Robust Fruit Segmentation for Automated Harvesting Robots under Adverse Environmental Conditions

Jinyao Zhou
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
jinyaozh@cs.cmu.edu

Yichen Ji
Heinz College
Carnegie Mellon University
Pittsburgh, PA 15213
yichenj@andrew.cmu.edu

Xiaolei Hu
MCS
Carnegie Mellon University
Pittsburgh, PA 15213
xiaolei2@andrew.cmu.edu

En Zheng
Department of Chemistry
Carnegie Mellon University
Pittsburgh, PA 15213
enzheng@andrew.cmu.edu

Abstract

This project addresses the critical challenge of optimizing the precision of fruit segmentation for the operation of harvesting robots under various environmental conditions. Current deep learning-based fruit segmentation systems demonstrate significant performance degradation when deployed in adverse conditions such as low-light, nighttime, foggy weather, and high-occlusion scenarios. Our proposed framework integrates a hybrid model of state-of-the-art objective detector and segmenter, environment-aware data augmentation using diffusion models to achieve robust performance across challenging environmental settings.

1 Overview and Context

1.1 Motivation

Agricultural automation represents a critical solution to labor shortages and efficiency challenges in modern agriculture. Existing fruit detection systems, while effective under controlled conditions, fail to maintain consistent performance when faced with real-world environmental variations. Current fruit can be classified into single-stage approaches (You Only Look Once (YOLO), Single Shot MultiBox Detector (SSD)) and two-stage methods (Faster R-CNN, Mask R-CNN), but both categories suffer from reduced accuracy under adverse conditions [1]. This limitation significantly restricts the practical deployment of autonomous harvesting systems in commercial agricultural operations.

1.2 Objective

The primary objective is to develop a robust fruit detection and segmentation system specifically optimized for peppers under adverse environmental conditions. Key goals include: (i) achieving stable segmentation performance (mAP diff ≤ 0.05) under varying illumination conditions including daylight, low light, and nighttime scenarios, (ii) implementing adaptive preprocessing that automatically adjusts to environmental conditions, (iii) developing a comprehensive training framework using both real outdoor data and synthetically generated adverse condition data.

2 Related Work and Background

2.1 Literature Review

Recent approaches for enhancing accuracy object detection and segmentation at adverse environments can be classified into three directions: data augmentation, object detection and segmentation, and preprocessing.

2.1.1 Data Augmentation

High-quality dataset generation remains a major challenge in computer vision, especially in specialized domains such as agricultural object detection. Traditional data collection and annotation pipelines are often time-consuming, labor-intensive, and limited in scope, which has motivated growing interest in synthetic dataset generation. To address these challenges, researchers have increasingly turned to advanced computational approaches. Among them, data augmentation has proven particularly effective: it allows artificial expansion of datasets by generating diverse, high-quality image variations from existing samples, thus enriching the data set without the need for additional physical data collection[2].

Conventional data augmentation techniques, such as geometric transformations (e.g. rotation, scaling, translation, flipping), and color adjustments[3], are effective in enhancing model robustness. However, they often produce highly correlated samples, which limits their ability to capture complex variations or invariant features inherent in training data. In practice, classical augmentation libraries such as the Python Imaging Library (PIL) remain widely used due to their stability, controllability, and preservation of image structure[4]. PIL enables non-destructive transformations—including brightness, contrast, and color adjustments—that can simulate environmental variations such as weak illumination or mild atmospheric effects while fully preserving object geometry. This property makes PIL-based augmentation especially valuable in tasks like agricultural detection, where spatial fidelity is essential for maintaining valid annotation labels. Subsequent advances in deep-generative modeling introduce generative adversarial networks (GANs) for data synthesis [5][6]. However, GAN-based approaches still face several limitations, such as training instability and model collapse[7].

Recently, probabilistic diffusion models, or simply diffusion models, have gained prominence in the field of image synthesis, offering a powerful new approach to augment image datasets to enhance machine vision systems [8][9][10]. Inspired by non-equilibrium thermodynamics[11], diffusion models progressively corrupt data with noise and then learn to reconstruct the original samples through a reverse Markov process starting from pure noise. Current advances in generative modeling have established diffusion-based approaches as state-of-the-art methods for image synthesis. Diffusion models can generally be categorized into pixel-space diffusion models, such as denoising diffusion probabilistic models (DDPM) [8], and latent-space diffusion models, such as latent diffusion models (LDM) [12].

DDPMs operate directly in pixel space, progressively adding and removing Gaussian noise to generate high-fidelity images, but this process incurs substantial computational costs and slow inference due to the high dimensionality of the data. Subsequent variants, including the improved DDPM and augmented diffusion model (ADM) frameworks, have enhanced sample quality and controllability but remain computationally intensive for large-scale or high-resolution image synthesis. Chen et al. demonstrated the generation of high-quality images for weed recognition using a classifier-guided diffusion model (ADM-G) based on a 2D U-Net architecture, showing that their method consistently outperforms several state-of-the-art GAN in both sample quality and diversity[10][13][14].

In contrast, the LDM performs the diffusion process in a compressed latent space learned by an autoencoder, substantially reducing computational costs while maintaining high visual fidelity. This architecture enables faster image generation and supports flexible conditioning mechanisms, such as text or structural guidance, that have been effectively implemented in models such as Stable Diffusion [15]. Because of their efficiency and controllability, LDMs are particularly well suited for large-scale synthetic dataset generation, making them a promising approach for agricultural imaging tasks that require diverse, high-resolution data with domain-specific variations. Qiu et al. proposed the use of LDMs for the detection of plant diseases by generating images of unseen classes, while Zhao et al. used Stable Diffusion to create extensive data sets of fruit images to improve robotic fruit picking

automation[16][17]. Together, these studies demonstrate the expanding potential of LDMs for diverse applications within the agricultural sector.

2.1.2 Object Detection and Segmentation

With regard to the detection and segmentation model, there are not many innovations focusing on robustness to diverse environmental conditions, but there are some general improvements for the overall detection and segmentation accuracy. Grounding DINO is open-set object detector which can detect arbitrary objects with human inputs such as category names or referring expressions by introducing language to a closed-set detector for open-set concept generalization[18]. It allow successfully recognize and classify objects or concepts it has never seen during its training phase. This can theoretically help detect objects in adverse environment, even if it is not trained with such data. However, when handling a specific detection task - for example, detecting a pepper - its performance is not competitive to classical models such as YOLOv9 fine-tuned with pepper fruit dataset [19].

In addition to the iterations of state-of-the-art models, some researchers tried to fuse existing models together for better performance. For example, Liang et al. proposed a visual detection method for nightly litchi fruits that combined YOLOv3 with U-Net segmentation, achieving promising results under low-light conditions [20]. Paul and Machavaram introduced YOLOvOVOD in nightcapsicum detection, a novel hybrid model that synergistically integrates the high-accuracy detection capabilities of YOLOv9 with the open vocabulary that prompts DINO Grounding [19]. Chen et al. managed to achieve multiscale feature extraction and fusion for citrus picking by adding the attention mechanism, the Convolutional Block Attention Module (CBAM), to the YOLOv7 network [21].

Two-stage detectors have also been widely adopted for fruit detection and instance segmentation. By first generating region proposals and then performing RoI-level refinement, this proposal–refinement framework helps improving accuracy and robustness in cluttered, occluded scenes. These advantages are well established in the two-stage literature and are further reinforced under adverse conditions via architectural and training refinements [22]. In agricultural settings, Faster R-CNN and Mask R-CNN remain prevalent in fruit detection research. For example, ROLS adapts a Mask R-CNN–style head to delineate grape instances and reports P=0.9662, R=0.9881, F1=0.977 on field imagery [23]. Similarly, AppleGrowthVision shows that enriching training diversity substantially strengthens a Faster R-CNN baseline for apple detection (e.g. +0.31 F1), achieving AP=0.743, AR=0.640, and mAP @ 0.5: 0.95=0.305 [24]. Collectively, these findings corroborate the suitability of two-stage architectures for precise, robust fruit localization and segmentation.

