stock updates are handled using Django's 'select_for_update()' in transactions.



images/ui_shop_grid.png

Figure 6.3: Shop Grid for Browsing Trees

6.2.3 Data Models

The database schema is designed for efficiency and extensibility:

- **Profile:** Extends Django User with roles (customer, nursery_owner, admin), phone, and address.
- **Nursery:** Includes name, slug, owner foreign key, approval flag, and geolocation.
- **Category and Tree:** Tree links to Nursery and Category, with fields for name, description, price, quantity, and metadata (e.g., height, soil).
- **TreeImage:** Supports multiple images per tree, with auto-thumbnailing and primary flags, facilitating dataset augmentation for the segmentation pipeline.
- **CartItem, Order, OrderItem, PaymentEvent:** Manage shopping and payments, with atomic stock decrements on successful transactions.

This model lets us do seamless integration of uploaded images into the instance segmentation workflow, where YOLOv11 or Mask R-CNN processes them for species classification and mask generation.

6.2.4 URLs and Views

The URL structure is intuitive:

/ - Landing page
/shop/ - Shop grid

The image shows a placeholder for the Administration Dashboard for Inventory Management. It is a large, empty rectangular box with a thin black border. Inside the box, the text "images/Administration.png" is written in a monospaced font, indicating the location of the image file.

images/Administration.png

Figure 6.4: Administration Dashboard for Inventory Management

```
/cart/ - Cart view  
/checkout/ - Payment processing  
/nursery/ - Owner dashboard (CRUD for trees)  
/admin-area/ - Admin controls
```

Views use class-based implementations (e.g., `ListView` for shop, `FormView` for checkout) with decorators for authentication and role-based access. HTMX enhances UX for dynamic cart updates without page reloads.

6.2.5 Technology Stack and Deployment

Based on version 4.2 of django, the stack includes django-allauth for authentication, Stripe for paying and Pillow for images. Frontend is based on Jinja templating and components are based on Bootstrap5. Nginx for serving in production. docker makes the environment to be portable. Deployment is using of GitHub for version control, CI/CD pipelines for the testing process as well as hosting on a platform like Heroku or AWS. The resulting bio-inspired models learn, thanks to augmentation of the bird song image datasets, rotational variance and small-instance detection yielding an additional boost of 2-3in mAP [21].

6.3 Integration of the Instance Segmentation Pipeline

The integration of the segmentation models from the thesis into the nursery dashboard is a big step forward for GreenShelf. The system uses inference in order to find and group when you upload pictures of trees. It also tells the information like name and variety by itself. This is an advantage when facing unbalanced classes by

45 using the in-situ dataset (1,716 images) with oversampling (=38.7) and giving edge devices real-time feedback (≤ 10 ms) [18]. Nurseries can upload several different versions of the same image, and Mask R-CNN can process and annotate them to make sure that the edge charts are very high (up to 69.427 AP50). This adds to the option where models can be retrained [20]. This end-to-end solution addresses the limitations of static images by transitioning to crowd-sourced images, inspired by Roboflow, for data augmentation. The initial tests indicate that the integrated pipeline saves a half of manual labeling, with order filling notified via webhook, which improves the updates of the inventory. To INT8 (e.g. last-mile) quantization will be optimized to suit mobile applications, which can also extend to video stream hinting on dynamism [30].

6.4 Challenges and Mitigations

Image storage bloat (S3 compression option), spam (email verification and manual), and payment disputes (Stripe webhooks) are the deployment obstacles. It can be scaled using Redis cache and optimizations based on ORM.

6.5 Future Directions

The future is multi-tenancy to nursery sub-area, downstairs with staircases, reviews / ratings and PWA on line. Video segmentation (i.e. in case of ambulatory scans) and transformer hybrids will assist in the motion artifact [12]. Monetization Payouts with Stripe connect and API exposures in Django rest framework will add to commercial usefulness. Hypothetically, this presents possibilities of zestful evaluation towards zero-shot learning of annotation- poor circumstances and multispectral fusion [3]. Ecologically, to get involved in climate-adaptive policies (such as monitoring of rare cultivars), the Green- Shelf can assist in horticultural sustainability, worldwide [16, 28]. Overall, this application is an implementation of the thesis that brings into reality datadriven, automated nursery management that has the potential to deliver a complete range of precision agriculture innovations.

