International Journal of Computational Methods and Experimental Measurements
https://www.acadlore.com/journals/IJCMEM

DeepHarvestNet: Depth-Visual Fusion Network for Robust Apple Detection in Complex Orchard Environments

Archana Uriti*, Naga Jyothi Pothabathula

Department of Computer Science and Engineering, GSCSE, GITAM (Deemed to be University), Visakhapatnam 530045, India

* Correspondence: Archana Uriti (auriti2@gitam.in)

Received: 07-14-2025 **Revised:** 08-20-2025 **Accepted:** 08-28-2025

Citation: A. Uriti and N. J. Pothabathula, "DeepHarvestNet: Depth-visual fusion network for robust apple detection in complex orchard environments," *Int. J. Comput. Methods Exp. Meas.*, vol. 13, no. 3, pp. 554–575, 2025. https://doi.org/10.56578/ijcmem130307.

Abstract: Accurate fruit recognition in natural orchard environments remains a major challenge due to heavy occlusion, illumination variation, and dense clustering. Conventional object detectors, even those incorporating attention mechanisms such as YOLOv7 with attribute attention, often fail to preserve fine spatial details and lose robustness under complex visual conditions. To overcome these limitations, this study proposes DeepHarvestNet, a YOLOv8-based hybrid network that jointly learns depth and visual representations for precise apple detection and localization. The architecture integrates three key modules: (1) Efficient Bidirectional Cross-Attention (EBCA) for handling overlapping fruits and contextual dependencies; (2) Focal Modulation (FM) for enhancing visible apple regions under partial occlusion; and (3) KernelWarehouse Convolution (KWConv) for extracting scale-aware features across varying fruit sizes. In addition, a transformer-based AdaBins depth estimation module enables pixel-wise depth inference, effectively separating foreground fruits from the background to support accurate 3D positioning. Experimental results on a drone-captured orchard dataset demonstrate that DeepHarvestNet achieves a precision of 0.94, recall of 0.95, and F1-score of 0.95—surpassing the enhanced YOLOv7 baseline. The integration of depth cues significantly improves detection reliability and facilitates depth-aware decision-making, underscoring the potential of DeepHarvestNet as a foundation for intelligent and autonomous harvesting systems in precision agriculture.

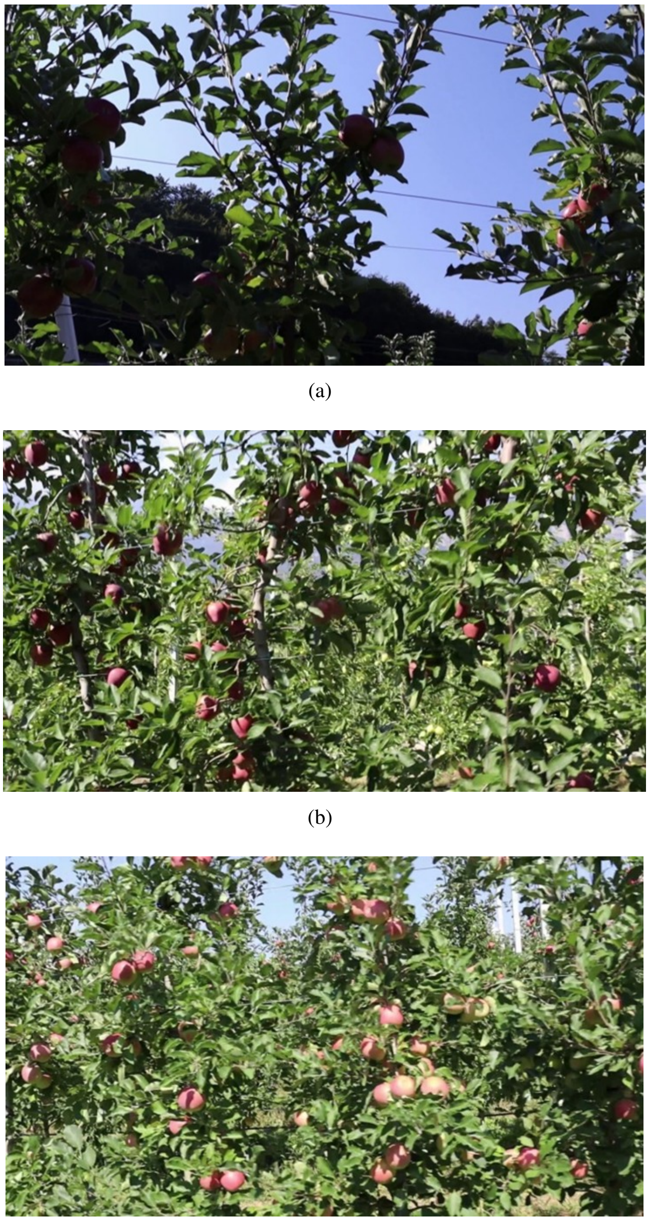
Keywords: Yolov7; Attribute attention mechanism; Adaptive pooling; Yolov8; Efficient bidirectional cross-attention; Focal modulation; KernelWarehouse Convolution; Adabins

1 Introduction

The world's demand for fresh fruits is increasing by the day, putting tremendous pressure on contemporary farming to boost productivity, efficiency, and minimize dependence on labor. As skilled labor increasingly becomes scarce and the cost of manual harvesting increases, most regions of fruit production are aggressively seeking automated solutions. In a variety of fruits, apples rank among the most economically important horticultural crops globally, being extensively grown in a wide range of climatic belts. Apple picking, relying on traditional manual labor, is labor-intensive and time-consuming as a result of the fragile nature of the fruit, inconsistent orchard design, and selective picking on account of ripeness and quality. Apple harvesting automation has the great potential to revolutionize orchard management by increasing consistency in harvesting, minimizing labor dependence, and facilitating constant tracking of yield and quality of fruit. In this regard, artificial intelligence together with computer vision has been a main enabler for the potential to detect, localize, and count apples automatically from sensor data using robotic and drone platforms published.

Bargoti et al. [1] developed a deep learning model by using Faster R-CNN to identify the fruits in high-resolution with an incorporation of data augmentation and it demonstrate the deep fruit detection feasibility for yield estimation. However it struggles with the occlusion, poor localization and lacks depth information. Sa et al. [2] developed a fruit detection system which is named as DeepFruits by using Faster R-CNN and it improves detection accuracy in detecting multiple fruit types. But it has limitation of handling heavy occlusion and lightning variations. Lin et al. [3] proposed a U-Net based segmentation model to identify the cucumber flowers and fruits which enables the automated monitoring plant growth. This model is not tested under occlusion, overlapping conditions and does not perform precise fruit localization.

Even with advancements in automation, precise detection of apples in natural orchard settings is a very challenging task. A few practical issues complicate the process. Occlusion is quite common, as apples are usually occluded behind leaves, branches, or other apples. Overlapping fruits make it difficult to identify instances uniquely, and light changes resulting from sunlight, shadows, and weather conditions further impair detection performance. These challenges are shown in Figure 1. These issues dramatically impact the faithfulness of automated detection systems, particularly when apples are tightly packed or at multiple distances and orientations within the canopy.



(a)

(b)

(c)

Figure 1. (a) Illumination variation, (b) Occluded by leaves or branches, (c) Overlapping of apples

To handle and tackle the agricultural detection problems, deep learning and computer vision usage have been increased in recent years. In which, Convolutional neural networks facilitated YOLO family have been accepted for balancing the real-time execution in fruit detection tasks. Still the performance lacks in handling the occlusion, overlapping, fruit localization and depth of each fruit. This provides a closer look in dealing these limitations with current deep-learning fruit detection models.

The Initial studies utilized the CNN and Mask-RCNN to perform fruit detection and its quality grading. For example, Chu et al. [4] utilized a Suppression Mask R-CNN framework to detect apples in complex backgrounds. It achieved higher precision than standard Mask-RCNN but still struggles in handling occlusion, accurate localization and requires considerable annotated data. Siricharoen et al. [5] proposed Mask R-CNN framework to localize and grade the maturity of pineapple fruits. It doesnot incorporate depth information that limits the precise localization and may not handle occlusion, overlapping fruits well due to controlled conditions. Janowski et al. [6] compared three methods like YOLO, Viola-Jones and Hough transform in automatic apple fruit detection and its count to estimate apple yield. These methods still challenging in handling occlusion and overlapping fruit. Gongal et al. [7] focused on technologies like Vision-based systems, LiDAR and depth sensors for fruit detection and localization in supporting automated harvesting and yield estimation problems. It gives overview on existing systems struggle with issues like occlusion, lightning conditions and precise localization. The integration of all these is cost effective challenge.

While these studies having limitations leads to increase the development of YOLO models which gives high speed with more accuracy in handling fruit detection in an orchard. The YOLO family is demonstrating the improvement in each version for fruit detection tasks and is suitable for this domain. For instance, Xue et al. [8] introduced improvised version of YOLOv2 and is termed as Tiny-YOLO-Dense by integrating into Tiny-YOLO architecture. This work improves the feature extraction in detection of immature mango fruit in an orchard environment. It addresses the challenges in fruit detection but not extensively discuss these challenges and lacking in depth information. Tian et al. [9] proposed an enhanced YOLOv3 model termed as YOLOv3-Dense to detect apples at various growth stages. It demonstrates superior performance than original YOLOv3 and Faster R-CNN in handling illumination and complex backgrounds. However the model struggles with detecting small or occluded apples and lacking depth information. Parico and Ahamed [10] proposed YOLOv4 framework by combining YOLOv4, YOLOv4-CSP and YOLOv4-tiny for real-time pear detection and uses DeepSORT for tracking in video. But it has limitations in handling occlusion and varying illumination. Some pears are missed when partially occluded that leads to errors in counting. Wang et al. [11] proposed a system that is based on YOLOv5 to recognize real time apple stem and calyxes for automated apple fruit harvest. The detection sometimes fails to handle partially occluded by stem and overlapped or shadowed. It used only 2D imaging so lacks in depth information. Ma et al. [12] proposed the enhanced YOLOv7-tiny for detecting apple fruit and counting small apples in complex background and weather. This method shows less performance when apples are very similar to background leading to missed detections. Also it remains challenging when more occlusions or overlapping apples and no depth information available. Liu et al. [13] proposed YOLOv8-MSP-PD which is a lightweight detection model that modified the backbone with MobileNetV4, adds a spatial pyramid pooling for multi-scale feature fusion and applied detection on Jinxiu Malus fruit. This model faces challenges in very abrupt light conditions and heavy occlusion. The total works on YOLO family of models gives a clear focus on how they are applied and suitable for fruit detection.

With this progressive improvements in YOLO models, inspired to research into crop related detection and analysis tasks. Li et al. [14] proposed Lemon-YOLO model that is mainly for lemons detection by modifying the YOLOv3 by a replacement of DarkNet-53 backbone with SE_ResNet module. Still it struggles with very small or severely occluded lemons and lacks depth information. Lin et al. [15] developed AG-YOLO model using a NextViT backbone together with Global Context Fusion Module to improve citrus fruit detection. But still struggles in heavily occluded and complex backgrounds and no depth information. Yu et al. [16] proposed an improved YOLOv5 model for detecting citrus fruit and introduced Receptive Field Convolutions with Full 3D weights to improve feature extraction under occlusion, overlap and variable natural lighting. However it struggle with many fruits missed and does not incorporate depth information. Uriti and Pothabathula [17] developed the enhanced YOLOv5 model with an integration of attribute attention and adaptive pooling that aims to reduce losses and improve performance under challenging conditions for detecting apple fruit. But this model stills struggles in handling occlusion and lacks depth information that limits the precise localization. Zhao et al. [18] proposed the improved YOLOv5 by incorporating the Cross Spatial Partial and Spatial Pyramid Pooling module for apple fruit detection. It shows the degraded performance when apples are densely overlapped and poor lighting and lacks depth information. Liu et al. [19] proposed YOLOv5-ACS which enhances the YOLOv5 model on apple fruit detection. The detection performance drops under severe overlap and occlusion and depth information is not addressed.

