Multiclass Fruit Detection Using Improved YOLOv3 Algorithm

Seema C. Shrawne, Jay Sawant, Omkar Chaubal, Karan Suryawanshi, Diven Sirwani, Vijay K. Sambhe
Department of CE and IT
Veermata Jijabai Technological Institute, H. R. Mahajani Marg, Matunga, Mumbai 400019.

Abstract—**Manual interventions continue to be used in fruit-picking and billing at large-scale fruit storage facilities. Recent advances in deep in learning approaches, such as one-stage detectors like You Only Look Once (YOLO) and Single Stage Detector (SSD), as well as two-stage detectors like Faster RCNN and Mask RCNN, aim to streamline the processes involved with fruit detection and enhance efficiency. However, these frameworks continue to suffer with multi-scale objects, in terms of performance and efficiency due to large parameter sizes. These problems increase when multi-class fruits are encountered. We propose an improved version of the one-stage detector framework YOLOv3 for multi-class fruit detection. Our proposed model addresses the challenges of multi-scale object detection and detection of different fruit types in an image by incorporating CNN, bottleneck, and Spatial Pyramid Pooling Fast (SPPF) modules in the Head, Neck, and custom backbone of the YOLOv3 framework. Optimization of learnable parameters for computational efficiency is achieved by concatenating features at different feature map resolutions. The proposed model incorporates fewer layers and parameters compared to YOLOv3 and YOLOv5 models. We performed extensive testing on three datasets downloaded from Roboflow and compared them with YOLOv3 and YOLOv5 models. Our model achieved mAP50 of 0.747 on Dataset 1 comprising images of apples, bananas, and oranges whereas Dataset 2 consisting of images of apples, oranges, lemon, and Pear, achieved mAP50 of 0.981. Testing the Mineapple dataset comprising on-tree images of apples of varied sizes, achieved an accuracy of 0.643. We observe that the performance of our model beats the performance of the YOLOv3 and YOLOv5 models.**

Keywords*—*Precision agriculture; yield estimation; fruit detection; YOLOv3; feature concatenation; spatial contexting

I. Introduction

With the growing population, providing food security is of utmost concern. Precision Agriculture comprises methods to optimize resources by automation of agricultural tasks like sowing, weeding, spraying, and harvesting driven by technology which helps in increasing food production [1]. Before harvesting, Yield estimation is necessary to avoid post-harvest losses of fruits caused by harvesting raw or overripe fruits [2], [3]. Accurate counting and effective invoicing of these fruits during harvest [12] and their storage in the warehouse are critical to orchard profitability. However, the current method of counting fruits, which involves physical labor, takes a long time, is prone to error, cannot keep up with the volume, and is negative to schedule management. Orchards with multiple types of fruits pose a challenge. Automating this process with robots is a viable answer [15], but these robots need powerful computer vision systems to detect and locate the different types of fruits in the orchard and warehouse environments. Adding to this, challenges, including various fruit sizes, colors, and dense foliage, make detection more difficult. The key feature of object identification is central to this vision system, allowing the robots to discern between different fruit types and perform appropriate picking and billing activities.

Fruit Detection models designed by many researchers prioritize certain properties of fruits to improve accuracy through specialized methods. Commonly used detectors include Mask RCNN[9], Faster RCNN[8], and different versions of YOLO (3, 4, 5, and 8), along with DenseNets and ResNets[11] as feature extractors. Visual attributes like color, texture, shape, and size are important properties for recognizing fruits. across different growth stages need to be considered to differentiate between fruit types. Detection of different types and sizes of fruits in an image is a challenging problem. Detecting inter-class similarities and intra-class variations is possible by a combination of low-level features and high-level semantics. In this study, we propose a fruit detection model with a custom backbone network for feature extraction at multiple levels. The YOLOv3 algorithm caught our attention, particularly through its simplicity coupled with precision without compromising speed. YOLOv3 is often used as a base model for modifications leading to continuous improvement, such as in [11], and has relatively lower training times to help achieve this. These reasons inspired us to make good use of it to develop an advanced one-stage fruit detection model. While deep convolutional networks have shown promise in fruit detection, we identified key challenges that serve as the primary objectives this research aims to solve: 1. Creation of lightweight models for practical use. 2. Effectively handling objects of different scales. 3. Achieving strong performance while maintaining efficiency rampant among fruit detection studies. 4. Training the model successfully on images such that each image has objects of different classes.