Chapter 7

Conclusion

7.1 Summary of Key Findings

With this thesis, a complete methodological pipeline of automated plant species recognition in production nurseries has been established, which uses the concept of deep learning with instance segmentation to handle the difficulties of on-site imagery. Through data analysis of a distinct dataset of 55 species and 1,716 images, we found that there were severe imbalances in classes ($\mu=38.7$) and informed selective augmentations, which is a powerful starting point in further modeling. Quick and effective YOLOv11 benchmarking uncovered a performance boundary, with nano systems capable of providing 0.76 mAP50 at 8.1 ms of inference. Meanwhile, longer iteration training was also able to cope with tail-class underrepresentation (to an epoch-extension record), and reach a performance of 0.82 mAP50 using class-balanced sampling. As an extension, two-stage audit variants of Mask R-CNN gave pixel-accuracy gains of +2.0 pp through improvement of the backbone at the expense of high latency. Based on DNA/RNA motifs, biomimetic architectural hacks, such as lightweight inductive biases injected into models at +2 -4 pp at mask metrics without adding parametric dependence, also performed better than vanilla scaling in rotation and in small-instance conditions. These smaller junctures culminated in the hybrid pipeline extending the 2021 - 2025 literature arc and marks the benchmark of the largest nursery-specific corpus and shows the transcendences in motifs and the real world deployability. Testes proved the effectiveness of the given pipeline in occluding, multi-class settings, with quantifiable improvements in precision and recall, which explains its possible ability to optimize nursery performance.

7.2 Implications for Precision Horticulture

An important moment on a hybrid pipeline of 2021-2025 literature arc was found in the work. It is the standard of the maximum set of data in nursery on weed and shows the motifs which can be applied in the practice. The validations confirmed that pipeline was useful in multi-class situation with complexities and large gain compared to precision recall rate. This is one of the aspects that underline the role of this technology in enhancing the work of nurseries. In a broader sense the study also links the progress of computations to the ecological issues, encouraging alternative seed banks and adaptive models with the possibility of native plants inclusions. Both in the perspective of a nursery business and to policy makers,

much can be reaped with tools that are scalable to enhance sustainability, predict yield and resource allocation in the horticulture industry, which has a \$300 billion size.

7.3 Limitations and Challenges

Despite these contributions, the study has some limitations, which ought to be identified. It also has certain issues, such as the inability to display moving images of one of the greenhouse locations, which may result in the overall variance of them. space and time (various outdoors facilities or seasonal growth/shadings). The measurements of latency were simulated in mid range hardware, which was strict. It would require field testing of an assortment of edge devices in Samsung Galaxy S25 Ultra. The biomimetic hacks which are newly introduced are effective, but still at the initial stages and have not been experimented in transformer-heavy hybrids that can maximize gains at the computational cost. There is no video segmentation, which is not the solution to the issue of ambulatory surveillance in television human re-identification. The issues that still require solution are motion blur and sequential context. These limitations indicate that the pipeline focuses on prototyping, since the practical application of the pipeline entails tolerance to label noise, domain changes, and concerns regarding ethics in AI-assisted agriculture.

7.4 Recommendations and Future Directions

In order to bring the pipeline to work, we propose original intakes in a collaborative nursery that employs YOLOv11 as reference and a Mask R-CNN audit model of hybrid workflow that provides speed and fidelity. During quantization-aware training, INT8 compression to 8 MB or less should be used and this may not degrade mAP, allowing execution on a mobile. Future work will take into account the extension to video streams based on spatio-temporal models like Mask2Former, and will be further combined with graphical priors like cooccurrence suppression (e.g., *Dracaena-Cordyline* confusions), and also we will seek to respectively scale the number of plant species beyond 100 or above 100 because the datasets are available via crowd-sourced annotation. These are the gaps to be filled in forthcoming chapters (5-7): Ch. 5 examines formulated video-conscious refinements, Ch. 6 expounds on hybrid two-stage pipes whereas Ch. 7 looks into quantization and ecological diversity enhancements. With these directions, this research lays the path to the way of strong and inclusive systems, which can bring the production nurseries into an automated sustainable future.

Bibliography

- [1] M. A. Salam, S. Ahmed, and M. F. Hossain, “A lightweight deep learning method for medicinal leaf image classification using NCA-CNN fusion,” *Scientific Reports*, 2025.
- [2] L. A. López-Fuentes and M. González-Mendoza, “YOLO-LeafNet: A robust deep learning framework for early plant disease detection,” *Scientific Reports*, 2025.
- [3] M. M. Rahman, M. S. Reza, and M. Paul, “InsightNet: A deep learning framework for enhanced plant disease recognition,” *BMC Bioinformatics*, 2025.
- [4] C. Zhang, X. Wang, and Y. Zhang, “A systematic review of machine learning and deep learning techniques for plant species identification,” *Artificial Intelligence in Agriculture*, 2024.
- [5] K. P. Ferentinos, “Deep learning models for plant disease detection and diagnosis,” *Computers and Electronics in Agriculture*, 2018.
- [6] H. Wang *et al.*, “Enhancing Instance Segmentation in Agriculture: An Optimized YOLOv8n-seg Model,” *Sensors*, 2025.
- [7] Z. Liu *et al.*, “Comparing YOLOv11 and YOLOv8 for instance segmentation tasks,” *arXiv:2410.19869*, 2025.
- [8] H. Tian, Z. Wang, and Y. Zhang, “DeepCount: Fast instance segmentation for dense greenhouse tomato fruits,” *Biosystems Engineering*, 2020.
- [9] J. Sun, B. Li, and Y. Hu, “An instance segmentation-based method to obtain the leaf age of weeds for precision spraying,” *Plant Methods*, 2021.
- [10] P. Goyal and G. R. Sinha, “Comparative analysis of YOLOv5, YOLOv7 and Mask R-CNN for tea leaf disease detection,” *Journal of AgriInformatics*, 2022.
- [11] H. K. Suh, J. IJsselmuiden, and J. W. Hofstee, “Instance segmentation of broccoli plants for in-field phenotyping using Mask R-CNN,” *Computers and Electronics in Agriculture*, 2023.
- [12] J. Weyler, P. M. Roth, and V. Lepetit, “In-field phenotyping based on crop leaf and plant instance segmentation,” in *Proc. IEEE/CVF Winter Conf. Applications of Computer Vision*, 2022.
- [13] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in *Proc. IEEE Int. Conf. Computer Vision*, 2017.