Other researchers expanded YOLO to different fruits, including pomegranate detection by Zhao et al. [20], olive fruit detection by Osco-Mamani et al. [21], and apple counting using CNNs by Hani et al. [22]. Fischer et al. [23] further extended fruit detection into tracking using quasi-dense learning. Although these studies improved efficiency, they remained challenged by occlusion, overlapping fruits, scale variation, and illumination changes, often leading to false positives or missed detections. Lightweight models also emerged, such as EfficientDet-Lite2 applied to apples by Agung Enriko et al. [24] and YOLOv7 for fruit health monitoring by Oei et al. [25], which enhanced portability but frequently sacrificed accuracy in complex orchard environments. More recent works, such as RLK-YOLOv8 by He et al. [26] for strawberries and YO-AFD by Wang et al. [27] for apple flower detection, introduced large kernels, multi-stage fusion, and attention mechanisms, achieving notable gains across growth stages. However, even these state-of-the-art YOLOv7 and YOLOv8 models do not explicitly address the combined challenges of partial visibility, overlapping fruits in dense clusters, multiscale variability, and—most critically—the need for three-dimensional depth estimation to support precise fruit localization.

From this, it is clear that while deep learning and YOLO-based detectors have advanced fruit detection across diverse crops, significant research gaps remain in robust apple detection under orchard conditions, particularly in handling occlusion, overlapping fruits, variable scales, illumination variation, and depth-aware localization. To bridge these gaps, the novelty of this study lies in the design of an improved YOLOv8-based detection framework that leverages context-aware spatial feature learning for partially visible fruits, incorporates overlap-aware strategies to separate densely clustered apples, integrates multiscale feature extraction to improve detection across fruit sizes and growth stages, and embeds depth estimation into the pipeline for accurate three-dimensional localization. This combination of improvements directly addresses the limitations of previous approaches and provides a more reliable and practical solution for automated apple detection in real orchard environments.

Though YOLOv7 combined with attribute attention and adaptive pooling showed improved detection under moderate conditions, it still suffered in detecting small or distant apples and failed under heavy occlusion. Additionally, lacking depth perception limited its effectiveness in robotic harvesting, where accurate spatial localization is essential to avoid collisions and false detections.

To resolve these issues, DeepHarvestNet enhances the YOLOv8 backbone with four specialized modules. The EBCA module enables strong bidirectional information flow across feature scales, improving the network's ability to isolate overlapping apples. FM replaces standard attention by dynamically focusing on visible fruit regions, significantly improving performance under occlusion. KWConv learns kernel sizes for maintaining apple detail at multiple scales, particularly those small or partially occluded. AdaBins incorporates depth estimation by learning adaptive bins for pixel-wise accuracy, allowing for robust foreground fruit separation from intricately textured backgrounds. Combined, these enhancements render DeepHarvestNet an end-to-end solution for accurate apple detection and depth-sensitive localization under difficult orchard environments.

The main objective of this work is summarized as below:

- Design a novel model that improves the apple fruit recognition under partial visible and occluded by leaves, branches.
- Enhance the novel model detection performance, incorporate feature learning strategy to handle overlapping fruits. Hence reduce the false positives and missed detections.
- Develop fruit localization model with an integration of multi-scale feature extraction and estimating the depth to filter out the background fruits and better localization.

The main goal of this study is a three stage enhanced process novel model. The first stage is mainly focusing on apple fruit detection under occluded by leaves or branches. Then the second stage improvise the model to handle overlapping of a fruit to reduce false positives and missed detections. The final stage is to focus on mutli-scale features and fruit localization. It provides the depth information of each fruit for handling foreground apples. This overall work is carried out by incorporating four components to handle all these challenges. Each challenge is taken care by individual new component that is added in the backbone and neck of YOLOv8 model.

The added new components in the proposed model is mainly to address the challenges in apple fruit detection for better accuracy in detection and providing depth information of each apple fruit. Here the EBCA is added in the backbone of model to solve the overlapping apples by providing cross regional interactions to identify features from different regions. This will lead to distinguish overlapping fruits by separate adjacent fruit boundaries. The FM module introduced by Yang et al. [28] is to extract the important features by suppressing irrelevant or occluded area. It improves the context-aware that highlights the useful information and thus provide the robust occlusion handling. Li and Yao [29] proposed KWConv module, that aggregates the kernels of different sizes in cpaturing the multi-scale information. This module helps in precise fruit localization and has ability to share apple features across layers in handling scale variations. The AdaBins which is a depth estimation approach proposed by Bhat et al. [30] that predicts the depth by adaptively partioning the bin widths. This method has Mini-ViT transformer module to produce pixel-wise depth of each fruit. The depth information helps in handling the precise localization of fruit and can differentiate the foreground and background fruits to avoid duplicate counts. Adding all these modules in the proposed DeepHarvestNet model will enhance the detection in handling fruit challenges in an orchard and also focus on providing 3D information in depth handling.

The introduced DeepHarvestNet workflow architecture, shown in Figure 2, builds on YOLOv8 with architectural improvements and depth estimation to facilitate secure apple detection. The workflow starts by preprocessing orchard images taken by drones—scaling all frames to a constant resolution and performing augmentations such as brightness adjustments and synthetic occlusions to improve model resilience against real-world lighting and occlusion variations.

For extracting features, the backbone incorporates Efficient Bidirectional Cross-Attention and Focal Modulation. EBCA facilitates multi-scale spatial feature interaction across bottom-up and top-down routes to assist the model in disambiguating overlapping apples, while FM dynamically highlights visible fruit regions in occlusion environments.

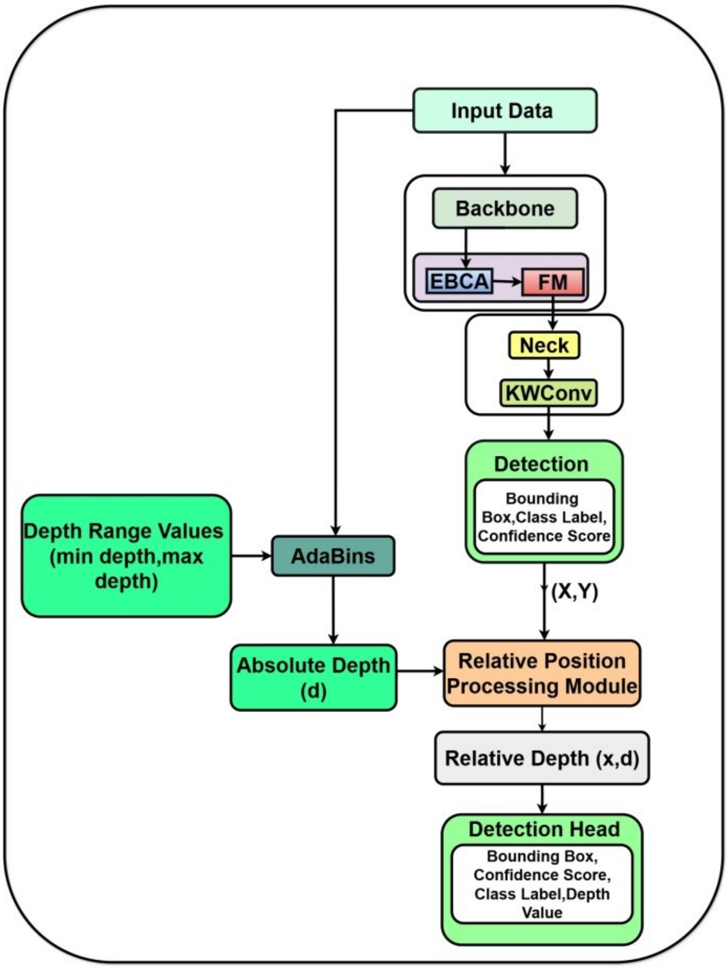

Input Data
Backbone
EBCA
FM
Neck
KWConv
Detection
Bounding Box,Class Label, Confidence Score
(X,Y)
Depth Range Values (min depth,max depth)
AdaBins
Absolute Depth (d)
Relative Position Processing Module
Relative Depth (x,d)
Detection Head
Bounding Box, Confidence Score, Class Label,Depth Value

Figure 2. Workflow of DeepHarvestNet

KernelWarehouse Convolution in the neck modifies kernel sizes according to object size, retaining small or far-away apple details. The detection head subsequently generates bounding boxes, class predictions, and confidence scores. In parallel, a transformer-based AdaBins module predicts pixel-wise depth by discretizing depth into adaptive bins, supporting accurate 3D localization. Bochkovskiy et al. [31] explore the use of Vision Transformers (ViTs) for detection tasks to demonstrate the capability is more than traditional CNNs. This method helps in improving the accuracy in pixel level depth estimation.

This modular pipeline solves important detection problems: EBCA disentangles grouped apples, FM enhances occlusion robustness, KWConv enriches scale-sensitive information, and AdaBins provides depth-conscious localization—rendering DeepHarvestNet very well-suited for robotic harvesting in complicated orchard environments.

2 Literature Review on YOLO-Based Fruit Detection and Depth Estimation

The detection of fruit in orchards has evolved significantly since deep learning has advanced, particularly with the YOLO family of models. The introduction of YOLOv4 proposed by Bochkovskiy et al. [**?**] brought notable improvements in real-time object detection by integrating CSPDarknet53 and PANet, allowing efficient feature reuse and enhanced localization in agricultural images. Building on this, YOLOv5 which is presented by Khanam and Hussain [**?**] optimized training speed and deployment efficiency with its auto-learning bounding box anchor mechanism and model scaling capabilities, making it widely used for fruit recognition. The further development of YOLOv7 by Wang et al. [32] improved inference speed without sacrificing accuracy, enabling high-performance detection in dense and occluded fruit clusters. More recently, Varghese and M [33] introduced YOLOv8 has shown state-of-the-art accuracy in fruit detection tasks due to its decoupled head architecture and streamlined post-processing pipeline. While CNN-based detectors have improved detection accuracy, they often struggle to capture long-range dependencies, especially in cluttered orchard environments. To address this, transformer-based models such as the Pyramid Vision Transformer v2 (PVTv2) have been proposed by Wang et al. [**?**]. PVTv2 introduces improved spatial reduction attention to enhance multi-scale feature representation, making it suitable for complex object detection tasks. Similarly, Tu et al. [34] introduced MaxViT model incorporates both local and global attention using multi-axis attention blocks, which enables better occlusion handling and scale adaptation in dense scenes like fruit orchards.

Incorporating depth information has become crucial for distinguishing overlapping fruits and enhancing localization precision. The AdaBins framework introduced adaptive bin widths for monocular depth estimation, which significantly improved depth prediction in cluttered scenes. In parallel, Godard et al. [35] developed Monodepth2 applied self-supervised learning to monocular RGB images, generating high-quality disparity maps that support 3D understanding in open-field agricultural scenes. These methods enable robust perception even under variable illumination and partially visible targets. Model optimization is also essential for real-time agricultural applications where resources are limited. Ding et al. [36] proposed ResRep pruning strategy introduced a method to decouple the learning and pruning phases, enabling lossless channel pruning and making deep models deployable on edge devices without compromising accuracy. Such efficiency gains are crucial when deploying complex architectures like DeepHarvestNet on UAVs or mobile robots in the field.