We decided to achieve this by constructing our own variant of the YOLOv3 [4], one that would take on all four challenges while providing much better results. Our proposed model addresses these challenges by incorporating special modules and optimizing parameters for computational efficiency. The model shall work on both single class as well as multi class fruit detection. We utilized three key datasets, one of which is a benchmark dataset, detailed further in Part C of the Methodology section. Detailed explanations of our novel methods are provided in the Proposed System section in this paper, showcasing our contribution to advancing fruit detection technology, while the next subsection is a tiny yet precise gist of why we chose our system in the first place.

A. Our Contributions

1. We modified the existing Darknet-53 Backbone of the YOLOv3 model by including multi-scale feature extraction and then arranging bottleneck layers to reduce the dimensionality of feature maps making the network more computationally efficient. In the head part of the model, high-level semantic features are concatenated with low-level details so that fruits of different sizes can be detected. With these modifications, our model can now facilitate the acquisition of more discriminative features by allowing gradients to flow directly during training. It also solves the challenge of detecting fruits of varying sizes.

2. Our model is trained on three datasets to prove effectiveness: Dataset 1 and Dataset 2, which consist of images of different fruits (mixed fruits), and Dataset 3, comprising of the Benchmark Mineapple Dataset consisting of apples in a dense orchard environment.

3. We have evaluated the model using various Performance metrics: Precision, Recall, mAP@50 & mAP@90 and then compared the results with YOLOv3 and YOLOv5l. Notably, running our model on Dataset 1 produced a mAP@50 of 0.747 , 0.981 on Dataset 2 and 0.643 on Dataset 3. The model achieved a higher mAP@50 on Dataset1 and Dataset3 and a higher Recall on all 3 datasets when compared with YOLOv3 and YOLOv5l models.

4. While designing the Backbone network, care was taken that the number of layers and number of parameters in our model are less then those in the standard YOLOv3 model and YOLOv5l model.

B. Organization of the Paper

The paper is organized as follows : The Section II is a detailed survey of existing research in the field, followed by Section III comprising of the dataset characteristics, our proposed system and subsequent model training. In Section IV are the results and discussions followed by the Conclusion.