2.1.3 Enhanced Image and Feature Extractor

Image signal processors (ISPs) can significantly impact the performance of downstream computer vision tasks while converting raw sensor signals into digital images. In terms of the segmentation task in adverse environments, researchers introduced various ISP pipeline and parameter tuning methodologies. Wang et al. proposed AdptiveISP, a task-driven and scene-adaptive ISP that uses deep reinforcement learning to automatically generate an optimal ISP pipeline and associated parameters to maximize detection performance [25]. Apart from adaptive pre-processing methodologies, there are also breakthroughs aiming at a specific environmental condition, such as low-light scenarios. Semantic-Guided Zero-Shot Learning (SGZ) is one such method, utilizing a recurrent network for image enhancement and an unsupervised semantic segmentation network for preserving semantic information under low-light conditions . This technique employs depthwise separable convolution to estimate pixel-wise light deficiency and is designed to improve visibility in dark scenes. YOLA, a novel feature extraction module introduced by Hong et al., is designed to learn illumination-invariant feature maps from low light images through a convolutional neural network and is easily integrated into existing object detection frameworks [26]. Integrating these ISP pipelines into the segmenter is promising in helping to improve detection and segmentation precision.

2.2 Background

In night-time greenhouse environments, accurate detection remains challenging due to low illuminance, complex occlusions from foliage and branches, and the strong color similarity between capsicum fruit and background vegetation. Paul et al. specifically address this setting by constructing a dedicated night-time capsicum dataset and benchmarking a series of modern YOLO detectors under

these conditions [19]. Their evaluation is carried out using precision, recall, F1 score and mean Average Precision at IoU 0.5 (mAP@0.5), metrics that directly reflect the reliability of detections required for downstream robotic manipulation. The study further compares these single-shot detectors with a zero-shot, open-vocabulary model (Grounding DINO), highlighting the trade-off between task-specific accuracy and prompt-based flexibility. As our work also targets robust capsicum detection in low-light greenhouse scenarios and reports performance in terms of precision, recall, F1 and mAP@0.5, this paper provides a natural and well-aligned reference point for both our problem formulation and evaluation protocol.

3 Methodology

In the project, we compared two pipelines that utilizes the state-of-art methodologies for object detection task in adverse environments. The first pipeline is based on an illumination-invariant feature extractor, which is pretrained with professional low-light datasets like Exdark. The second pipeline follows an image pre-processing strategy: we insert a dedicated pre-processing module (e.g., SGZ or IA-YOLO enhancement) before the prediction, which transforms degraded images into a form closer to normal illumination while keeping the detection model unchanged. For both pipeline, we used clear, normal-light pepper images for training, and a combination of augmented, synthesized and real-world low-light pepper images for test and performance evaluation. By evaluating these two pipelines on the same benchmark, we systematically compare the effectiveness and limitations of "feature-level robustness modeling" versus "input-level pre-processing" for object detection in adverse conditions.

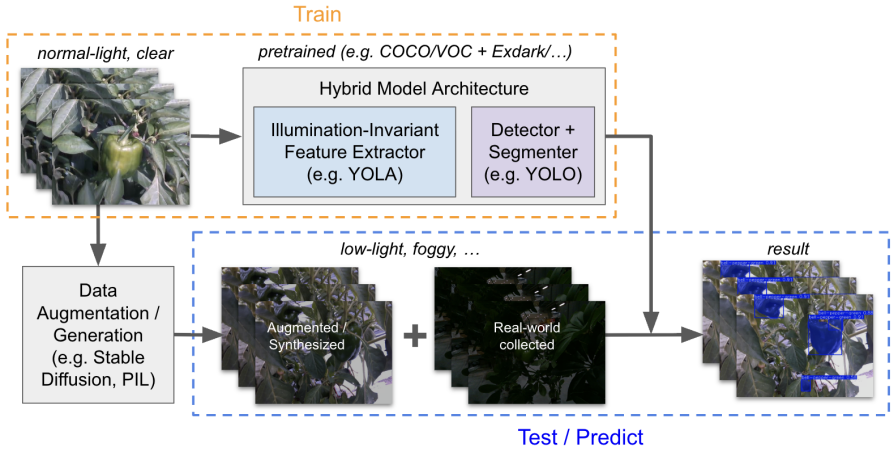

Figure 1: Pipeline with Illumination-invariant Feature Extractor

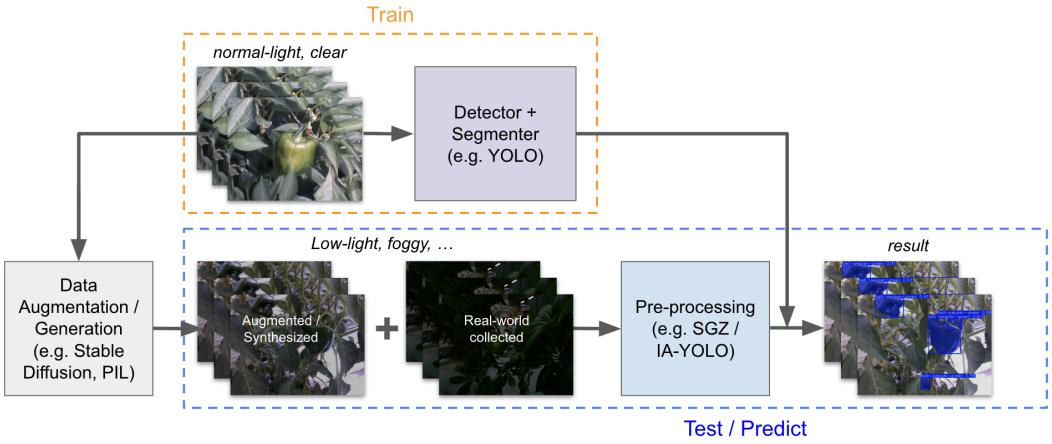

Figure 2: Pipeline with Image Pre-processing

3.1 Model Description

3.1.1 Data Augmentation

In this project, a hybrid image augmentation framework was developed by integrating pixel-level enhancement using the PIL with semantic-level image generation using the Stable Diffusion v1.5 Img2Img pipeline [12]. The workflow begins with preliminary adjustments in PIL, where brightness, contrast, and saturation are modified to simulate various physical lighting conditions, such as strong sunlight or overcast skies. For more complex scenarios, including rain or fog, PIL is further employed to overlay synthetic raindrops or fog gradients, producing visually consistent inputs for the subsequent diffusion-based refinement stage. The processed images are then enhanced using the Stable Diffusion v1.5 Img2Img model, implemented via the Hugging Face repository runwayml/stable-diffusion-v1-5, which corresponds to the official v1-5-pruned-emaonly.ckpt checkpoint released by CompVis and RunwayML. Custom-designed text prompts were constructed to represent diverse environmental conditions, including variations in lighting and weather such as strong noon sunlight, cloudy soft light, rainfall, and fog. Three key hyperparameters were systematically varied to assess their influence on image generation outcomes: the strength parameter (tested between 0.4 and 0.6) controlling the degree of transformation from the input image; the guidance scale (ranging from 7 to 9) determining the extent of prompt adherence; and the number of inference steps (set between 30 and 40) balancing image quality against computational cost. By systematically varying these parameters, the experiment aimed to analyze how different configurations influence the trade-off between image realism and diversity in the generated results. This hybrid approach preserves both pixel-level fidelity and semantic richness, producing visually coherent augmentations that enhance the robustness and generalization capability of downstream computer vision models.

3.1.2 Object Detection and Segmentation

To perform object detection and segmentation tasks, we selected YOLOv9 and Detectron2 as our baseline models.