Despite these technological advancements, existing systems still face limitations in detecting small, distant, or occluded fruits. Most approaches overfit to certain datasets or are not able to generalize across development stages and settings. This points to the requirement for an end-to-end solution that incorporates spatial attention, depth estimation, and multi-scale feature aggregation to enhance robustness and spatial accuracy. Towards this goal, the proposed DeepHarvestNet architecture merges Efficient Bidirectional Cross-Attention, Focal Modulation, and KernelWarehouse Convolution with AdaBins-based depth estimation to fully tackle issues in fruit detection and 3D localization and the fusion of these methods leads to improved YOLOv8 model proposed by Song et al. [**?**] for accurate detection and localize the tomato stems. Besides that, Liu et al. [37] introduced ConvNeXt, a lightweight ConvNet architecture that pointed out hybrid architectures fusing convolutional inductive bias and transformer pliability are both improved in terms of representation learning as well as feasibility of deployment on edge hardware. Most recently, Uriti and Pothabathula [38] presented a comprehensive review on evaluating the various object detection methods that is used for fruit detection. They highlighted the progress made by various deep learning models and comparison shown in improvement but still noted the challenges remains such as occlusion, overlapping, variable lighting, fruit localization and depth estimation in an orchard. The YOLOv5 model has been also extended towards stronger multi-scale fruit detection via integrating multi-branch heads with better attention blocks and was shown useful for occluded apples among heavy foliage. Yet, under hard occlusion, even attention-based CNNs do not fare well due to a semantic understanding deficiency. To address this, transformer-based detectors with spatial attention and cross-layer aggregation have been put forth, but bring along increased latency.

In short, fruit detection, localization, and depth estimation technology in agriculture has progressed considerably through the combination of deep learning models, attention mechanisms, and novel architectures. Researchers have overcome major issues like occlusion, changing illumination conditions, and overlapping objects through the use of YOLO-based detection architectures, multi-scale feature extraction algorithms, and monocular depth estimation techniques such as AdaBins, which together boost fruit detection system accuracy and efficiency. The introduction of attention mechanisms and feature fusion methods has further advanced the models' ability to focus on certain features, thus improving detection performance in extremely complicated orchard scenarios. Building on top of these advancements, this paper addresses the challenge of enhancing fruit recognition systems by incorporating even more advanced attention modules and state-of-the-art depth estimation approaches, with the aim of improving spatial precision and adaptability for real-time crop management systems. Through the application of depth-aware recognition combined with advanced localization methods, the system proposed here aims to mitigate longstanding problems of object overlap and visual blocking ubiquitous in orchard-based fruit detection applications. In the future, ongoing research will probably focus on further enhancing such frameworks to gain stable real-time operation under varied agricultural environments. This survey of recent methods forms the basis for designing the envisioned DeepHarvestNet architecture, which combines state-of-the-art deep learning approaches, attention-based feature augmentation, and depth estimation to address orchard fruit detection complexities directly. Through the solution of detection, localization, and depth estimation simultaneously, the envisioned framework aims to provide more accurate, scalable, and efficient solutions that shape the future of smart agriculture.

3 Materials and Methods

This section discusses the methodology used to design the proposed DeepHarvestNet framework for accurate apple fruit detection and depth-aware 3D localization in orchard settings. The process is divided into three major phases: (i) Data Acquisition, (ii) Data Augmentation and Preprocessing, and (iii) Model Development, where the DeepHarvestNet architecture utilizes several state-of-the-art modules to enhance detection accuracy, occlusion robustness, and accurate 3D spatial localization.

3.1 Data Acquisition

The accuracy and dependability of the model are greatly dependent on the diversity and quality of data that is gathered. Joora Drones Pvt. Ltd., being a service organization based in Visakhapatnam district, Andhra Pradesh state, and incubated in an innovation center, conducted the data collection using the DJI Mavic 3 drone, a high-tech drone that is characterized by its lightness, improved flight stability, and high-resolution video capacity. This drone comes equipped with a Hasselblad L2D-20c camera that has a 4/3 CMOS sensor, 24 mm equivalent focal length, and 84° field of view. The aperture varies from f/2.8 to f/11 so that it is flexible in all types of lighting conditions, and the camera can record up to 5.1K video resolution along with 20MP photo capture. The drone was operated at altitudes ranging from 20 to 60 meters over chosen agricultural fields in India, recording canopy-level and full-plot images under ambient light conditions. Videos were recorded at varying angles and weather conditions ranging from clear to partly cloudy skies, with no artificial lighting. The three-axis gimbal of the drone, GPS hover, and vision sensors provided stability and accuracy, which resulted in reliable and distortion-free footage.

3.2 Data Augmentation

In this study, various data augmentation techniques were applied to the original apple orchard images to improve the robustness of the detection model under diverse real-world conditions. Figure 3 illustrates the representative augmentations applied to the original dataset, which consists of drone-captured apple orchard images obtained from publicly available sources. These augmentations include brightness variation, rotation, motion blur, mirroring, and Gaussian noise—each designed to simulate challenges typically encountered in aerial agricultural imaging.

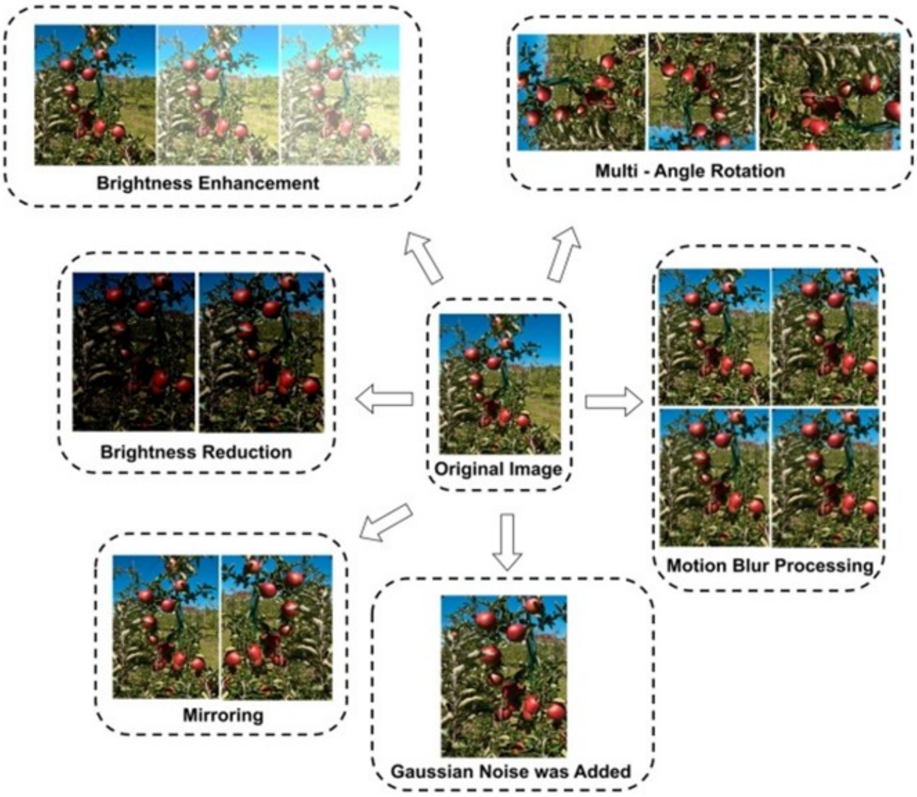


Figure 3. Data augmentation process

• Brightness Variation: To simulate lighting fluctuations, the RGB images were first converted into HSV color space using the ‘rgb2hsv’ function. The brightness component (V channel) was scaled using different coefficients. For brightness enhancement, the formulas H+S+1.2V and H+S+1.6V were used, while for brightness reduction, H+S+0.6V and H+S+0.8V were applied. The modified images were converted back to RGB using the ‘hsv2rgb’ function. These changes help the model learn to detect apples under both high- and low-illumination conditions.

• Image Rotation: The images were rotated by fixed angles of 900, 1800 and 2700to mimic different camera orientations. This rotation helps the model maintain detection accuracy even when the image capture angles vary.

• Mirroring: Horizontal image flipping was performed by mirroring pixel values along the vertical axis. This technique increases dataset diversity and allows the model to generalize symmetric apple features effectively.

• Motion Blur: To simulate the blurring effect caused by drone movement during aerial data collection, four motion blur kernels were applied with configurations of (6, 30), (6, 135), (7, 45), and (7, 90)—where the first number represents the blur length and the second the angle in degrees. This prepares the model to detect apples even when motion artifacts are present.

• Gaussian Noise Addition: Gaussian noise with a variance of 0.02 and zero mean was added to simulate sensor noise caused by environmental factors such as poor lighting or electronic interference. This makes the product more resistant to deterioration in image quality.

By these augmentation techniques, the dataset was profoundly enhanced, and the detection model could learn more generalizable features and function consistently under diverse imaging conditions.

3.3 Proposed DeepHarvestNet Architecture for Enhanced Detection Framework and Depth Estimation

The designed DeepHarvestNet presented in Figure 4 is a shared model that uses deep learning for object detection and depth estimation of spatial locations. The architecture has a tailored backbone, an improved neck structure, and a dual-head module for depth estimation and object detection. The model can be implemented on high-resolution orchard images of size 640×640 to precisely calculate spatial locations and 3D scene understanding in intricate agricultural environments.

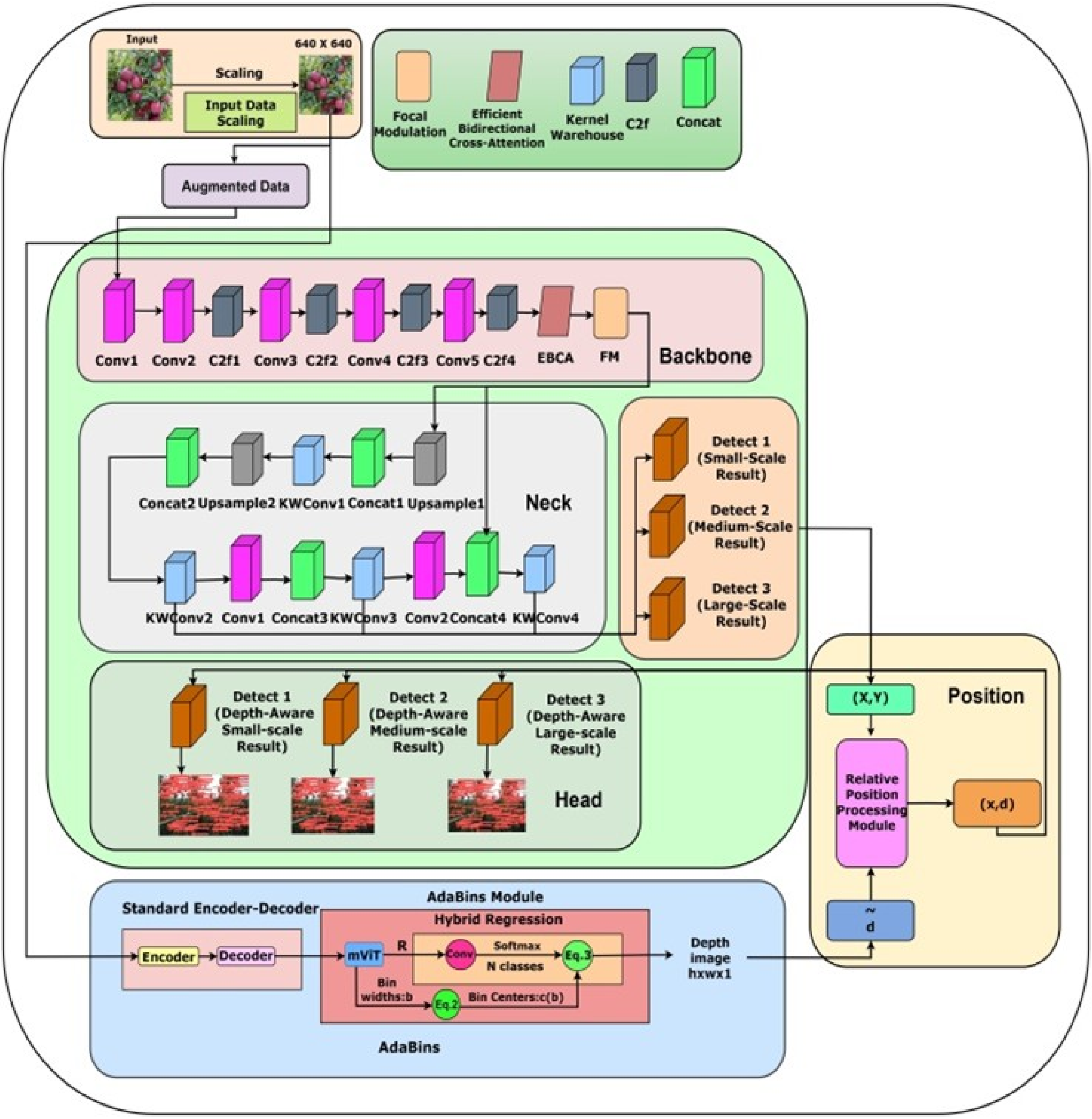


Figure 4. Data augmentation process

The DeepHarvestNet architecture proposed here adds an improved object detection and depth prediction framework designed for apple fruit localization under orchard scenarios. As shown in Figure 4, the architecture comprises four main components: (i) input preprocessing and augmentation, (ii) feature extraction backbone, (iii) multi-scale feature fusion neck, and (iv) hybrid detection and depth estimation head combined with 3D position refinement.