II. Related Work

Before delving into our own model, we explored various research contributions, each shedding light on distinct advancements in real-time fruit detection. The study in [5] utilizes the YOLOv4 neural network to enhance real-time banana recognition in complex orchards. It addresses similarity, occlusion, and uncertainties by extracting complex features. The model, based on YOLOv4 with CSPDarknet53, includes the FPN+PAN module, SPP module, and Mish activation function. The DIOU_nms algorithm improves detection confidence. Comparisons show YOLOv4 surpasses YOLOv3 and traditional methods in accuracy and speed, with an average execution time of 0.171s and a detection rate of 99.29%. The YOLOMuskmelon model /citec2, blends speed and accuracy for enhanced fruit detection. It features a ResNet43 backbone with ReLU activation, SPP for improved accuracy, FPN for efficient feature extraction, and DIoU NMS for efficiency. With an AP of 89.6%, it outperforms YOLOv3 and YOLOResNet50 but slightly lags YOLOv4 at 91.6%. Notably, it operates at 96.3 fps, faster than YOLOv3, YOLOv4, and YOLOResNet50, highlighting its potential for real-time fruit harvesting robots due to its speed advantage over YOLOv4. A bottleneck network module C2f is the building block in the YOLOv8 model for feature extraction. In study [7] YOLOv8 model achieved mAP50 of 99.5% for ripeness detection of apples and pears. Results were compared with CenterNet model with ResNet50 backbone. A light-weight model based on YOLOv5 for real-time applications is proposed in study [8] to detect strawberry fruits. A detection speed of 7.30ms and average precision 89.7% is reported. An attention module integrated with YOLOv7 in study [9] to detect kiwi fruits. Channel and spatial features extracted by the attention module improve the accuracy of detecting small and overlapping fruits. Comparison results of YOLOv8 and Mask RCNN in [9]show that YOLOv8 is better than Mask RCNN in terms of accuracy and speed. An experiment to detect objects of two types, trunks and apple tree branches, and next to detect green apples in an orchard environment confirmed the suitability of YOLOv8 for real-time detection for applications in robotic harvesting. To enhance real-time fruit recognition speed and accuracy, [11] introduces YOLOv5s. It targets applications on low-power devices and fruit-harvesting robots [16]. Adjustments to the backbone network, adaptive image scaling, and computed anchor boxes were made using a dataset of 1,350 strawberry and 1,959 jujube photos. Improvements like Stem, AC, Maxpool, CBS, SPPF, and CAM enhance adaptability to low-power devices. Validation and test results show mAP values of 93.4% and 96.0%, respectively. Operating at 74 fps on videos, YOLOv5s outperforms models like YOLOv4-tiny, YOLOv7-tiny, and GhostYOLOv5s in robustness and efficiency. This study [12] introduces a multi-cluster green persimmon recognition approach using an enhanced Faster RCNN model. It utilizes a dataset of 9,300 images captured under diverse natural light conditions, including scenarios with multiple fruits, leaf shadows, and overlapping clusters. The upgraded model integrates a weighted ECA mechanism into three key feature layers and enhances the DetNet feature extractor to balance information levels. It incorporates multi-scale features and employs K-means clustering for bounding box clustering and anchoring. Achieving a mAP of 98.4%, the model surpasses the traditional Faster RCNN by 11.8%, demonstrating significant improvements in identifying green persimmons, especially in complex and obscure environments. An input comprising of RGB and HSV images of Oranges fed to MaskRCNN improved segmentation accuracy in [13] Next, [14] uses the VGG16 architecture and Faster R-CNN model to detect kiwifruits in orchard photos under varied lighting and time conditions. With a dataset of 2400 images at 2352x1568 resolution, each containing at least 30 kiwifruits, the model excels in recognizing kiwifruits despite occlusion, overlap, and lighting variations. Outperforming ZFNet, it proves effective in dynamic agricultural settings, ensuring high accuracy and minimal false negatives in kiwifruit identification.

In this study [11], authors enhance the YOLOv3 model for automated oil palm loose fruit identification, integrating DenseNet for feature reuse, swish activation, and multi-scale detection to improve small object accuracy. Diversifying a dataset from 700 UAV and mobile camera photos to 6300 images, they boost model performance across detection metrics. Outperforming YOLOv3, Faster R-CNN-ResNet101, YOLOv3 tiny, YOLOv2, and SSD-MobileNet, the model achieves superior average precision, overlap metrics, and F1-score while maintaining computational efficiency.

A study [12] introduces a real-time olive fruit detection system using advanced deep learning frameworks like YOLOv5x, YOLOv5s, and YOLOR. YOLOv5s combines YOLO Layer, PANet, and CSPDarknet for detection. With 40,834 annotated olive fruit images, the system achieves 62 FPS detection speed and the highest 0.75 mAP_0.5 precision, making YOLOv5s the optimal choice for automating olive harvesting challenges. The project [17] applies deep learning to enhance papaya recognition in natural orchard settings. The YOLOv5s-Papaya model integrates bidirectional weighted feature pyramid network, Ghost module, and coordinate attention module for dense multitarget detection. Utilizing mosaic data augmentation, adaptive anchor computation, and PANet framework ensures multiscale feature fusion. With 1,000 diverse photos, the model achieves 92.3% average precision, 83.4% recall, and 90.4% precision, surpassing previous YOLO versions. The working of Two-stage Detectors and YOLO architecture and its successors have been reviewed in [18] A study [19] extensively examines the YOLO series, assessing their designs, regression methods, and performance on MSCOCO and Pascal VOC datasets. YOLO models demonstrate superior detection accuracy and speed compared to two-stage detectors like RCNN, Fast-RCNN, and Faster-RCNN, making them ideal for real-time applications in machine learning and deep learning tasks. An attention mechanism to improve the localization of fruits is introduced in YOLOv5 architecture between the backbone and neck region in [20]. Results on fruits like apple, oranges, grapes on state-of-the-art models show that the proposed model has a better target detection and generalization ability.The bidirectional attention mechanism extracts features from horizontal and vertical directions, assigns weights followed by concatenation. The paper in study [21] presents the GCS-YOLOV4-Tiny model, which enhances the YOLOV4-Tiny architecture for faster fruit detection by integrating spatial pyramid pooling (SPP), squeeze and excitation (SE) modules, and group convolution. Evaluations on Mango YOLO, Rpi-Tomato, and F. margarita datasets demonstrate significant improvements over YOLOV4-Tiny, achieving a 17.45% increase in mean average precision and a 13.80% rise in F1-score. These enhancements optimize both accuracy and speed in fruit detection tasks. In [22], authors have proposed a system for selecting image regions based on features like LBP, HOG, color histograms, and shape features with a weighted score for combining features. Improvement of region proposals based on Edgeboxes are proposed. A dataset of 18,155 images like apples, pears, kiwis, and persimmon are trained on the system and then compared with DPM (Deformable Parts Model), CNN with SVM and Faster RCNN.A detection rate of 0.9632 and a FPPI (False positive per image) of 0.0682 is reported. In this study, multiple types of fruits within the same image are included in the dataset. It is found that almost all studies have performed detection on images having fruits of single type within an image. The study on muticlass fruit detection with multiple fruits within the same image is carried out in study [22] and study [10]. In study [22], a custom region selection method is proposed and has very good accuracy when compared with Faster RCNN and other models. Faster RCNN F1 score is also nearly equal to the new method. The images are multiclass but fruit instances are not overlapping or together. In study [10] multiple types of classes like trunk and branches are detected in an image.