You Only Look Once (YOLO) series of models is one of the most influential frameworks for real-time object detection and segmentation due to its unified, single-stage architecture that balances speed and precision. It is widely adopted in agricultural applications, including crop recognition, fruit counting, weed discrimination, and detection of plant disease [1]. YOLOv9 is the latest version of this series released in 2024, which shows superior detection accuracy, especially on small and complex objects [27]. Our baseline model is fine-tuned with open-sourced pepper dataset based on pretrained YOLOv9c-seg weights. The preliminary fine-tuned epoch is 10, which will be extended in the later stage of the project. The images are resized to 320x320 before being fed into the pre-trained model with batch size of 16. Other hyperparameters can be found in the code to be released in the github repository after the final report.

Regarding Dectectron2, we use the standard `mask_rcnn_R_50_FPN_3x` configuration as the baseline. In this setting, Detectron2 builds a two-stage Mask R-CNN under the GeneralizedRCNN meta-architecture. A ResNet-50 backbone extracts features that a Feature Pyramid Network (FPN) merges into multi-scale maps. A Region Proposal Network (RPN) then generates candidate regions, which are pooled via RoIAlign. In the second stage, a FastRCNNConvFCHead performs classification and box regression on RoI features, while a parallel MaskRCNNConvUpsampleHead predicts per-instance masks. The model thus outputs class labels, bounding boxes and instance masks. Finally the "3×" schedule indicates the model is trained for 3 times of the standard duration, allowing more iterations to achieve higher accuracy and robustness. With this setting, we fine-tune this baseline on an open-source pepper dataset using the above configuration, initializing from the COCO-pretrained `mask_rcnn_R_50_FPN_3x` weights provided by the Detectron2 model zoo.

3.1.3 Illumination-Invariant Feature Extraction

As introduced in YOLA, an Illumination-Invariant Module (IIM) is inserted as an explicit illumination-invariant feature extractor in front of an off-the-shelf detector (e.g., YOLO) [26]. Given an input RGB image $I \in \mathbb{R}^{H \times W \times 3}$, the processing pipeline of IIM is as follows.

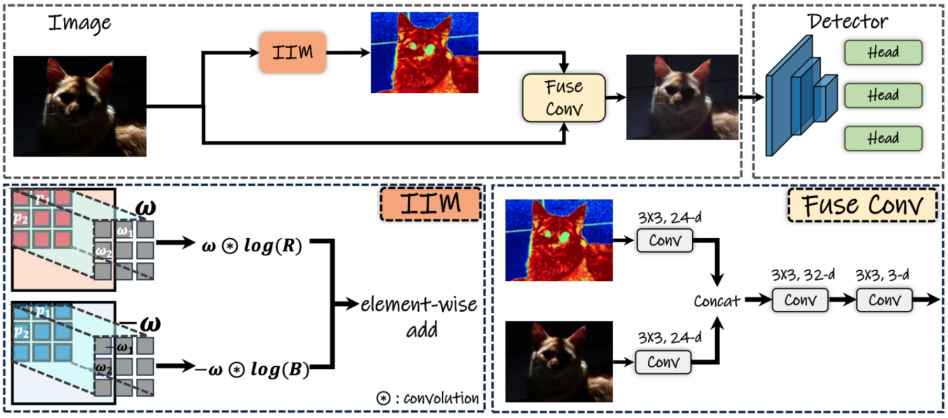

Figure 3: YOLA Pipeline [26]

First, the logarithm is applied to each color channel to transform the multiplicative terms in the Lambertian reflection model into additive ones:

$$I = [R, G, B] \quad \longrightarrow \quad [\log(R), \log(G), \log(B)].$$

Then, the IIM contains a set of shared convolution kernels $\{\mathcal{W}_i\}_{i=1}^{n}$, each of spatial size $k \times k$, constrained to have zero mean, i.e., $\overline{\mathcal{W}_i} = 0$. For each kernel $\mathcal{W}_i$, we construct "differential convolutions" over the three channel pairs $(R, B), (R, G), (G, B)$:

$$f_{\mathcal{W}_i}(I) = \begin{bmatrix} \mathcal{W}_i \circledast \log(R) - \mathcal{W}_i \circledast \log(B) \\ \mathcal{W}_i \circledast \log(R) - \mathcal{W}_i \circledast \log(G) \\ \mathcal{W}_i \circledast \log(G) - \mathcal{W}_i \circledast \log(B) \end{bmatrix}$$

where $\circledast$ denotes 2D convolution. This construction performs weighted differences within each channel and across channels, which cancels out the local illumination term and the dependence on surface normal and light direction under the Lambertian assumption.

The outputs from all kernels are concatenated along the channel dimension and passed through a 3×3 fusion convolution layer to produce a 3-channel illumination-invariant feature map aligned with the original image. Finally, we add this feature map back to the input image in an element-wise manner and feed the result into the detection backbone.

When $\mathcal{W}_i$ is fixed to adjacent difference filters with entries restricted to ± 1, the IIM degenerates to *IIM-Edge*, which effectively acts as a multi-scale edge extractor. With learnable zero-mean kernels, the module can instead discover richer, task-relevant patterns that remain approximately illumination-invariant.

3.1.4 Adaptive Image Enhancement

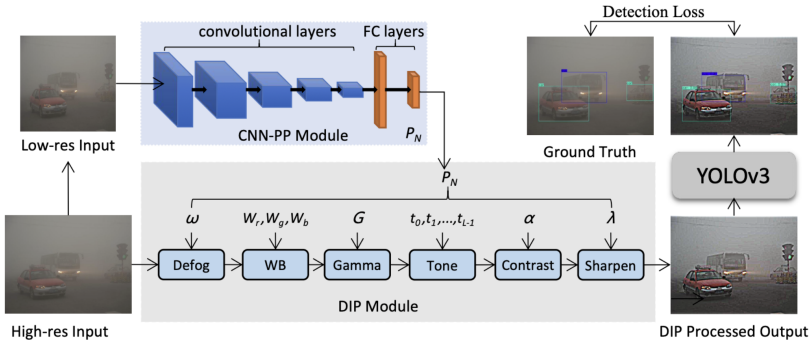

Figure 4: IA-YOLO Pipeline [28]

As shown in Fig. 4, Image-Adaptive YOLO (IA-YOLO) augments a standard YOLOv3 detector with a small CNN-based parameter predictor (CNN-PP) that analyzes a low-resolution copy of each input

image and predicts the hyperparameters of a differentiable image-processing (DIP) module made of white-balance, gamma, contrast, tone filters. The high-resolution image is then passed through this DIP module and the enhanced result is fed into YOLOv3.[28] The DIP module applies four learnable pixel-wise filters: white balance (WB), gamma, contrast, and tone to each input image. Their mapping functions are summarized in Table 1. WB scales the RGB channels by learnable factors W_r, W_g, W_b. Gamma filter raises each pixel to a learnable exponent G. The contrast filter linearly blends the original pixel with an enhanced luminance version using a weight α. The tone filter uses a piecewise-linear tone curve parameterized by $\{t_0, \ldots, t_{L-1}\}$. These differentiable mappings allow the detector to jointly learn optimal color, contrast, and tone adjustments for robust detection under adverse conditions. Then, the detector, CNN-PP, and DIP modules are trained jointly, using only the detection loss. In our pepper-detection project, we adapt this idea by tried to place a similar CNN-PP + DIP front-end before our fruit detector and training on both clean greenhouse images and synthetic low-light / foggy versions of the same scenes. Therefore, the model could automatically choose appropriate enhancement parameters for each frame and thereby improve the robustness of pepper detection under adverse illumination and visibility.