The input pipeline starts by resizing all raw orchard images to a uniform resolution of 640×640 pixels in order to have uniform input dimensions for training and inference. Augmentation techniques such as random brightness normalization and synthetic occlusion generation are used to make the model robust against typical orchard issues such as inconsistent illumination and partial exposure of fruit.

In the Backbone, hierarchical visual features are learned by sequential convolutional layers (Conv1 to Conv5). For strengthening feature learning, C2f modules are introduced at various points, followed by two essential attention-based innovations: Efficient Bidirectional Cross-Attention and Focal Modulation. EBCA facilitates long-range bidirectional feature interactions such that the model can differentiate between overlapping apples by relating spatially distant yet semantically relevant regions. Simultaneously, FM substitutes traditional self-attention with dynamic feature suppression of occluded areas and highlighting visible parts of apple fruits, thus enhancing detection robustness against severe occlusions.

The extracted features are passed into the Neck, where multi-scale feature fusion is performed. Here, feature maps are first upsampled and concatenated, followed by the application of KernelWarehouse Convolution blocks. KWConv adaptively selects convolutional kernels based on object scale, ensuring the preservation of fine-grained details for both small and large apple instances across multiple spatial scales. This allows the model to maintain scale-invariant detection performance even under varying fruit sizes and distances.

The output from the Neck feeds into the Detection Head, which operates at three spatial scales (small, medium, and large) to localize apple fruits of different sizes. Each scale outputs bounding boxes, class labels, and confidence scores. Simultaneously, depth estimation is performed through a dedicated AdaBins Module. The AdaBins depth estimation pipeline consists of a standard encoder-decoder network augmented with a Mini Vision Transformer (mViT) that captures global context information, followed by a hybrid regression mechanism that discretizes depth predictions into adaptive bins. This enables pixel-wise absolute depth estimation directly from monocular RGB input, producing precise distance information for each detected apple.

Finally, the detection outputs (X, Y coordinates) and the estimated depth (d) are combined in the Relative Position Processing Module (RPPM). This module fuses positional and depth information to generate complete 3D localization data (X, Y, d), facilitating depth-aware robotic harvesting and accurate yield estimation.

3.3.1 Efficient bidirectional cross-attention for overlapping object separation

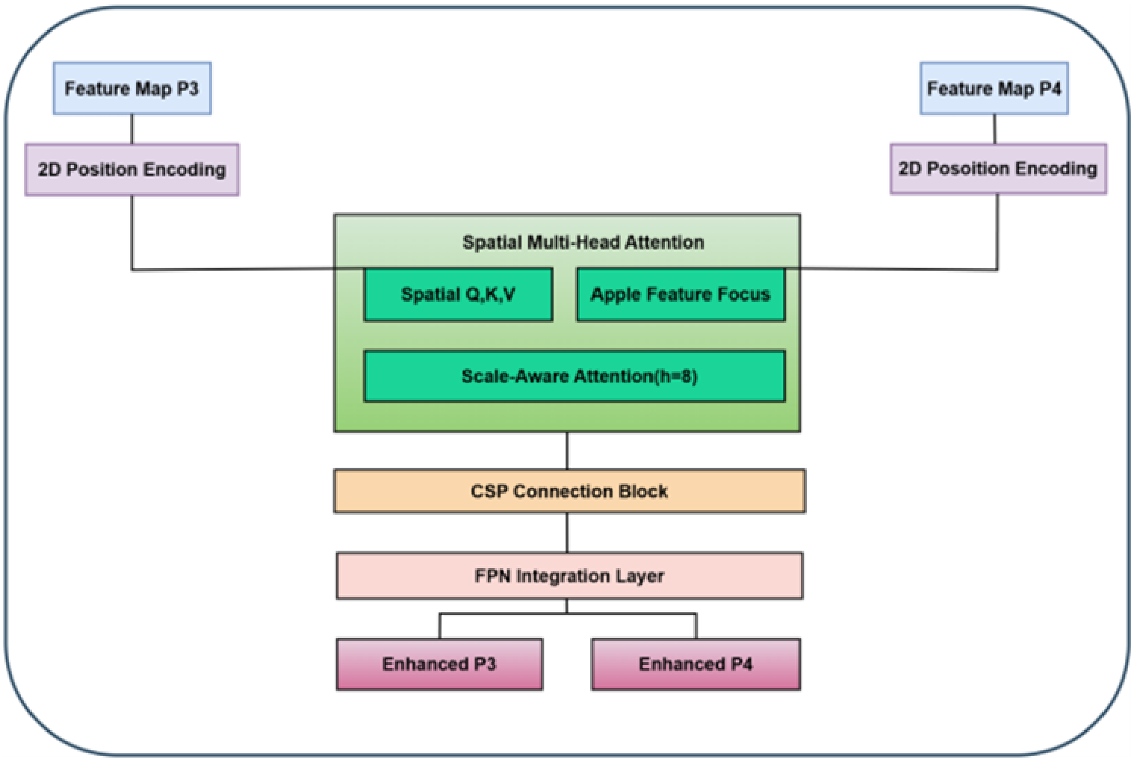


Figure 5. Efficient bidirectional cross-attention architecture

The Efficient Bidirectional Cross-Attention mechanism is designed to enhance the localization accuracy of apple fruits, particularly in scenarios where multiple apples overlap or appear closely clustered. By modeling bidirectional interaction across spatial features from multiple scales, EBCA allows the network to distinguish between individual apple instances even in complex orchard environments. Zhu et al. [39] proposed vision transformer architecture which is termed as BiFormer which is mainly for efficient feature representation by capturing local and global dependencies with the help of bi-level attention mechanism. The architecture is shown in Figure 5.

The EBCA module receives multiscale feature maps denoted as, denoted as $F_3 \in R^{H_3 * W_3 * C}$ and $F_4 \in R^{H_4 * W_4 * C}$, where H, W, and C represent height, width and channel depth, respectively. These feature maps are first enhanced with 2D positional encodings, denoted as PF_3 and PF_4 to incorporate spatial information critical for distinguishing closely positioned apples. The positionally encoded feature maps are formulated as:

$$F_3^1 = F_3 + PF_3, \quad F_4^1 = F_4 + PF_4 \tag{1}$$

To capture the relationships between the enhanced multiscale features, spatial multi-head attention is computed by projecting the feature maps into query(Q), key(K) and value(V) vectors using learned projection matrices. Spatial multi-head attention is computed using $\mathrm{Q} = \mathrm{W_Q F_3^1},\ \mathrm{K} = \mathrm{W_K F_4^1}$, and $\mathrm{V} = \mathrm{W_V F_4^1}$ producing:

$$\mathrm{Attention(Q, K, V)} = \mathrm{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)\mathrm{V} \tag{2}$$

where, d_k is the dimension of the key vector.

To focus attention specifically on apple fruit regions, an Apple Feature Focus mask Ma is applied to emphasize fruitrelevant spatial regions, by computing a 1×1 convolution over the concatenated positionally encoded features $F_3^1 \oplus F_4^1$:

$$M_a = \sigma\left(Conv_{1*1}\left(F_3^1 \oplus F_4^1\right)\right) \tag{3}$$

where, $\bigoplus$ denotes channel-wise concatenation and σ represents the sigmoid activation function. The final attended feature is obtained by applying element-wise multiplication $\odot$ between the attention output and the mask:

$$F_{attn} = M_a \odot \text{ Attention } (\mathrm{Q},\ \mathrm{K},\ \mathrm{V}) \tag{4}$$

To incorporate multi-resolution cues and scale-awareness, scale-aware attention with eight attention heads is applied. This enables the model to capturing diverse apple fruit sizes across orchard scenes:

$$MSA\left(F_3^1, F_4^1\right) = Concat\left(head_1, \ldots\ldots head_8\right) W_0 \tag{5}$$

where each head captures spatial dependencies at different scales, improving detection robustness for both small distant apples and larger foreground instances.

Following attention refinement, the resulting features are passed through a Cross-Stage Partial (CSP) block to enhance learning stability and preserve feature diversity:

$$F_{CSP} = Concat\left(F_{attn}^{(1)}, Conv\left(F_{attn}^{(2)}\right)\right) \tag{6}$$

Finally, an FPN (Feature Pyramid Network) integration layer aligns the channels and outputs the enhanced multiscale feature maps, which are:

$$\text{Enhanced P3, Enhanced P4 } = \mathrm{FPN}\left(F_{CSP}\right) \tag{7}$$

This full EBCA design allows the model to effectively isolate individual apple fruits even in highly clustered, overlapping orchard scenes, significantly improving object separation accuracy and overall localization precision for apple harvesting applications.

3.3.2 Focal modulation for occlusion handling and feature enhancement

The Focal Modulation module dynamically refines feature representations by emphasizing relevant spatial regions while suppressing irrelevant or occluded areas. This selective attention mechanism is particularly effective for handling occlusion scenarios commonly encountered in apple orchards, where fruits may be partially blocked by leaves, branches, or overlapping apples illustrated in Figure 6. By focusing attention on visible apple features and diminishing background noise, FM enhances both object detection accuracy and subsequent depth estimation performance.

The FM mechanism operates through a four-step pipeline, transforming and enriching feature representations for improved downstream tasks. The steps involved are as follows:

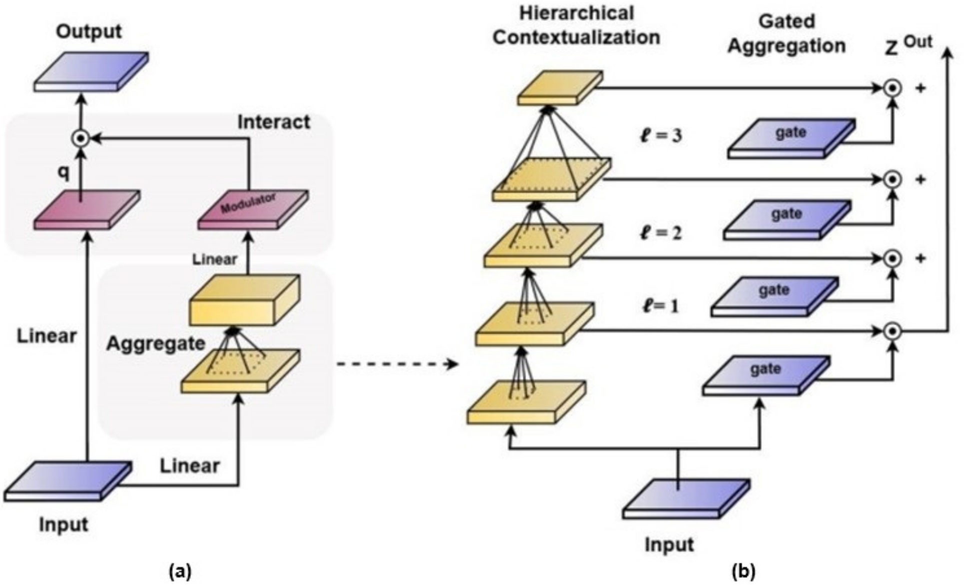


Figure 6. (a) Focal modulation, (b) Context aggregation

1. Linear Transformation

The input feature map X is projected into a new representation space using a learnable weight matrix W:

$$X^1 = X.W \tag{8}$$

This operation extracts relevant feature vectors that serve as the foundation for subsequent context aggregation and modulation.