III. Datasets and Methodology

Extensive studies over the backbone network of YOLOv3 model and family of YOLO object detectors went into the initial part of our research. Architectural patterns in the detector were interpreted to understand how the detector captures various features. The next step was to examine existing research like those seen in sSection II to understand modifications in standard YOLOv3 and YOLOv5 models and their weak points which would grant us clarity as to what novel changes we could make to the single stage detector to overcome challenges with fruit detection in particular. This section is a detailed account of the technicalities behind a standard YOLOv3 detector and then the step by step working of our proposed system. We also formally introduce each dataset with its contents and their specifications.

A. YOLOV3 Model

YOLOv3 (You Only Look Once, Version 3) emerges as a pivotal advancement in real-time object detection, brought to fruition by the collaborative efforts of Joseph Redmon and Ali Farhadi. Building upon the foundations laid by YOLO and YOLOv2, YOLOv3 represents a significant leap forward in accuracy and speed, setting new benchmarks in the field. Released in 2018, this iteration refines its predecessors' successes while introducing key architectural intricacies that elevate its performance.At its core, YOLOv3 utilizes a variant of Darknet, originally a 53-layer network trained on ImageNet, which has now evolved into a 106-layer fully convolutional architecture tailored specifically for object detection tasks. This expanded architecture enables YOLOv3 to process images comprehensively, integrating global context during inference and improving its accuracy in detecting various objects.

The algorithm operates as a Convolutional Neural Network (CNN), drawing inspiration from ResNet and FPN architectures. Darknet-53, YOLOv3's feature extractor, incorporates skip connections and three prediction heads, facilitating spatial compression and precise detections. These architectural enhancements contribute significantly to YOLOv3's ability to detect objects accurately and efficiently in real-time scenarios.

In comparative evaluations against other popular frameworks like Faster R-CNN and MobileNet-SSD, YOLOv3 consistently demonstrates its superiority. It achieves a remarkable 37 mAP on the COCO-2017 validation set at 608x608 resolution, outpacing Faster R-CNN architectures while maintaining a significant speed advantage—17 times faster, to be exact. This speed-to-accuracy ratio positions YOLOv3 as a leading choice for real-time object detection tasks, especially in scenarios requiring rapid and precise identification of objects.

Anchored on concepts like anchor boxes, k-means clustering, and a meticulously designed network structure, YOLOv3 excels in detecting small targets with exceptional accuracy, making it a preferred solution for a wide range of applications demanding robust and efficient object detection capabilities.