Table 1: The mapping function of pixel-wise filter [28]

Filter	**Parameters**	**Mapping Function**
WB	W_r, W_g, W_b: factors	$P_o = (W_r r_i,\ W_g g_i,\ W_b b_i)$
Gamma	G: gamma value	$P_o = P_i^G$
Contrast	α: contrast value	$P_o = P_i^G$
Tone	t_i: tone params	$P_o = \left(L_{t_r}(r_i),\ L_{t_g}(g_i),\ L_{t_b}(b_i)\right)$

3.1.5 Low-light Oriented Image Enhancement

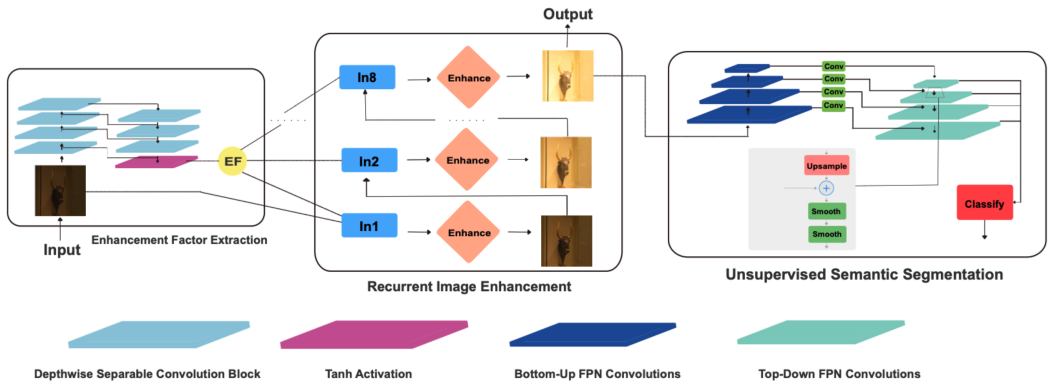

Figure 5: SGZ Model Architecture [29]

The Semantic-Guided Zero-Shot Learning (SGZ) method is a three-stage network designed for robust low-light image/video enhancement that operates without requiring paired images, unpaired datasets, or segmentation annotations.[29] The process begins with the Enhancement Factor Extraction Network (EFE), a lightweight, adaptive fully convolutional network that estimates the pixel-wise light deficiency and records it as an enhancement factor (xr). Next, the Recurrent Image Enhancement Network (RIE) progressively lightens the image, using the previous stage's output and the enhancement factor as its inputs. Finally, the Unsupervised Semantic Segmentation Network (USS) is introduced to preserve semantic information during the intensive enhancement. During training, the RIE and USS layers are frozen, and the USS calculates the semantic loss (Lsem) which is combined with four other non-reference loss functions to form the total loss, which in turn updates only the parameters of the EFE network.

3.2 Dataset

The detection and segmentation model was trained on an open-source bell pepper dataset released through Roboflow [30]. The dataset contains 2,127 images that capture several basic environmental variations, including indoor and outdoor normal lighting, occlusion, and mild blur. However, it lacks

sufficient samples from adverse environmental conditions. Our goal is to evaluate the robustness of the model under such challenging scenarios, for which it was not explicitly trained.

To enable systematic robustness evaluation under conditions such as low light and fog, additional images were generated and incorporated into the testing and prediction pipeline. The final dataset consists of both synthetic images and real outdoor images kindly provided by the Robotic Systems Development (MRSD) Team E — VADER at Carnegie Mellon University, collectively covering a wide range of environmental scenarios.

3.3 Evaluation Metric

Primary evaluation metrics include standard detection measures (mAP@0.5, precision, recall) and environment-specific performance assessments in different lighting and weather conditions. Computational efficiency will be assessed through measurements of inference time, memory usage, and parameter count.

3.4 Loss Function

The Illumination-Invariant Module from YOLA is trianed with Illumination Invariant Loss (II Loss) [26]. To prevent trivial kernels that satisfy the zero-mean constraint but rely on non-local cancellations, the IIM is encouraged to produce similar responses for the same image under different brightness transformations. Given an input image I and a gamma-based brightness transform $\sigma(\cdot)$, another image $\sigma(I)$ with different illumination but identical semantics is obtained. For each kernel output $f_{\mathcal{W}_i}(\cdot)$, a Huber-style consistency loss is defined:

$$L_{\text{II}} = \begin{cases} \frac{1}{2} \left\| f_{\mathcal{W}_i}(I) - f_{\mathcal{W}_i}(\sigma(I)) \right\|_2^2, & \Delta \leq \beta, \\ \left\| f_{\mathcal{W}_i}(I) - f_{\mathcal{W}_i}(\sigma(I)) \right\|_2 - \frac{1}{2}\beta, & \text{otherwise}, \end{cases}$$

where Δ is the absolute feature difference and β is a threshold empirically set to 1. This term enforces stability of the IIM features against illumination changes.

4 Baseline and Extensions

4.1 Baseline Selection and Evaluation

One of the reference baselines for this project is from Paul and Machavaram, as shown in Tables 2 and 3. Since the data set of this work is not open to the public, we trained our own baseline with a fine-tuned YOLOv9 based on an open source data set, which is posted in the baseline section. The computational efficiency metric will only be shown as a reference and will not be our core evaluation criterion.

Table 2: Baseline Evaluation Metrics of Fine-tuned YOLO Models [19]

Models	Size (MB)	Precision	Recall	F1 score	mAP@0.5	Detection Speed (FPS)
Yolov5s	13.9	0.741	0.798	0.769	0.791	75.82
Yolov7s	71.7	0.805	0.866	0.834	0.867	34.37
Yolov8s	21.5	0.845	0.839	0.842	0.925	**45.23**
Yolov9c	**49.1**	**0.898**	**0.864**	**0.881**	**0.947**	38.46

Table 3: Baseline Computational Efficiency Metric [19]

GPU Used	Memory (GB)	Models	Image size	Batch Size	Epochs	Training Time (min)
NVIDIA Tesla T4	16	YOLOv5s	416	32	100	6.84
	16	YOLOv7s	640	16	100	29.86
	16	YOLOv8s	800	16	100	25.68
	16	**YOLOv9c**	640	16	100	**23.34**

4.2 Baseline Reproduction

The prediction result of the fine-tuned YOLOv9 model is shown in Fig. 6. With 10-epoch fine-tuning with bell pepper dataset, the segmentation precision can reach 0.82 and mean average precision can hit 0.85 at 0.5 IoU. The fine-tuned Detectron2 model trained for 10 epochs shows mAP@0.5 = 0.8429 for bounding boxes and 0.8525 for segmentation.

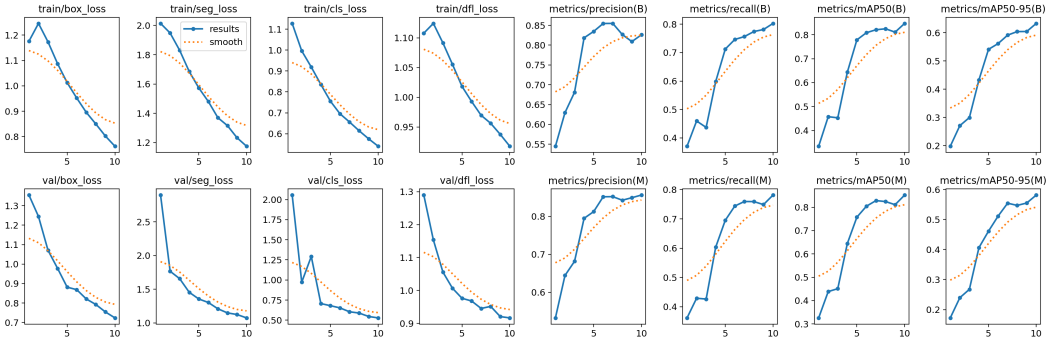

Figure 6: Loss and Evaluation Metrics of Fine-tuned YOLOv9 Baseline

Table 4: Baseline Evaluation Metrics (B: BoundingBox, M: Mask)

Models	Precision (B/M)	Recall (B/M)	mAP@0.5 (B/M)	F1 score
YOLOv9c-seg	0.782/0.856	0.809/0.781	0.827/0.852	0.82 @0.455
Detectron2			0.843/0.853	

Both YOLOv9c-seg and Detectron2 achieve very similar baseline performance on our dataset in terms of mAP@0.5 for both bounding boxes and masks. Given this comparable accuracy but the substantially smaller parameter size of YOLOv9c-seg (28M vs. 44M for Detectron2), we adopt YOLOv9c-seg as the primary detector for all subsequent experiments to reduce model complexity and improve efficiency without sacrificing performance.