2. Aggregation Operation

A global context representation F_C is obtained by aggregating the transformed features through mean pooling:

$$F_C = \frac{1}{N} \sum X^1 \tag{9}$$

This aggregated context F_C captures essential spatial information across the feature map, providing reference signals for enhancing task-relevant features while minimizing background distractions in apple detection.

3. Modulation Operation

A learned modulator applies element-wise interactions to enhance or suppress certain features. The modulation process is defined as:

$$F_m = q \cdot \sigma\left(F_c\right) + F_c \tag{10}$$

Here q is the modulated query, and σ is the activation (e.g., sigmoid), which adaptively emphasizes apple-relevant features while suppressing occluded or irrelevant regions.

4. Output Transformation

The modulated features F_m undergo a final transformation to obtain the output feature map Y:

$$Y = W_m \cdot F_m \tag{11}$$

The resulting feature map Y is then passed into subsequent detection heads for refined prediction of apple locations, even under challenging occlusion conditions.

5. Hierarchical Contextualization and Gated Aggregation

To further improve contextual understanding, the modulated output is processed through a hierarchical aggregation structure, capturing multi-level semantic features critical for robust apple fruit detection. This structure operates at three semantic levels:

Level $\ell = 1$: Extracts fine-grained local features of individual apples

$$f_1(x) = Conv(x) \cdot W_1 \tag{12}$$

Level $\ell = 2$: Encodes mid-level spatial clusters, including apple bunches and partial occlusions

$$f_2(x) = Conv(x) \cdot W_2 \tag{13}$$

Level $\ell = 3$: Captures broader context, such as rows of trees and orchard structure

$$f_3(x) = Conv(x) \cdot W_3 \tag{14}$$

Each level's output is gated to emphasize critical patterns:

$$g_i(x) = \sigma\left(W_{g_i} . f_i(x)\right), i = 1, 2, 3 \tag{15}$$

The final aggregated output Z_{out} combines all levels:

$$Z_{\text{out}}(x) = g_1(x) \cdot f_1(x) + g_2(x) \cdot f_2(x) + g_3(x) \cdot f_3(x) \tag{16}$$

This hierarchical approach enables the model to attend to both fine details (individual apples) and global orchard-level patterns, thereby providing robust detection even in densely populated and visually complex apple orchard environments.

Chu et al. [40] developed occluder-occludee network which is called as O2RNet to enhance the detection accuracy under severe occlusion conditions. It provides the spatial relationship between occluding and occluded fruits in an orchard that helps in better detection. The Focal Modulation module offers a robust solution for managing heavy occlusion in apple detection tasks. By adaptively regulating feature importance based on context-sensitive modulation and hierarchical aggregation, the model is capable of accurately localizing visible apples while completely suppressing interfering factors. This adaptive approach greatly improves the model's performance in detecting apples in cluttered orchard scenes, providing both high detection accuracy and better depth estimation.

3.3.3 Kernel Warehouse Convolution for scale-adaptive apple localization

To dynamically scale receptive fields and better extract diverse contextual information in complicated orchard images, the proposed KernelWarehouse Convolution module provides an adaptive convolutional scheme to enhance localization accuracy of apple fruits under different size, scale, and density situations illustrated in Figure 7. By assembling scale-aware convolution kernels from a learnable warehouse, KWConv enhances the spatial sensitivity and robustness of the detection pipeline.

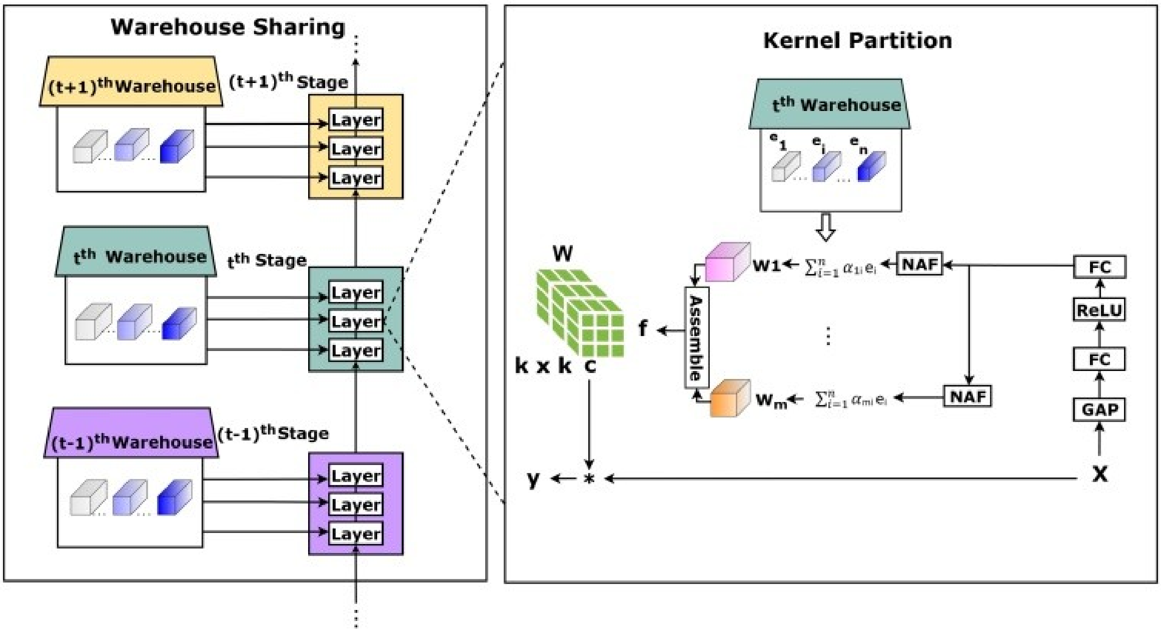


Figure 7. Workflow of KernelWarehouse Convolution

1. Warehouse Sharing Framework

KWConv is structured around a multi-stage feature extraction pipeline, where each stage t maintains a dedicated kernel repository known as the t^{th} Warehouse, contains a collection of learnable kernel elements $\{e_1, e_2, \ldots\ldots e_n\}$ shared across multiple convolutional layers within the stage.

- Kernel Reusability: Instead of training new convolution kernels for each layer independently, KWConv reuses a shared kernel set, improving consistency of learned features across layers and reducing computational overhead.
- Stage-Wise Interaction: Warehouses are progressively updated across stages $(\mathrm{t}-1) \rightarrow \mathrm{t} \rightarrow (\mathrm{t}+1)$, allowing bidirectional refinement of kernel representations. This inter-stage flow enhances the adaptability of the model to changing spatial distributions of apple features across the network depth.

2. Kernel Partition and Dynamic Assembly

KWConv dynamically assembles convolution kernels that are adapted to spatially varying content. Each kernel w_j is constructed as a weighted sum of base kernel elements from the warehouse:

$$w_j = \sum_{i=1}^{n} \alpha_{ji} \cdot e_i \quad \mathrm{j} = 1, \ldots,\ \mathrm{m} \tag{17}$$

where,

- e_i is the i^{th} kernel element in the warehouse.
- α_{ji} represents the attention weight that determines the contribution of e_i to the j^{th} assembled kernel w_j.
- m is the total number of adaptive kernels used in the layer.

This mechanism supports content-aware and spatially adaptive convolution, enabling the model to better localize apples across varying sizes and densities.

3. Gating and Attention Mechanism

To compute the attention weights α_{ji}, a global descriptor is extracted from the input feature map X using Global Average Pooling (GAP), followed by a two-layer fully connected network with a ReLU activation:

$$g = RELU\left(FC_2\left(FC_1(GAP(X))\right)\right) \tag{18}$$

This global descriptor g is fed into a Non-Linear Attention Function (NAF), which produces attention coefficients for every kernel element.

$$\alpha_{ji} = NAF_j(g) \tag{19}$$

These weights guide the selective kernel composition process so that each output kernel is shaped by the most contextually relevant warehouse components.

4. Adaptive Convolution Operation

Adaptively constructed kernels $\{W_1, W_2, \ldots, W_m\}$ is then used on the input feature map through normal convolution.

$$y = w * f \tag{20}$$

where,

- $W \in R^{k*k*c}$ is the dynamically generated convolution kernel tensor.
- f represents the input feature map.
- $*$ denotes convolution operation.
- y refers the output feature map.

The KWConv module presents an efficient approach to multi-scale feature adaptation with the integration of kernel reuse, composition-by-attention, and hierarchical refinement. Its capacity to produce spatially adaptive filters custom-fit to input features renders it well-adapted to intricate apple orchard environments where the sizes, locations, and occlusion rates of fruits differ greatly. By refining the localization process, KWConv significantly boosts detection performance and contributes to more accurate and scale-aware apple fruit recognition.

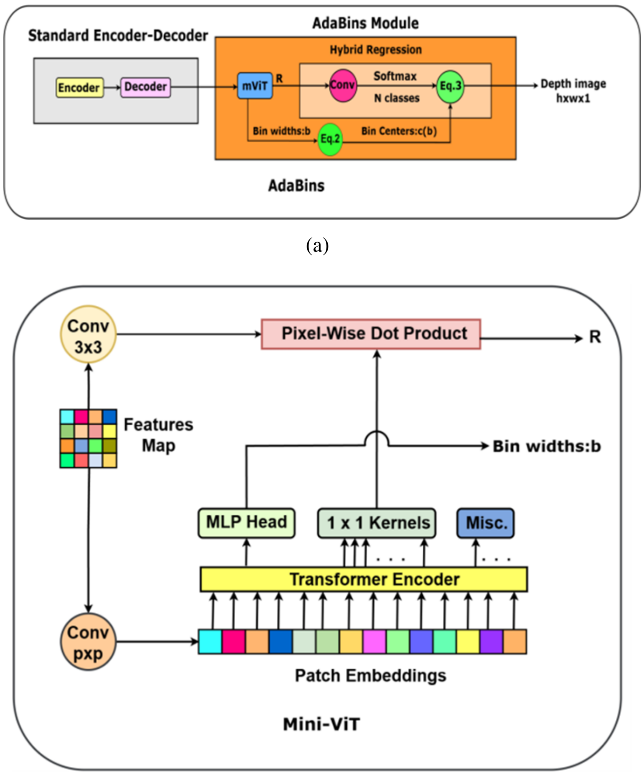
3.3.4 AdaBins for absolute depth estimation of apple fruits

To incorporate depth awareness into the DeepHarvestNet architecture, we integrate the AdaBins module, which has demonstrated superior performance in monocular depth estimation. This integration is particularly valuable for enhancing spatial precision and improving the detection of foreground apple fruits under challenging orchard conditions involving overlapping canopies, variable fruit distances, and occlusions.

An Adaptive binning technique is introduced by the AdaBins architecture illustrated in Figure 8, allowing for accurate absolute depth estimate from single monocular RGB pictures. The core innovation lies in its ability to learn non-uniform, data-driven depth bin centers, dynamically adjusted based on scene content. This allows the system to effectively handle varying depth distributions commonly observed in orchard environments.