1) Bounding Box Prediction in YOLOv3: In the context of YOLOv3's object detection technique, bounding box prediction entails generating bounding box attributes such as coordinates and size. This operation is facilitated by 1 x 1 detection kernels, which have the following shape: 1 x 1 x (B x (5 + C)). Here, B specifies the number of bounding boxes per cell, '5' represents the box attributes (x, y, width, height, confidence), and C denotes the number of classes. This method allows YOLOv3 to correctly forecast multiple bounding boxes for each object class in an image. Object confidence is an important factor in assessing if an object is present within a predicted bounding box. This confidence measure is generated using binary cross-entropy, which assesses the likelihood of an object appearing within a certain bounding box region. A higher object confidence score implies a larger possibility of the object's presence, whereas a lower score indicates a lower probability.

Bounding Boxes Prediction:
Shape: $1 \times 1 \times (B \times (5 + C))$
Where B represents the number of bounding boxes per cell, 5 denotes the box attributes (x, y, width, height, confidence), and C signifies the number of classes.
Object Confidence:
Object Confidence = $p(\text{Object}) \times \text{IoU}_{\text{pred}}^{\text{truth}}$

B. Proposed System

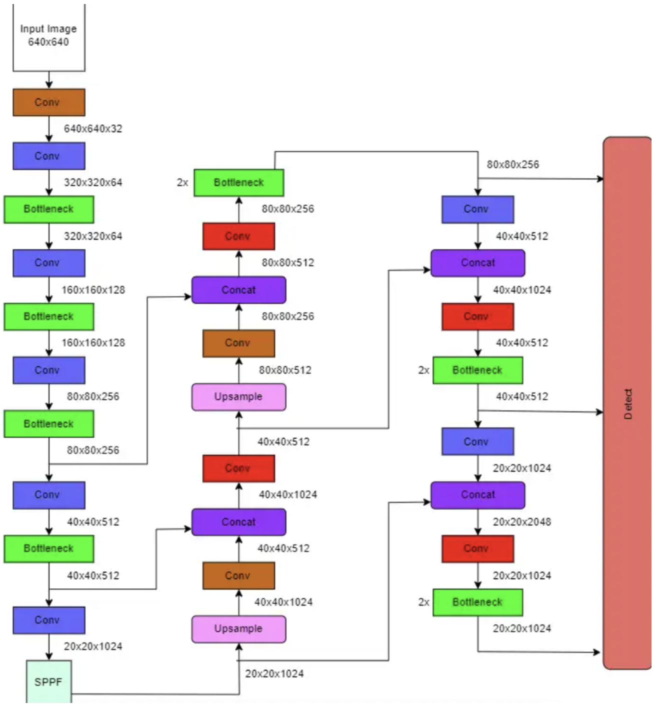


Fig. 1. Proposed system architecture.

The proposed system shown in Fig. 1 utilizes convolutional layers and bottleneck blocks in the backbone to extract hierarchical features, followed by a multiscale feature extraction process. In the head section, the system merges features from both the backbone and the multiscale extractor to enhance object detection capabilities. This integrated feature representation is then utilized for precise object localization and recognition, improving overall performance in detecting objects within images.

Our object detection system represents a sophisticated fusion of architectural components aimed at advancing object detection capabilities within deep learning frameworks. Once an image is passed, it undergoes resizing to 640 x 640 size and post this, enters the backbone of our network which contains a series of several convolutional layers, each strategically designed to capture intricate spatial patterns and hierarchical features from input feature maps.

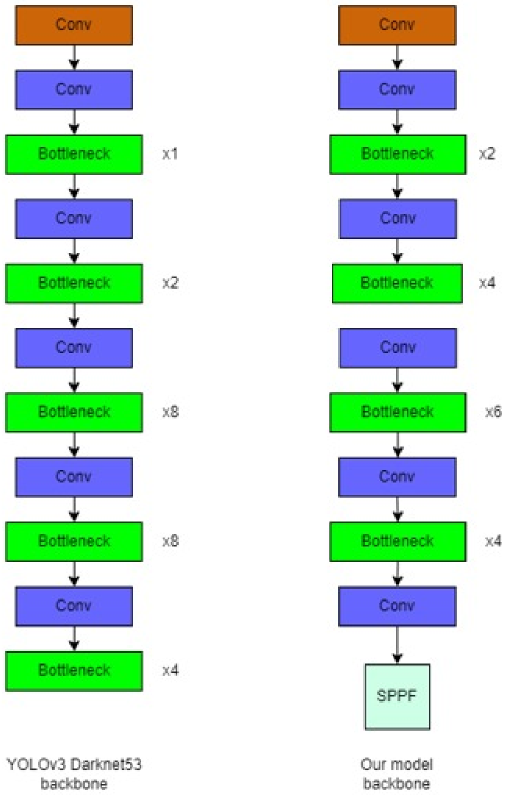


Fig. 2. Backbone of YOLOv3 and our model.