Compared to the selected baseline from Paul et al., our baseline didn't reach their performance. However, considering the epoch number, the size of the model, and the unavailability of their dataset, our baseline performance is reasonable and sufficient for us to move forward.

5 Results and Analysis

5.1 Results from Pipeline with Illumination-invariant Feature Learning

In this experiment, we fine-tuned a pre-trained model (on COCO + ExDark datasets) with our pepper dataset for 24 epochs, reducing the learning rate from 1e-3 by a factor of 10 at epoch 18 and 23.

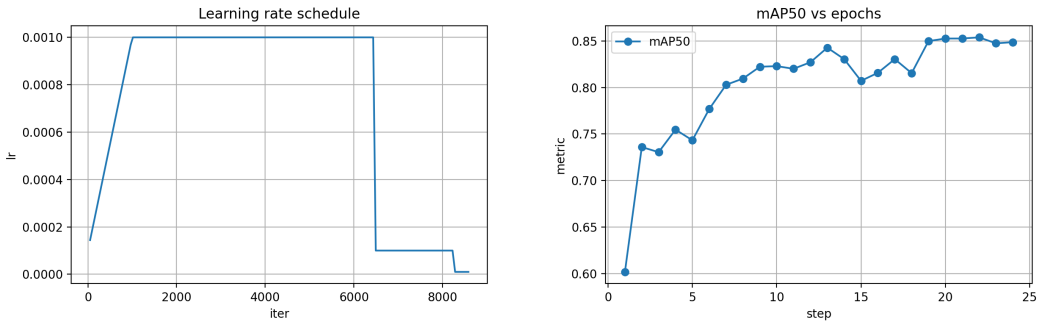

Figure 7: Learning Rate and mAP50 vs Iteration / Epochs.

Table 5: Pepper Detection Performance of Baseline YOLOv9 and Proposed YOLA+YOLOv3.

Training Dataset	Test Dataset	Model	BBox mAP50	BBox P (Precision)	BBox R (Recall)
Normal	Normal	YOLOv9	0.827	0.782	0.809
Normal	Normal	YOLA+YOLOv3	0.854	0.246	0.933
Normal	Low-light	YOLOv9	0.747	0.753	0.707
Normal	Low-light	YOLA+YOLOv3	0.839	0.283	0.911

The mAP50 increases rapidly in the first few epochs, from around 0.60 to approximately 0.75, and then continues to grow more slowly, eventually reaching about 0.85. After roughly 20 epochs, the curve stabilizes with only minor fluctuations, indicating that the model converges without evident overfitting under the current training configuration.

Table 5 reports the quantitative comparison between the baseline YOLOv9 detector and the proposed YOLA+YOLOv3 detector under different training and testing conditions. When both training and testing are performed under normal lighting but evaluated on low-light images. YOLOv9 reaches an mAP50 of 0.747 with a precision of 0.753 and a recall of 0.707. In contrast, YOLA+YOLOv3 attains an mAP50 of 0.839 and increases the recall to 0.911. These results demonstrate that the proposed YOLA IIM module significantly enhances the robustness of the detector, especially when dealing with challenging low-light conditions.

However, this improvement in recall and mAP50 comes at the cost of lower precision. In both evaluation setups, the bounding box precision drops when YOLA is introduced: from 0.782 to 0.246 in the normal-light test and from 0.753 to 0.283 in the low-light test. This indicates that the model detects more true objects but also produces more false positives.

5.2 Results from Pipeline with Image Pre-processing

With this pipeline, we tried two different image pre-processing models: SGZ and IA-YOLO. Unfortunately, IA-YOLO was not successfully trained and tested, which is discussed in the Section 5.3.

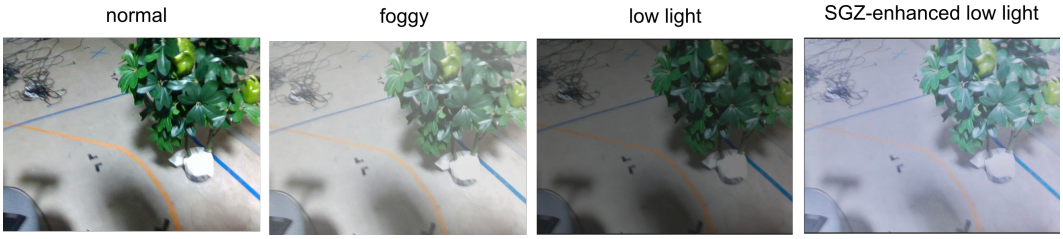

Figure 8: Comparison of Fine-Tuned YOLOv9 Detection Performance on Pepper Images Across Four Test Conditions: Original, Foggy, Low Light, and Low Light Conditions Preprocessed via the SGZ Method

Table 6: Pepper Detection Performance of Baseline YOLOv9 of at Different Conditions and After Preprocessing via SGZ

Dataset Condition	BBox mAP50	BBox P (Precision)	BBox R (Recall)
Normal	0.827	0.782	0.809
Low Light	0.747	0.753	0.707
Foggy	0.710	0.729	0.689
SGZ-Enhanced Low Light	0.595	0.777	0.491

Therefore , we attempted the second approach SGZ for pre-processing of low light images with the aim to improve the detection and segmentation of peppers in low light condition (Fig. 8). As shown in Table 6 Our fine-tuned YOLOv9 is highly effective under ideal conditions, achieving a peak mAP50 of 0.827 on the Normal dataset . When faced with natural degradation like Low Light or Fog, the model shows great resilience, maintaining strong Recall and only dropping to the 0.71 to 0.74 range. However, when tested on the SGZ-Enhanced Low Light data, performance collapses to our

lowest score: 0.595 mAP50. This failure is driven by a massive drop in Recall to just 49. The SGZ pre-processing step, intended to clarify the image, actually destroys the visual features the model relies on to find objects, rendering the model less effective than using the original, noisy images.

With the aim to improve the detection and segmentation of peppers in low light conditions, we applied the SGZ method for preprocessing low-light images. Our fine-tuned YOLOv9 model is highly effective under ideal conditions, achieving a peak BBox mAP50 of 0.827 on the Normal dataset. The model demonstrated a Precision (P) of 0.782 and a Recall (R) of 0.809 in this ideal setting. When faced with natural degradation in adverse environmental conditions, the model's performance decreased: On the Low Light dataset, BBox mAP50 dropped to 0.747, with Precision at 0.753 and Recall at 0.707. On the Foggy dataset, BBox mAP50 dropped further to 0.710, with Precision at 0.729 and Recall at 0.689. However, when tested on the SGZ-Enhanced Low Light data, model performance collapsed to our lowest score: BBox mAP50 of 0.595. This failure was driven by a massive drop in Recall (R) to 0.491, indicating the model was unable to locate over half of the actual peppers. Although Precision (P) remained high at 0.777 (comparable to the normal condition), the poor Recall resulted in an overall highly ineffective system.