The AdaBins depth estimation pipeline comprises three key components:

i. a standard encoder-decoder architecture.
ii. a Mini Vision Transformer (mViT) module.
iii. a hybrid depth regression block for pixel-wise depth prediction.



(a)

(b)

Figure 8. (a) AdaBins architecture, (b) mViT architecture

1. Input Specification

The AdaBins module receives two primary inputs:

- A monocular RGB image captured from the orchard.
- A user-defined depth range values $[u_{\min}$ and $u_{\max}]$ which specifies the possible minimum and maximum distances corresponding to the orchard layout.

This configuration enables the model to regress absolute depth values directly, rather than relying on relative or normalized depth scales.

2. Encoder-Decoder Architecture

The RGB input image is initially processed by a convolutional encoder-decoder pipeline:

- The encoder extracts hierarchical features while progressively reducing spatial resolution.
- The decoder reconstructs a dense feature map, preserving essential semantic structures necessary for subsequent adaptive binning and depth regression stages.

3. Adaptive Bin Width Estimation via MiniViT

To capture contextual and spatial cues:

- A lightweight Mini Vision Transformer (mViT) processes the decoded feature map, modeling long-range dependencies and global scene structure.
- The mViT output is fed into a bin regressor that predicts soft weight parameters www, which define the adaptive bin centers.

The formula to calculate the bin centers b which is available in the Figure 8a. AdaBins architecture gives the adaptive bin centers b and is computed as:

$$\mathrm{b} = \mathrm{softmax}(\mathrm{w}) \cdot (\mathrm{u}_{\max} - \mathrm{u}_{\min}) + \mathrm{u}_{\min} \tag{21}$$

where,

- b represents the set of predicted bin centers distributed across the specified depth range.
- The softmax normalization ensures that the bins are dynamically adjusted in response to the scene's depth complexity.

This adaptive binning allows the network to allocate finer resolution to regions where depth variation is high (e.g., clusters of overlapping apples) while using coarser resolution for homogeneous regions (e.g., uniform backgrounds).

4. Bin-wise Classification and Depth Regression

For each pixel, AdaBins performs a bin classification, assigning soft σ_k across K adaptive bins. The final absolute depth d for each pixel is computed through hybrid regression as represented with the depth image in the Figure 8a. AdaBins Architecture and is computed as:

$$d = \sum_{k=1}^{K} \sigma_k * b_k \tag{22}$$

where,

- d is the estimated depth at the pixel,
- K is the total number of bins,
- σ_k is the soft assignment probability for the K^{th} bin (it represents each pixel belonging to which bin),
- b_k is the corresponding bin center determined in Eq. (2).

This hybrid strategy effectively combines classification confidence with continuous regression to produce precise depth estimates at pixel level.

5. Output: Absolute Depth Map

The last output of the AdaBins module is a dense absolute depth map where every pixel stores its real-world distance from the camera. This depth map greatly helps separate foreground apples from background features like leaves, branches, and orchard rows, thus enhancing both localization and counting accuracy.

3.3.5 Relative Position Processing Module (RPPM)

The Relative Position Processing Module is proposed to improve object localization accuracy by improving the spatial depth relations between apple fruits and background structures. It is especially useful for handling viewpoint changes, camera displacements, and depth normalization across frames to ensure robust and stable apple fruit identification under different orchard conditions.

Workflow of RPPM

- Input from AdaBins: The absolute depth map $d(x, y)$ produced by the AdaBins module is used as the input to the RPPM. Concurrently, the spatial coordinates (x, y) corresponding to candidate apple fruit detections from the neck layer of the detection network are given as input to the module.
- Conversion to Relative position: Since the absolute depth values may vary due to camera orientation, platform movement, and orchard terrain, it is essential to compute depth-relative positioning that accounts for the optical geometry of the imaging setup. To achieve this, the absolute depth $d(x, y)$ for each detected apple is adjusted relative to the optical center (c_x, c_y) of the camera:

Refining depth using (x, y) is computed as:

$$D_{final}(x, y) = d(x, y) * \left(1 + \alpha \frac{x - c_x}{c_x} + \beta \frac{y - c_y}{c_y}\right) \tag{23}$$

where,

- $D_{final}(x, y)$ is the refined depth after processing relative position.
- $d(x, y)$ is the absolute depth value from AdaBins at pixel (x, y).
- c_x and c_y are the optical center coordinates of the image.
- (x, y) values represent the center coordinates of the fruit in the image.
- c_x and c_y can be taken by using the image dimensions.

$$\left(c_x = \frac{image\ width}{2}, c_y = \frac{image\ heigh}{2}\right)$$

- α and β can be determined by using $\alpha = \frac{B}{f_x}, \beta = \frac{B}{f_y}$.
- f_x and f_y are the focal lengths along x and y direction.
- B is the baseline parameter that represents an offset factor related to camera position.

Table 1. Depth thresholds

Distance (cm)	Max. Depth (m)
20	0.9
40	1.2
60	1.4
80	1.6

After obtaining the relative depth values, the next step involves classifying whether the detected apple fruits lie in the foreground or background. This is achieved by applying depth thresholding, where predefined depth limits are used to segment apples based on their distance from the camera. The threshold values, as presented in Table 1, are determined by considering both the drone's flying altitude and the maximum observable depth range of the camera system within the orchard environment.

The algorithm to consider the foreground apples only is as below steps:

Step 1: Check if camera_distance exists in the table

- If yes, use its corresponding max depth.
- If no, find the two closest distances and interpolate the max depth.

Step 2: Interpolation formula (if needed)

$$D_{\max} = D_1 + \frac{camera\_distance - d_1}{d_2 - d_1} * (D_2 - D_1) \tag{24}$$

where,

- d_1 and d_2 are the closest camera distances.
- D_2 and D_1 are the corresponding max depths.

Step 3: Compare apples depth d with the threshold D_{max}

If $d \leq D_{max}$ then the Apple is in Foreground otherwise Apple is in background.

3.4 Loss Function

In the proposed DeepHarvestNet architecture, precise localization is essential due to the complex orchard conditions involving occlusions, dense clustering, and varying scales of apple fruits. To enhance the model's sensitivity toward these challenging scenarios, we adopt the Focal Intersection-over-Union (FIoU) loss function in the training process. Unlike conventional IoU-based loss functions that assign equal importance across all samples, FIoU introduces a compound formulation that applies a focal modulation combined with a logarithmic transformation of the IoU score. This design emphasizes difficult-to-predict cases by assigning greater penalties to poorly aligned bounding boxes. As a result, the network is encouraged to prioritize the correction of harder instances, such as small-sized apples, partially occluded fruits, and overlapping clusters, which frequently occur in natural orchard environments.

The FIoU loss function used in defined as:

$$\text{Focal IoU } = \left(1 - \frac{\text{Bp} \cap \text{Bg}}{\text{Bp} \cup \text{Bg}}\right)^{\gamma} * \log \frac{(\text{Bp} \cap \text{Bg})}{(\text{Bp} \cup \text{Bg})} \tag{25}$$

where,

- Bp and Bg denote the predicted and ground truth bounding boxes, respectively.
- Bp∩Bg represents the intersection area between the predicted and ground truth bounding boxes.
- Bp∪Bg represents the union area of the predicted and ground truth bounding boxes.

4 Experimental Results and Discussion

4.1 Experimental Platform

The suggested DeepHarvestNet framework for apple detection and depth estimation was developed and evaluated on a workstation running Ubuntu 16.04 LTS. The deployment was done using Python as the programming language and PyTorch deep learning framework, and it was possible to train high-performance models and host them. The hardware system involved an Intel Core i7 processor, 24 GB of RAM, and an NVIDIA GeForce RTX 3090 graphics processing unit with a 384-bit memory interface and a base frequency of 1395 MHz with the ability for parallel computation at a high rate.

To accelerate the training time, the system employed CUDA Toolkit and cuDNN (CUDA Deep Neural Network Library) to boost GPU efficiency for deep learning processes. DeepHarvestNet architecture enhances the baseline YOLOv8 model through the integration of Efficient Bidirectional Cross-Attention, Focal Modulation, KernelWarehouse Convolution, and AdaBins to improve feature representation and depth-aware detection. Experiments were performed at an Intersection over Union (IoU) threshold of 0.75 to compare bounding box prediction accuracy.

4.2 Evaluation Metrics

The performance of the recommended DeepHarvestNet model is evaluated quantitatively on three typical metrics: Precision (P), Recall (R), and F1-Score, which in combination offer a thorough estimate, particularly under orchard-specific conditions such as class imbalance, occlusion, and overlapping apples.

Precision (P): Measures the proportion of correctly predicted apples among all predicted positives, reflecting the model's ability to minimize false positives:

$$\text{Precision } = \frac{TP}{FP + TP} \tag{26}$$

Recall (R): Assesses the proportion of actual apples correctly detected, indicating the model's capability to minimize false negatives:

$$\text{Recall } = \frac{TP}{TP + FN} \tag{27}$$

F1-Score: Represents the harmonic mean of Precision and Recall, offering a balanced measure of detection performance

$$\text{F1-Score} = 2 * \left(\frac{\text{Precison } * \text{ Recall}}{\text{Precision } + \text{ Recall}} \right) \tag{28}$$

Here, TP (True Positives) are correctly identified apples, FP (False Positives) are non-apples misclassified as apples, and FN (False Negatives) are apples missed by the model.

These metrics collectively provide a robust evaluation of detection accuracy and robustness in challenging orchard environments.

4.3 Comparative Evaluation

To assess the performance of the proposed DeepHarvestNet architecture, a set of experiments was carried out, with the outcomes consolidated in Table 2. The model's effectiveness was benchmarked against multiple well-known object detection frameworks, such as Faster R-CNN, SSD, RetinaNet, YOLOv5, YOLOv7, and YOLOv8. Additionally, a comparison was made with an improved variant of YOLOv7 that integrates Attribute Attention Mechanism and Adaptive Pooling.

Table 2. Key parameters of our model

	Original Images			**Illumination Images**		
Detection Methods	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Faster R-CNN	0.75	0.81	0.76	0.69	0.71	0.70
SSD	0.81	0.85	0.83	0.70	0.77	0.72
RetinaNet	0.86	0.89	0.87	0.80	0.84	0.82
YOLOv5	0.91	0.93	0.92	0.82	0.9	0.83
YOLOv7	0.91	0.92	0.91	0.81	0.89	0.81
Improved YOLOv7 (Attribute Attention + Adaptive Pooling)	0.93	0.96	0.94	0.84	0.92	0.85
YOLOv8	0.94	0.96	0.95	0.89	0.92	0.90
DeepHarvestNet	0.96	0.97	0.97	0.94	0.95	0.95

The development of YOLO family by improvising each version with an adding new module that contribute the better fruit detection performance. The key changes in the YOLO versions allow to handle occlusion, lighting variation, multi-scale feature extraction, overlapping and depth aware. YOLOv4 showed an advancement by integrating Cross-Stage Partial, Spatial Pyramid Pooling and Mish activation funcitons. By adding these components helps in improving the feature extraction and gradient flow, enabling the model to handle challenging lighting in an orchard. Building on this, the YOLOv5 introduce a lightweight deployment, mosaic data augmentation to enhance detection of fruit in balancing the speed and accuracy.The model can be improvised by incorporating modules like CBAM attention and specific improvements like SPPFCSPC in handling the robust detection under occlusion, small or blur fruits and overlapping fruits. Further the YOLOv7 is enhanced in adoption of Extended Efficient Layer Aggregation Network, re-parameterization and planned gradient path allocation. These improvements help in handiling the better fruit detection under occlusion and improved feature reuse. Most recently, YOLOv8 model contributed better localization and handle the occulsion, overlapping by incorporating anchor-free detection head, decoupled classification and adaptive loss functions resulting in precise localization and detection accuracy under complex orchards. With an enhancement to the model, depth information can be possible with 3D localization. Collectively, these continous advancements in YOLO versions i.e. from YOLOv4 to YOLOv8 have increasingly enhanced the speed and accuracy of fruit detection under occlusion, overlapping fruits, variable lighting and small targets and depth information. These YOLO version models are more suitable for automated fruit monitoring in an orchard with bettter performance.