1) Custom Backbone: These convolutional layers are complemented by multiple bottleneck blocks, inspired by the modern architecture Darknet-53 as seen in Fig. 2. The bottleneck blocks in Fig. 4 play a crucial role in feature learning by incorporating residual connections, thereby facilitating the direct flow of gradients during training and mitigating the vanishing gradient problem. This approach not only enhances the model's ability to learn discriminative features but also reduces computational complexity, making the system more efficient and scalable, making it highly instrumental for deployment in resource-constrained environments like autonomous fruit-picking robots or edge computing devices in warehouses. What closely follows this is the integral part of the system's object detection prowess: an innovative inclusion of a multi-scale feature extractor called the SPPF (Spatial Pyramid Pooling Fast) layer. SPPF addresses the challenge of efficiently handling objects of varying sizes within input images. Traditional CNNs often require fixed-size input, limiting their efficacy in detecting objects at different scales. SPPF overcomes this limitation by enabling the network to operate on feature maps of arbitrary sizes, allowing for the detection of objects across multiple scales within the same image.

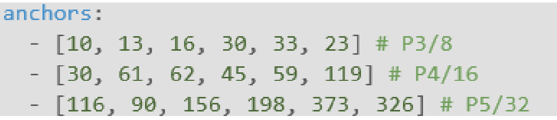
Through hierarchical feature fusion in the neck part via concatenation and multi-scale feature representation, the system achieves enhanced contextual understanding. This is crucial in a mixed fruit setting where for example the model must differentiate between a lemon and an apple. The SPPF layer's implementation of spatial pyramid pooling and factorization techniques contributes significantly to the system's efficiency, enabling it to handle variable input sizes while capturing multi-scale features effectively. What happens with this module is the network effectively partitions feature maps into progressively smaller segments. Segmentation enhances the model's ability to focus on and detect smaller objects by increasing both the resolution and receptive area, which is crucial in densely packed environments like orchards. This comprehensive architecture represents a substantial advancement in object detection methodologies, offering a scalable, efficient, and accurate solution for complex visual recognition tasks within the realm of computer vision and deep learning research.

2) Anchor Design Scheme: Anchors for different feature map sizes are shown in Fig. 3.

Anchors for feature map scale P3/8: These anchors are used with the feature map at a scale where the input image is downsampled by a factor of 8. For example, if the input image is 640x640, the feature map size would be 80x 80.

Anchors for feature map scale P4/16: These anchors are used with the feature map at a scale where the input image is downsampled by a factor of 16. For example, if the input image is 640x640, the feature map size would be 40x40.

Anchors for feature map scale P5/32: These anchors are used with the feature map at a scale where the input image is downsampled by a factor of 32. For example, if the input image is 640x640, the feature map size would be 20x20.



```
anchors:
  - [10, 13, 16, 30, 33, 23] # P3/8
  - [30, 61, 62, 45, 59, 119] # P4/16
  - [116, 90, 156, 198, 373, 326] # P5/32
```

Fig. 3. Anchors.

3) Feature Engineering Techniques: High-Level Features: These features are representations of abstract and semantic information about the input data. They typically capture complex patterns, object shapes, textures, and context within the image. High-level features are crucial for tasks such as object classification and scene understanding. As an example round form of the apple has a slight asymmetry that necessitates accurate feature detection because it can appear in varied sizes depending on the variety or maturity level. This also serves crucial when having to discern against a similar looking fruit such as in Dataset 2 we trained our model on, and most of this occurs under poor lighting or dense conditions.

Low-Level Features: These features represent more fine-grained details and local patterns in the input data. They typically capture simple structures such as edges, corners, textures, and colors. Low-level features are important for tasks that require precise localization or detection of specific visual elements. This would particularly help with oranges, bananas and pears wherefruit skin gradients and other minor colour changes can all be used to distinguish between different varieties and stages of maturity during the harvest cycle.