This result indicates that the SGZ pre-processing step, intended to clarify the image, actually destroys the visual features the model relies on to find objects, rendering the model less effective than using the original, noisy low-light images. The drastic reduction in Recall suggests that the enhancement process severely altered the object's features (such as color, texture, or edge boundaries), making them unrecognizable as True Positives by the detector.

5.3 Failure Cases Analysis

When attempting to adapt the IA-YOLO framework to our pepper dataset, we encountered substantial implementation and training difficulties. The model failed to learn meaningful features: the validation mAP stayed near zero and the predictions were dominated by oversized or missing bounding boxes, suggesting unresolved issues in the code. Because joint training of YOLO with other modules is computationally expensive (each epoch runs for more than an hour on our hardware), systematically debugging these issues and running full training schedules was not feasible within the project timeline. As a result, we were unable to reliably reproduce the IA-YOLO results reported in the original paper and instead focused our experiments on the other two approaches.

6 Discussion

6.1 Data Augmentation

Fig. 9 illustrates a comparison between the original images from dataset[30], and those generated using a combination of PIL-based parameter adjustments and SD prompts. In the PIL preprocessing stage, brightness, contrast, and color balance were customized for each environmental condition. Additionally, synthetic raindrops and fog gradients were applied to create visually consistent inputs for the subsequent diffusion-based refinement stage. The SD prompts further enhanced the realism and environmental fidelity of the generated images.

The experimental results highlight the pivotal role of PIL-based preprocessing in maintaining structural integrity and visual coherence prior to diffusion-based refinement. Adjusting brightness, contrast, and color balance through PIL enables controlled pixel-level enhancement, producing realistic illumination variations that simulate diverse environmental conditions, such as the contrast between strong and weak lighting shown in the figure. This preprocessing step also provides SD with consistent, high-quality inputs, effectively preventing overcorrection and preserving object boundaries during subsequent semantic generation.

Building upon this foundation, the SD prompts further enhance image realism by incorporating semantic cues related to lighting direction, weather effects, and atmospheric tone, although further exploration is needed to optimize prompt design. Notably, we observed a trade-off between transformation strength and guidance scale: while higher strength encourages environmental diversity, it can distort structural features; conversely, excessive prompt adherence may restrict variation and introduce artifacts such as color oversaturation. The empirically derived optimal settings (`strength = 0.4-0.45`, `guidance_scale = 8.0-9.0`) represent a balanced configuration between realism and diversity.

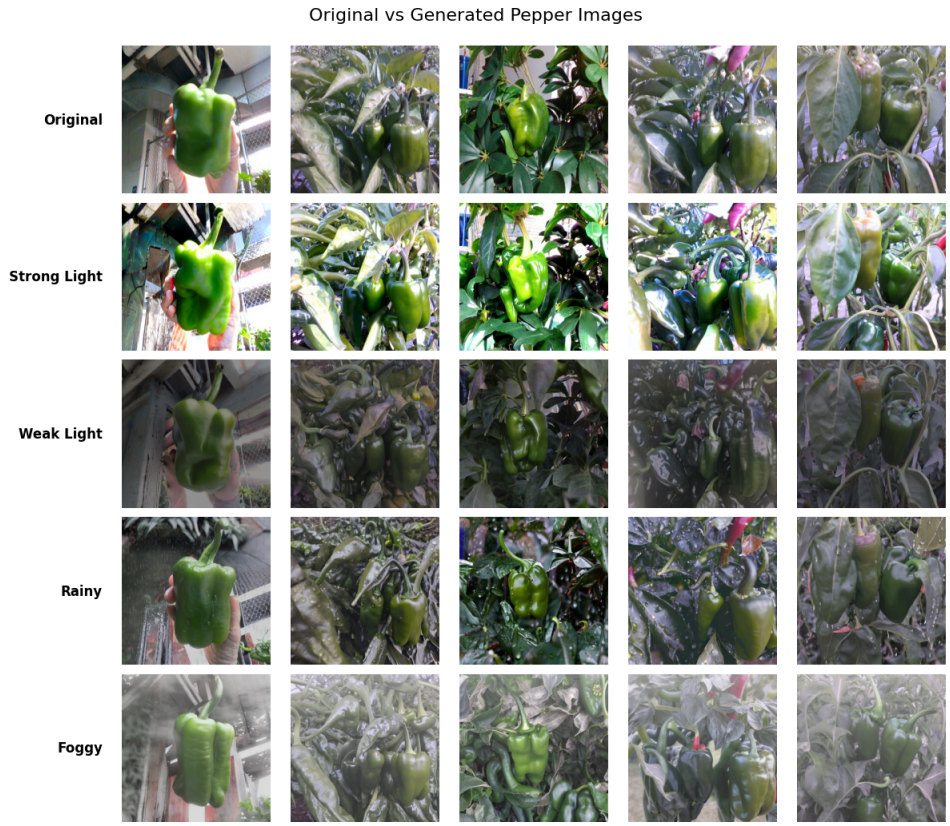

Figure 9: Original and generated images under four prompt conditions: Strong Light, Weak Light, Rainy, and Foggy. Each category corresponds to a distinct lighting or weather scenario.

However, when Stable Diffusion was applied to images with diverse, real-world settings—rather than the close-up pepper images shown in the figure—the fidelity of the generated outputs dropped sharply. In farm scenes where peppers are partially covered by green leaves, the model frequently altered the spatial structure, changing the position, visibility, or even the number of peppers as shown in Fig. 10. Even after adjusting the transformation strength, these inconsistencies remained unavoidable, making the original annotation files unusable. Due to time and workload constraints, we ultimately adopted the PIL-based augmented dataset instead (Fig. 11). By carefully tuning the PIL parameters, we were able to mimic various adverse weather conditions while preserving structural fidelity, allowing us to reuse the existing annotations.

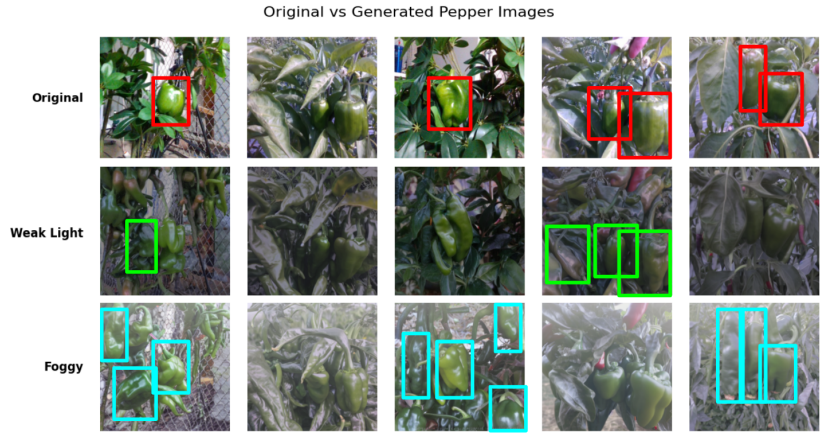

Figure 10: Original and generated images by using PIL-SD hybrid augmentation.

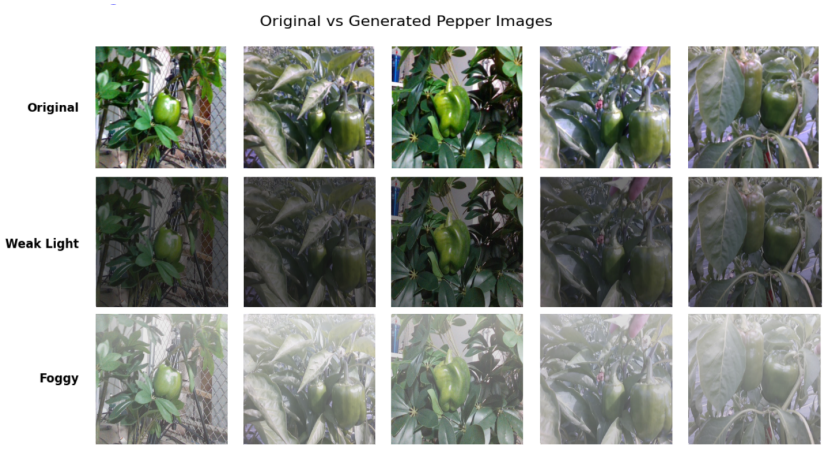

Figure 11: Original and generated images by using PIL-based augmentation.