The comparison was done with uniform conditions for all the models, so there is a fair and unbiased comparison. Detection performance was measured in terms of metrics such as Precision, Recall, and F1-Score.The results clearly show how DeepHarvestNet improves upon introducing its advancements, showcasing its better capability to accurately localize and detect apples, particularly in challenging cases with occlusion, different sizes of fruits, and cluttered backgrounds. The extensive comparison of all the models, including the proposed model, is given in Table 2.

From the performance comparison in Table 2, the DeepHarvestNet model outperforms all the current detection methods under both original and illumination-changed image conditions. For original images, DeepHarvestNet has the highest Precision, Recall and F1-Score values of 0.96, 0.97 and 0.97 respectively, reflecting its high accuracy in detecting apples under normal conditions. Even in difficult illumination fluctuations, the model sustains strong performance, achieving 0.94 Precision, 0.95 Recall and 0.95 F1-Score, better than the next best model (YOLOv8) in all the metrics. The results demonstrate DeepHarvestNet capability to deal well with occlusion, diverse fruit sizes, and inconsistent lighting, which were weaknesses in earlier methods such as Faster R-CNN, SSD, and even earlier YOLO versions.

To better visualize and interpret this comparative performance, a bar plot representation of the metrics in Table 2 can be used. This visual depicted in Figure 9 aid helps in clearly identifying performance differences across models for both original and illumination-affected images, providing a quick and intuitive understanding of DeepHarvestNet improvements over the baseline and advanced detection frameworks.

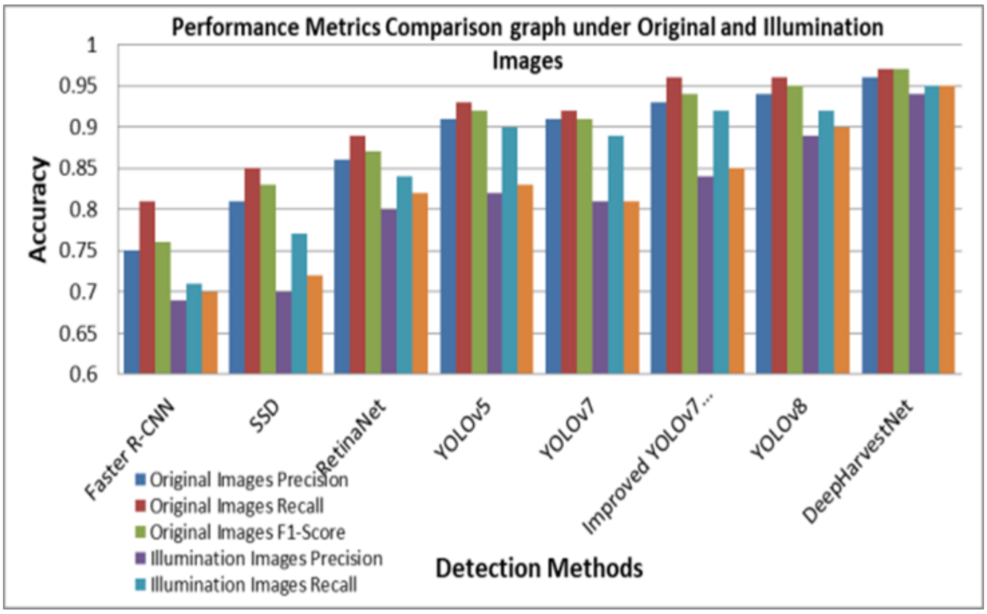
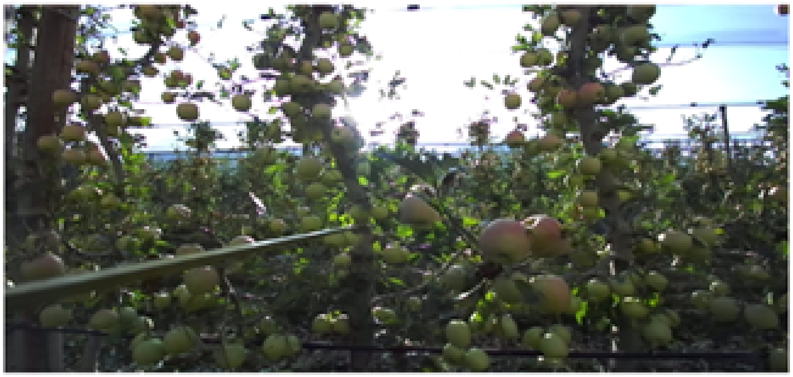
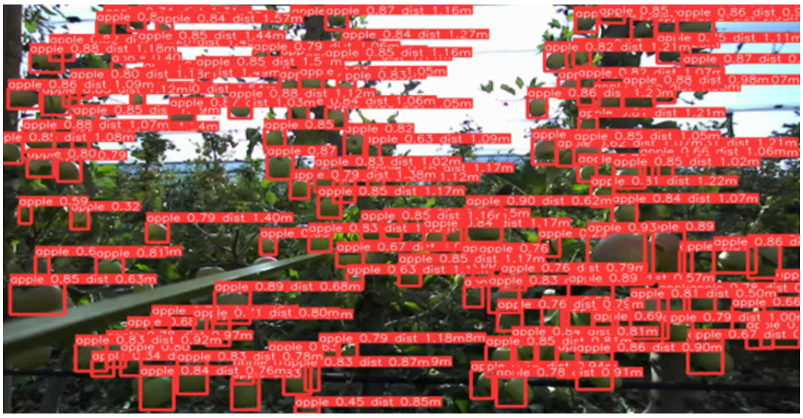


Figure 9. Comparative bar plot showing Precision, Recall and F1-Score for various object detection models on original orchard images and under illumination variation conditions



(a)



(b)

Figure 10. (a) Apple (gold delicious), (b) Apple (gold delicious) detected image by DeepHarvestNet

The Figure 10 shows the output of gold delicious with improved detection accuracy and followed by depth estimation using DeepHarvestNet architecture.

4.4 Discussion

The DeepHarvestNet framework was developed in direct response to the limitations identified in the Improved YOLOv7 model, which employed YOLOv7 with attribute attention and adaptive pooling. While that approach performed reasonably under ideal conditions, it struggled significantly with occlusion, overlapping fruits, and detection of small or distant apples. Specifically, the attribute attention failed to focus effectively under dense canopy cover, and fixed pooling layers often led to loss of detail for tiny or clustered apples. These shortcomings led to false positives and inconsistent localization in complex orchard scenarios. The current work addresses these issues by introducing targeted architectural improvements in the form of EBCA, FM, KWConv, and AdaBins within the YOLOv8 backbone, each selected to overcome a specific shortcoming from Improved YOLOv7.

The DeepHarvestNet framework that is submitted demonstrates considerable enhancement in apple detection and depth estimation through four essential components: Efficient Bidirectional Cross-Attention, Focal Modulation, KernelWarehouse Convolution, and AdaBins. Each of them tackles important shortfalls from previous models. EBCA facilitates efficient information interaction among multi-scale features by top-down and bottom-up attention, enabling the model to more accurately separate foreground apples from overlapping background fruits—a problem salient in previous methods. FM substitutes normal self-attention with context-sensitive modulation, selectively highlighting visible fruit areas while minimizing background noise, thereby improving detection even when over 70% of the fruit is occluded. KWConv improves feature fusion by adaptively adjusting the receptive field to the fruit size so that small or far-away apples are preserved and precisely localized.

Simultaneously, AdaBins enhances the depth estimation module by dynamically segmenting depth ranges into data-driven bins and allowing for fine-grained, pixel-wise depth prediction. Such adaptive binning addresses depth ambiguity in dense orchard scenes and allows for sharp separation of the foreground apples from the background. These modules combined provide a unified solution—enhancing detection accuracy as well as spatial comprehension. The model ably handles issues like fruit grouping, scale variability, and occlusion, thus making DeepHarvestNet a resilient architecture specific to real-world orchard settings, particularly for applications like automatic harvesting and precision crop monitoring.

To compare the detection performance of DeepHarvestNet, we performed a comparative analysis with classical and state-of-the-art fruit detection models, as shown in Figure 11. The proposed DeepHarvestNet had the best detection accuracy of 95.0%, surpassing recent state-of-the-art architectures like O2RNet (94.0%) and YOLO11 with Vision Transformers (94.0%). These models were tailored to solve occlusion and grouped fruit detection, but DeepHarvestNet proved better performance because of the incorporated architectural modules—Efficient Bidirectional Cross-Attention (EBCA) for separating overlaps, Focal Modulation for feature extraction resilient to occlusion, and KernelWarehouse Convolution with AdaBins for depth-guided, scale-aware localization.

In comparison with slightly older but still applicable techniques like YOLOv7 with attention (91.5%) and multi-scale YOLOv5 (89.2%), DeepHarvestNet achieves a dramatic improvement. This performance difference signifies the shortcomings of existing models in dealing with dense fruit structures and spatial ambiguity. Interestingly, traditional models such as Faster R-CNN utilized in Bargoti and Underwood [1] and Sa et al. [2] had much lower accuracies of 86.3% and 84.7%, respectively, due to difficulties in keeping pace with contemporary orchard complexity without attention or depth-aware modules. These outcomes confirm that DeepHarvestNet not only performs state-of-the-art accuracy but also shows stable robustness under actual orchard conditions and therefore is highly appropriate for precision agriculture use cases like robotic harvesting and extensive orchard monitoring.

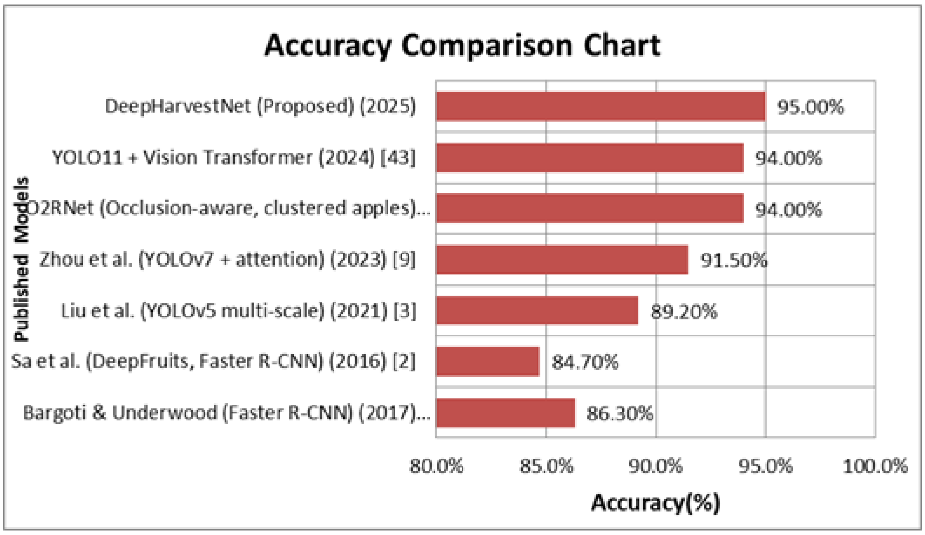


Figure 11. Comparative analysis of detection accuracy between DeepHarvestNet and state-of-the-art fruit detection models in orchard settings

5 Conclusion and Future Scope

The current research proposed DeepHarvestNet, a dedicated apple fruit detection and depth estimation model optimized to tackle the vital issues of occlusion, overlap, and scale variance in intricate orchard scenes. By implementing Efficient Bidirectional Cross-Attention, Focal Modulation, and KernelWarehouse Convolution in the YOLOv8 backbone, the developed model gains better spatial localization and resilience, especially in detecting heavily clustered and partially occluded apples. In addition, the AdaBins module integration allows for precise monocular depth estimation by adaptively estimating scene geometry, successfully separating foreground apples from background objects. The experimental results of achieving a Precision of 0.94, Recall of 0.95, and F1-Score of 0.95 show that DeepHarvestNet surpasses baseline and state-of-the-art models consistently under normal and difficult light conditions. The shown effectiveness and robustness of this method point to its high prospects for implementation in precision agriculture tasks, such as autonomous apple harvesting and orchard monitoring systems.