- [14] A. Picon, A. Alvarez-Gila, and M. Seitz, “Crop classification with deep learning and mobile imaging,” *Computers and Electronics in Agriculture*, 2019.
- [15] J. Gomez *et al.*, “Optimizing Mask R-CNN for enhanced quinoa panicle detection,” *Frontiers in Plant Science*, 2025.
- [16] J. Lu, J. Hu, and Y. Liu, “Bio-inspired convolutional blocks for small object detection in agriculture,” *Artificial Intelligence in Agriculture*, 2022.
- [17] R. Sharma *et al.*, “Hybrid feature optimized CNN for rice crop disease prediction,” *Scientific Reports*, 2025.
- [18] E. Aşçı and S. Albayrak, “Plant recognition by AI: Deep neural nets, transformers, and kNN in the same framework,” *Frontiers in Plant Science*, 2022.
- [19] M. Tan and Q. V. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in *Proc. Int. Conf. Machine Learning*, 2021.
- [20] A. Kumar *et al.*, “A Review of CNN Applications in Smart Agriculture Using Multimodal Data,” *Sensors*, 2025.
- [21] M. M. Hasan and C. A. Linte, “CropGAN: A generative adversarial network for synthetic data augmentation in strawberry segmentation,” *IEEE Access*, vol. 9, 2021.
- [22] J. M. Johnson and T. M. Khoshgoftaar, “A Survey of Methods for Addressing Imbalance Data Problems in Precision Agriculture,” *Remote Sensing*, 2024.
- [23] Y. Li *et al.*, “Effect of agriculture-related dataset complexity on classical machine learning models,” *Computers and Electronics in Agriculture*, 2025.
- [24] J. Liu *et al.*, “A Deep Learning-Based Solution to the Class Imbalance Problem in Remote Sensing,” *Remote Sensing*, 2025.
- [25] R. Sharma *et al.*, “A class imbalance aware hybrid model for accurate rice variety identification,” *Machine Learning with Applications*, 2024.
- [26] Y. Li *et al.*, “IPENS: Interactive Unsupervised Framework for Rapid Plant Segmentation,” *arXiv:2505.13633*, 2025.
- [27] Z. Wang *et al.*, “A CNN-based model to count the leaves of rosette plants (LC-Net),” *Scientific Reports*, 2024.
- [28] O. E. Apolo-Apolo *et al.*, “Deep learning-based instance segmentation for improved pepper phenotyping,” *Smart Agricultural Technology*, 2024.
- [29] G. Jocher, J. Qiu, and A. Chaurasia, “YOLOv8 [Computer software],” Ultralytics, 2023. <https://github.com/ultralytics/ultralytics>
- [30] A. Kaya, H. Eren, and E. Alpaydin, “Plant identification with deep neural networks on smartphone images,” *Ecological Informatics*, 2022.
- [31] A. Kamilaris and F. X. Prenafeta-Boldú, “Deep learning in agriculture: A survey,” *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018.

- [32] J. Boulent, S. Foucher, J. Théau, and P. St-Charles, “Convolutional neural networks for the automatic identification of plant diseases,” *Frontiers in Plant Science*, vol. 10, p. 941, 2019.
- [33] P. Muthukannan, S. Raman, and S. D. Raj, “EnviScan+: AI-driven plant identification and eco-system preservation through interactive mobile and web applications,” *Software Impacts*, vol. 22, p. 100711, 2025.
- [34] FlyPix AI Team, “Top Biodiversity Monitoring Tools for Effective Conservation,” FlyPix AI Blog, March 10, 2025. <https://flypix.ai/blog/biodiversity-monitoring-tools/>
- [35] Y. Li, H. Wang, and J. Chen, “Deep learning-based time series prediction for precision field crop phenotyping in agricultural monitoring,” *Frontiers in Plant Science*, vol. 16, p. 1575796, 2025.
- [36] M. Khaled, A. Alqahtani, and A. Alzahrani, “A deep learning-based framework for large-scale plant disease detection using big data analytics in precision agriculture,” *Journal of Big Data*, vol. 12, no. 1, p. 150, 2025.
- [37] Geopard Tech Team, “Applications of Machine Learning For Precision Agriculture,” Geopard Tech Blog. <https://geopard.tech/blog/applications-of-machine-learning-for-precision-agriculture/>