6.2 YOLO Detector

The normalized confusion matrix in Fig. 12 provides a detailed view of the per-class recognition capability of the fine-tuned YOLOv9 detection model. Overall, the model demonstrates strong classification consistency for both bell pepper and pepper peduncle, but several inter-class confusion patterns are observed.

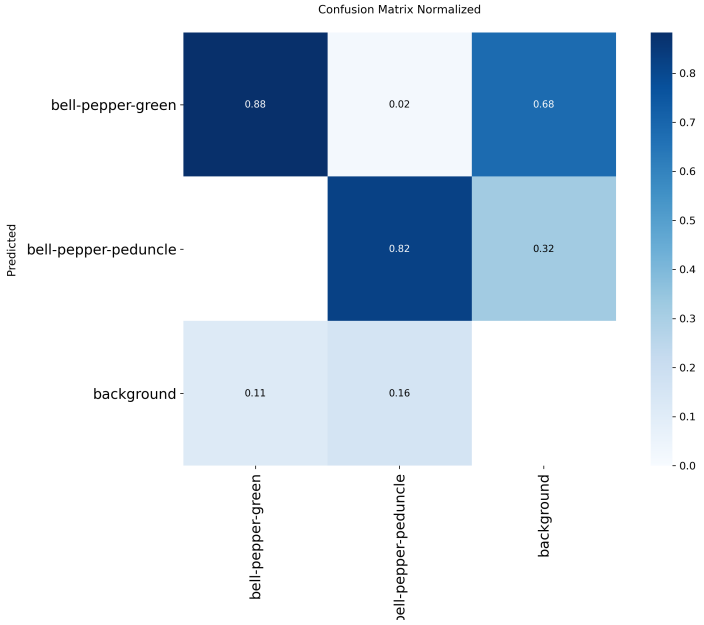

Figure 12: Normalized Confusion Matrix of Fine-tuned YOLOv9 Baseline

The diagonal values, representing correctly predicted objects, show high confidence for both major classes: bell-pepper-green achieves 0.88, indicating that most bell pepper bodies are reliably detected and classified; bell-pepper-peduncle scores 0.82, slightly lower, suggesting that the peduncle (fruit stem) is more challenging to distinguish. In contrast, background samples are sometimes misclassified as pepper objects, with 0.11 and 0.16 confusion rates toward pepper and peduncle, respectively. This reveals that the model still produces a non-trivial number of false positives in cluttered backgrounds.

There are several factors that may contribute to these confusions. Firstly, the edges of the peduncle and shadowed fruit may have similar color intensity distributions under special lighting. Moreover, the difference between pepper and peduncle is diminished when they are occluded by leaves, which is challenging for the model to differentiate. Lastly, considering that green peppers and peduncles

share a similar color to green leaves, the model is more likely to fail in detecting and segmenting the two classes from the background.

Aside from the confusion between classes, the fine-tuned YOLOv9 model also shows flaws in segmenting occluded objects. As shown in Fig. 13, the occluded peppers are boxed as separated peppers instead of as a whole, which can be a problem for automatic harvesting. In addition, the segmentation between pepper and leaves is vague and there is a background area mistakenly recognized as pepper. Although segmenting occluded fruit is not the main task of this research, it will also be a critical bottleneck in the application of fruit harvesting.

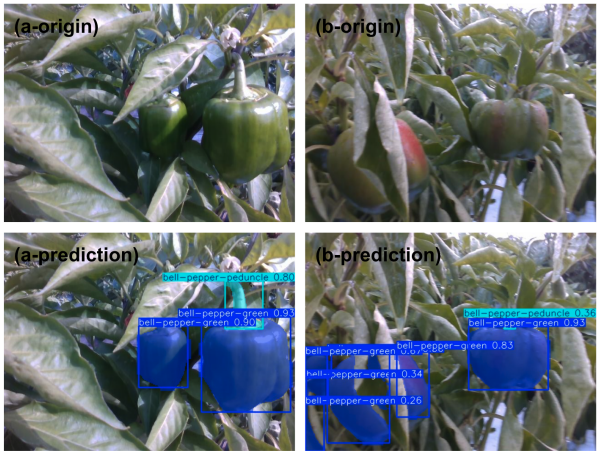

Figure 13: Predicted segmentation from fine-tuned YOLOv9 test set. Image b shows much worse segmentation performance (low confidence and incorrectness) due to leaves occlusion compared to Image a.

6.3 Precision vs Recall Balance

The experimental results in Table 5 show that the YOLA IIM module consistently improves the overall detection performance in terms of mAP50 and, in particular, recall. The gain is especially pronounced in the low-light test scenario, where traditional detectors such as YOLOv9 tend to miss small, low-contrast or heavily occluded objects. This suggests that the IIM module strengthens the spatial and semantic feature aggregation, allowing the network to respond more strongly to weak or ambiguous object cues that would otherwise be suppressed.

From an application perspective, such behavior is highly desirable in safety-critical scenarios, for example in robotic perception, autonomous navigation, or video surveillance, where missing an object can be much more costly than producing a few additional false alarms. In these settings, YOLA can be interpreted as a "recall booster" placed in front of the standard YOLO detector, expanding the set of candidate detections that subsequent modules can further verify or filter.

On the other hand, the noticeable drop in precision reveals a clear trade-off between recall and false-positive rate. The strong feature enhancement provided by IIM likely amplifies not only object-related activations but also background noise and clutter. As a result, the classification head becomes more inclined to fire on ambiguous regions, leading to a larger number of incorrect detections. For applications that require very high precision like pepper harvest, this behavior may be undesirable unless additional mechanisms are introduced to suppress false positives.

7 Future Directions

Although diffusion-based augmentation was initially planned, it was not fully implemented due to the time and workload constraints associated with generating structurally consistent labels. In our preliminary tests, Stable Diffusion produced visually plausible images but often altered object geometry—changing the location, shape, or count of peppers—making the original annotations unusable without substantial post-processing. Developing a reliable workflow for label regeneration exceeded the scope of the current project. In future work, we aim to revisit diffusion-based augmentation by in-

corporating ControlNet, a framework that provides fine-grained structural conditioning through edge maps, depth maps, segmentation masks, or canny features[31]. By anchoring the generative process to scene structure, ControlNet offers a promising solution for maintaining object fidelity while still enabling realistic variations in lighting, weather, and texture. This would allow diffusion models to generate augmented images that remain fully compatible with existing annotations. Beyond structural control, future experiments will also explore systematic prompt engineering and model-parameter tuning to better manipulate environmental attributes such as illumination intensity, shadow quality, fog density, and atmospheric conditions. These studies will help identify prompt and configuration strategies that maximize realism, diversity, and domain relevance.

Regarding future works for IA-YOLO, with more time and computing power, we could first cleanly re-implement it on our dataset, fix the training bugs, and verify that the image processing module is receiving the correct inputs. We could then train the full model for more epochs, possibly with a lighter backbone and newer YOLO variant to reduce training time.

To mitigate false positive issue of YOLA IIM while preserving the recall advantage, here are also some directions to explore. Firstly, loss re-weighting strategies such as Focal Loss or stricter IoU-based classification thresholds could be employed to down-weight low-confidence predictions. Secondly, an additional lightweight attention or saliency-suppression branch after the IIM module could explicitly learn to suppress background responses. Investigating these options is an important step towards making YOLA suitable for a broader range of deployment scenarios.