In future research, adding a real-time tracking module like ByteTrack or DeepSORT would allow for ongoing fruit counting and production estimation over time. Moreover, adding the ability to detect multiple classes of fruits and grade ripeness can enhance its use across different orchard conditions.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] S. Bargoti and J. P. Underwood, “Deep fruit detection in orchards,” *IEEE Int. Conf. Robot. Autom.*, vol. 2017, pp. 3626–3633, 2017. https://doi.org/10.1109/ICRA.2017.7989417

[2] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. McCool, “DeepFruits: A fruit detection system using deep neural networks,” *Sens.*, vol. 16, no. 8, p. 1222, 2016. https://doi.org/10.3390/s16081222

[3] K. Lin, L. Gong, Y. Huang, C. Liu, and J. Pan, “Deep learning-based segmentation and quantification of cucumber flowers and fruits in a plant factory,” *Front. Plant Sci.*, vol. 10, p. 919, 2019. https://doi.org/10.3389/fpls.2019.00155

[4] P. Chu, Z. Li, K. Lammers, R. Lu, and X. Liu, “Deep learning-based apple detection using a suppression mask R-CNN,” *Pattern Recognit. Lett.*, vol. 147, pp. 206–211, 2021. https://doi.org/10.1016/j.patrec.2021.04.022

[5] P. Siricharoen, W. Yomsatieankul, and T. Bunsri, “Fruit maturity grading framework for small dataset using single image multi-object sampling and mask R-CNN,” *Smart Agric. Technol.*, vol. 3, p. 100130, 2023. https://doi.org/10.1016/j.atech.2022.100130

[6] A. Janowski, R. Kaźmierczak, C. Kowalczyk, and J. Szulwic, “Detecting apples in the wild: Potential for harvest quantity estimation,” *Sustainability*, vol. 13, no. 14, p. 8054, 2021. https://doi.org/10.3390/su13148054

[7] A. Gongal, S. Amatya, M. Karkee, Q. Zhang, and K. Lewis, “Sensors and systems for fruit detection and localization: A review,” *Comput. Electron. Agric.*, vol. 116, pp. 8–19, 2015. https://doi.org/10.1016/j.compag.2015.05.021

[8] Y. Xue, N. Huang, S. Tu, L. Mao, A. Yang, and X. Zhu, “Immature mango detection based on improved YOLOv2,” *Trans. Chin. Soc. Agric. Eng.*, vol. 34, no. 7, pp. 173–179, 2018. https://doi.org/10.11975/j.issn.1002-6819.2018.07.022

[9] Y. Tian, G. Yang, Z. Wang, H. Wang, E. Li, and Z. Liang, “Apple detection during different growth stages in orchards using the improved YOLO-V3 model,” *Comput. Electron. Agric.*, vol. 157, pp. 417–426, 2019. https://doi.org/10.1016/j.compag.2019.01.012

[10] A. I. B. Parico and T. Ahamed, “Real time pear fruit detection and counting using YOLOv4 models and deep SORT,” *Sens.*, vol. 21, no. 14, p. 4803, 2021. https://doi.org/10.3390/s21144803

[11] Z. Wang, L. Jin, S. Wang, and H. Xu, “Apple stem/calyx real-time recognition using YOLO-V5 algorithm for fruit automatic loading system,” *Postharvest Biol. Technol.*, vol. 185, p. 111808, 2022. https://doi.org/10.1016/j.postharvbio.2021.111808

[12] L. Ma, L. Zhao, Z. Wang, J. Zhang, and G. Chen, “Detection and counting of small target apples under complicated environments by using improved YOLOv7-Tiny,” *Agronomy*, vol. 13, no. 5, p. 1419, 2023. https://doi.org/10.3390/agronomy13051419

[13] Y. Liu, X. Han, H. Zhang, S. Liu, W. Ma, Y. Yan, L. Sun, L. Jing, Y. Wang, and J. Wang, “YOLOv8-MSP-PD: A lightweight YOLOv8-based detection method for Jinxiu malus fruit in field conditions,” *Agronomy*, vol. 15, no. 7, p. 1581, 2025. https://doi.org/10.3390/agronomy15071581

[14] G. Li, X. Huang, J. Ai, Z. Yi, and W. Xie, “Lemon-YOLO: An efficient object detection method for lemons in the natural environment,” *IET Image Process.*, vol. 15, no. 9, pp. 1998–2009, 2021. https://doi.org/10.1049/ipr2.12171

[15] Y. Lin, Z. Huang, Y. Liang, Y. Liu, and W. Jiang, “AG-YOLO: A rapid citrus fruit detection algorithm with global context fusion,” *Agriculture*, vol. 14, no. 1, p. 114, 2024. https://doi.org/10.3390/agriculture14010114

[16] Y. Yu, Y. Liu, Y. Li, C. Xu, and Y. Li, “Object detection algorithm for citrus fruits based on improved YOLOv5 model,” *Agriculture*, vol. 14, no. 10, p. 1798, 2024. https://doi.org/10.3390/agriculture14101798

[17] A. Uriti and N. J. Pothabathula, “A deep learning-based object detection approach enhanced with attention mechanism in YOLO,” *AIP Conf. Proc.*, vol. 3099, p. 020020, 2025. https://doi.org/10.1063/5.0279360

[18] Z. Zhao, J. Wang, and H. Zhao, “Research on apple recognition algorithm in complex orchard environment based on deep learning,” *Sensor*, vol. 23, no. 12, p. 5425, 2023. https://doi.org/10.3390/s23125425

[19] J. Liu, C. Wang, and J. Xing, “YOLOv5-ACS: Improved model for apple detection and positioning in apple forests in complex scenes,” *Forests*, vol. 14, no. 12, p. 2304, 2023. https://doi.org/10.3390/f14122304

[20] J. Zhao, C. Du, Y. Li, and et al., “YOLO-Granada: A lightweight attented YOLO for pomegranate fruit detection,” *Sci. Rep.*, vol. 14, p. 16848, 2024. https://doi.org/10.1038/s41598-024-67526-4

[21] E. Osco-Mamani, O. Santana-Carbajal, I. Chaparro-Cruz, D. Ochoa, and S. Alcazar-Alay, “The detection and counting of olive tree fruits using deep learning models in Tacna, Perú,” *AI*, vol. 6, no. 2, p. 25, 2025. https://doi.org/10.3390/ai6020025

[22] N. Häni, P. Roy, and V. Isler, “Apple counting using convolutional neural networks,” *arXiv Prepr.*, vol. arXiv:2208.11566, 2022. https://doi.org/10.48550/arXiv.2208.11566

[23] T. Fischer, X. Zhang, C. Li, and H. Chen, “QDTrack: Quasi-dense similarity learning for appearance-only multiple object tracking,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 12, pp. 15 380–15 393, 2023. https://doi.org/10.1109/TPAMI.2023.3301975

[24] I. K. Agung Enriko, E. L. Istikhomah Puspita Sari, I. A. Yamin, and N. Albar, “Detection of fresh and root apples using the TensorFlow lite framework with EfficientDet-Lite2,” *Int. J. Intell. Syst. Appl. Eng.*, vol. 12, no. 11s, pp. 566–569, 2024. https://doi.org/10.1109/IJISAE.2024.4477

[25] T. A. Oei, M. S. Wijaya, J. B. Simanjuntak, E. F. A. Sihotang, and E. Irwansyah, “Using YOLOv7 to detect the health and quality of fruits,” *EasyChair Prepr.*, vol. 10546, 2023.

[26] L. He, D. Wu, X. Zheng, F. Xu, S. Lin, S. Wang, F. Ni, and F. Zheng, “RLK-YOLOv8: Multi-stage detection of strawberry fruits throughout the full growth cycle in greenhouses based on large kernel convolutions and improved YOLOv8,” *Front. Plant Sci.*, vol. 16, p. 1552553, 2025. https://doi.org/10.3389/fpls.2025.1552553

[27] D. Wang, H. Song, and B. Wang, “YO-AFD: An improved YOLOv8-based deep learning approach for rapid and accurate apple flower detection,” *Front. Plant Sci.*, vol. 16, p. 1541266, 2025. https://doi.org/10.3389/fpls.2025.1541266

[28] J. Yang, C. Li, X. Dai, L. Yuan, and J. Gao, "Focal modulation networks," *arXiv Prepr.*, vol. arXiv:2203.11926, 2022. https://doi.org/10.48550/arXiv.2203.11926

[29] C. Li and A. Yao, "KernelWarehouse: Rethinking the design of dynamic convolution," *arXiv Prepr.*, vol. arXiv:2406.07879, 2024. https://doi.org/10.48550/arXiv.2406.07879

[30] S. F. Bhat, I. Alhashim, and P. Wonka, "AdaBins: Depth estimation using adaptive bins," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, 2021, pp. 4008–4017. https://doi.org/10.1109/CVPR46437.2021.00400

[31] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada, 2021, pp. 12 159–12 168. https://doi.org/10.1109/ICCV48922.2021.01196

[32] C. Y. Wang, A. Bochkovskiy, and H. Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, BC, Canada, 2023, pp. 7464–7475. https://doi.org/10.1109/CVPR52729.2023.00721

[33] R. Varghese and S. M., "YOLOv8: A novel object detection algorithm with enhanced performance and robustness," in *Proceedings of the International Conference on Advanced Data Engineering and Intelligent Computing Systems (ADICS)*, Chennai, India, 2024, pp. 1–6. https://doi.org/10.1109/ADICS58448.2024.10533619

[34] Z. Tu, H. Talebi, H. Zhang, F. Yang, P. Milanfar, A. Bovik, and Y. Li, "MaxVit: Multi-Axis vision transformer," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022, pp. 459–479. https://doi.org/10.1007/978-3-031-20053-3_27

[35] C. Godard, O. M. Aodha, M. Firman, and G. Brostow, "Digging into self-supervised monocular depth estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), 2019, pp. 3827–3837. https://doi.org/10.1109/ICCV.2019.00393

[36] X. Ding and et al., "ResRep: Lossless CNN pruning via decoupling remembering and forgetting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada, 2021, pp. 4490–4500. https://doi.org/10.1109/ICCV48922.2021.00447

[37] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, 2022, pp. 11 966–11 976. https://doi.org/10.1109/CVPR52688.2022.01167

[38] A. Uriti and N. J. Pothabathula, "Evaluating object detection approaches for fruit detection in precision agriculture: A comprehensive review," in *Proceedings of the International Conference on Intelligent Computing and Sustainable Innovative Technologies (IC-SIT)*, Bhubaneswar, India, 2024, pp. 1–6. https://doi.org/10.1109/IC-SIT63503.2024.10862633

[39] L. Zhu, X. Wang, Z. Ke, W. Zhang, and R. Lau, "BiFormer: Vision transformer with Bi-Level routing attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, BC, Canada, 2023, pp. 10 323–10 333. https://doi.org/10.1109/CVPR52729.2023.00995

[40] P. Chu, Z. Li, K. Zhang, D. Chen, K. Lammers, and R. Lu, "O2RNet: Occluder-occludee relational network for robust apple detection in clustered orchard environments," *Smart Agric. Technol.*, vol. 5, p. 100284, 2023. https://doi.org/10.1016/j.atech.2023.100284