To enhance SGZ preprocessing performance, the primary strategy should be task-driven enhancement by training the SGZ network end-to-end to optimize the final YOLOv9 detection loss (mAP/mAP50) on the enhanced images. This approach forces the SGZ enhancement to prioritize features that specifically benefit pepper detection, rather than general low-light clarity. Additionally, performance can be improved by refining the semantic guidance of the SGZ network; this involves fine-tuning its Unsupervised Semantic Segmentation (USS) component with labels specific to "pepper" and "foliage" to better differentiate, preserve, and emphasize the unique shape and texture of the pepper from its background.

Moreover, more hyperparameter ablations during model training can be explored for higher mAP with existing pipeline. We can also explore different detectors other than YOLO, such as Detectron2, and transfer the pipeline to segmentation tasks.

8 Conclusion

This project explored three complementary strategies for improving low-light detection robustness: dataset augmentation, low-light-oriented image preprocessing, and illumination-invariant feature learning. PIL-based augmentation was ultimately adopted as the primary approach, as it preserved geometric structure and enabled direct reuse of existing annotations. Among the enhancement methods evaluated, SGZ improved visibility in dark scenes but occasionally degraded detection accuracy due to over-alteration of critical visual cues, suggesting that additional fine-tuning is needed. In contrast, the YOLA method proved overly aggressive for our pepper-detection setting and sometimes introduced artifacts that led to false positives. The diffusion-model augmentation pipeline could not be fully completed within the project timeline, as generating structurally consistent labels for synthetic images required additional development beyond the current scope. While each component was evaluated individually, integrating these strategies into a unified, end-to-end robustness pipeline remains a key direction for future work.

9 Administrative Details

9.1 Team Contributions

The workload is distributed among 4 team members as follow: all 4 team members did literature reviews about object detection and segmentation performance in adverse environments, especially the applications in agriculture industry, while En Zheng focusing on data augmentation, and Jinyao Zhou, Yichen Ji and Xiaolei Hu focusing on detection and segmentation models and image enhancements. In the baseline, methodology, result and discussion sections, En contributed to diffusion model and data augmentation related parts. Jinyao took charge of proposing the overall methodology and

pipeline, fine-tuning YOLOv9 baseline and adaptation of Illumination-Invariant Feature Extractor YOLA. Yichen implemented and fine-tuned Detectron2 baseline and explored the IA-YOLO pipeline. Xiaolei was in charge of implementing SGZ pipeline.

9.2 GitHub Repository

The project code and documentation will be maintained at: `https://github.com/kaleido-jean/CMU-IDL-F25-Project-FruitDetectionInAdverseEnvironment`. The repository will be kept private until final peer review period.

References

[1] Feng Xiao, Haibin Wang, Yueqin Xu, and Ruiqing Zhang. Fruit detection and recognition based on deep learning for automatic harvesting: An overview and review. *Agronomy*, 13(6), 2023. ISSN 2073-4395.

[2] Mingle Xu, Sook Yoon, Alvaro Fuentes, and Dong Sun Park. A comprehensive survey of image augmentation techniques for deep learning. *Pattern Recognition*, 137:109347, 2023.

[3] Nour Eldeen Khalifa, Mohamed Loey, and Seyedali Mirjalili. A comprehensive survey of recent trends in deep learning for digital images augmentation. *Artificial Intelligence Review*, 55(3): 2351–2377, 2022.

[4] P Umesh. Image processing in python. *CSI Communications*, 23(2):23–24, 2012.

[5] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[6] Yuzhen Lu, Dong Chen, Ebenezer Olaniyi, and Yanbo Huang. Generative adversarial networks (gans) for image augmentation in agriculture: A systematic review. *Computers and Electronics in Agriculture*, 200:107208, 2022.

[7] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1):53–65, 2018.

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[9] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[11] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015.

[12] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[14] Dong Chen, Xinda Qi, Yu Zheng, Yuzhen Lu, Yanbo Huang, and Zhaojian Li. Synthetic data augmentation by diffusion probabilistic models to enhance weed recognition. *Computers and Electronics in Agriculture*, 216:108517, 2024.

[15] Suraj Patil, Pedro Cuenca, Nathan Lambert, and Patrick von Platen. Stable diffusion with diffusers. *Hugging Face Blog*, 2022. `https://huggingface.co/blog/stable_diffusion`.

[16] Noriyuki Mori, Hiroki Naito, and Fumiki Hosoi. Application of a latent diffusion model to plant disease detection by generating unseen class images. *AgriEngineering*, 6(4), 2024.

[17] Kun Zhao, Minh Nguyen, and Weiqi Yan. Evaluating accuracy and efficiency of fruit image generation using generative ai diffusion models for agricultural robotics. In *2024 39th International Conference on Image and Vision Computing New Zealand (IVCNZ)*, pages 1–6. IEEE, 2024.

[18] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection, 2024.

[19] Ayan Paul and Rajendra Machavaram. Advancing capsicum detection in night-time greenhouse environments using deep learning models: Comparative analysis and improved zero-shot detection through fusion with a single-shot detector. *Franklin Open*, 10:100243, 2025. ISSN 2773-1863.

[20] Cuixiao Liang, Juntao Xiong, Zhenhui Zheng, Zhuo Zhong, Zhonghang Li, Shumian Chen, and Zhengang Yang. A visual detection method for nighttime litchi fruits and fruiting stems. *Computers and Electronics in Agriculture*, 169:105192, 2020. ISSN 0168-1699.

[21] Junyang Chen, Hui Liu, Yating Zhang, Daike Zhang, Hongkun Ouyang, and Xiaoyan Chen. A multiscale lightweight and efficient model based on yolov7: Applied to citrus orchard. *Plants*, 11(23), 2022. ISSN 2223-7747.

[22] Zhenwei He and Lei Zhang. Multi-adversarial faster-rcnn for unrestricted object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6667–6676, 2019.

[23] Anjana K. Nellithimaru and George A. Kantor. Rols : Robust object-level slam for grape counting. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2648–2656, 2019.

[24] Laura-Sophia von Hirschhausen, Jannes S. Magnusson, Mykyta Kovalenko, Fredrik Boye, Tanay Rawat, Peter Eisert, Anna Hilsmann, Sebastian Pretzsch, and Sebastian Bosse. Applegrowthvision: A large-scale stereo dataset for phenological analysis, fruit detection, and 3d reconstruction in apple orchards, 2025.

[25] Yujin Wang, Tianyi Xu, Fan Zhang, Tianfan Xue, and Jinwei Gu. Adaptiveisp: Learning an adaptive image signal processor for object detection. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 112598–112623. Curran Associates, Inc., 2024.

[26] Mingbo Hong, Shen Cheng, Haibin Huang, Haoqiang Fan, and Shuaicheng Liu. You only look around: Learning illumination-invariant feature for low-light object detection. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 87136–87158. Curran Associates, Inc., 2024.

[27] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. Yolov9: Learning what you want to learn using programmable gradient information. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, pages 1–21, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-72751-1.

[28] Wenyu Liu, Gaofeng Ren, Runsheng Yu, Shi Guo, Jianke Zhu, and Lei Zhang. Image-adaptive yolo for object detection in adverse weather conditions. *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1792–1800, Jun. 2022. doi: 10.1609/aaai.v36i2.20072. URL `https://ojs.aaai.org/index.php/AAAI/article/view/20072`.

[29] Shen Zheng and Gaurav Gupta. Semantic-guided zero-shot learning for low-light image/video enhancement. In *Proceedings of the IEEE/CVF Winter conference on applications of computer vision*, pages 581–590, 2022.

[30] pepperpeople. All pepper datasets dataset. `https://universe.roboflow.com/pepperpeople/all-pepper-datasets`, oct 2023. visited on 2025-10-24.

[31] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023.