Fine-grained information refers to subtle, detailed, and specific visual characteristics present in the input data. It includes features such as small textures, intricate patterns, or subtle color variations.

Concat: Concatenating features from different layers of the backbone network serves several purposes- Hierarchical Feature Fusion: Features extracted from different layers of the backbone network capture information at various levels of abstraction. By concatenating these features, the model can leverage both low-level details and high-level semantic information simultaneously. This hierarchical feature fusion helps improve the model's ability to detect objects of different sizes and complexities.

Multi-Scale Feature Representation: Object detection often requires analyzing images at multiple scales to detect objects of different sizes. Features from different layers with varying receptive fields can effectively detect objects of varying sizes.This multi-scale feature representation enhances the model's robustness to scale variations in objects. The perfect example of this is the Minneapple Dataset we used (Dataset 3). Although Dataset 1 and 2 involved large scale fruits, Minneapple involved a densely packed apple orchard with several tiny fruits of very small scale, and yes, our model adapted well there too.

Enhanced Contextual Information: Concatenating features from different layers enriches the contextual information available to the model. Features from shallow layers provide fine-grained spatial details, while features from deeper layers offer more abstract semantic information. Combining these features allows the model to better understand the context of objects in the image and make more accurate predictions. Let's say in dataset 2 where both the shape of an orange as well as the understanding of its colour comes together to make accurate classification and not confuse it with a lemon.

SPPF: The Spatial Pyramid Pooling Fast layer, handles objects of various sizes within the input image. Handling Variable Input Sizes: One challenge in object detection is efficiently handling objects of different sizes within the input image. Traditional convolutional neural networks (CNNs) require fixed-size input images, which can be limiting when dealing with objects at different scales. The SPPF layer addresses this challenge by allowing the network to operate on feature maps of arbitrary sizes, enabling it to detect objects at multiple scales within the same image.

Spatial Pyramid Pooling: The SPPF layer implements a spatial pyramid pooling operation, which divides the input feature map into multiple regions of varying sizes and then pools features from each region separately. By using pooling operations with different window sizes, the SPPF layer captures features at multiple scales, allowing the network to be more robust to variations in object size.

Factorization: The term "factorization" in SPPF refers to the decomposition of the pooling operation into smaller, more manageable components. Instead of applying pooling operations directly to the entire feature map, the SPPF layer applies them to smaller regions or subregions of the feature

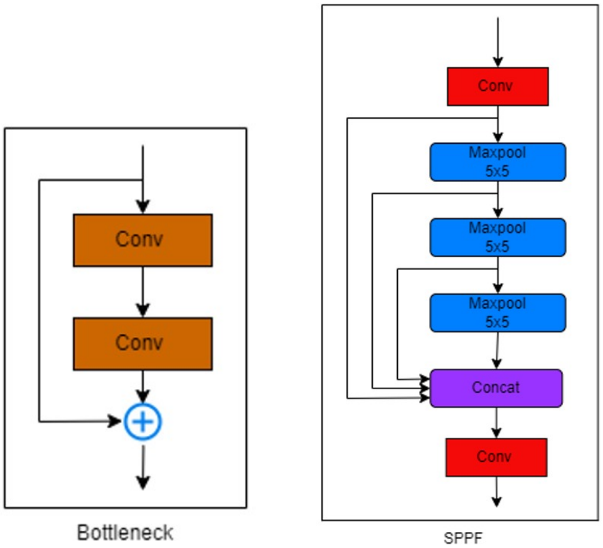


Fig. 4. Key system blocks.

map, reducing the computational complexity of the operation. This factorization process helps maintain the efficiency of the network while still capturing multi-scale features effectively.

Improved Spatial Context: By incorporating features from multiple spatial scales, the SPPF layer enhances the spatial context available to the network. This improved spatial context enables the network to better understand the spatial relationships between objects and their surroundings, leading to more accurate object detection results. A fruit picking robot working in a packed warehouse would be able distinguish between the fruit and non fruit background objects more efficiently, speeding up the process.

Detect: passing inputs of different scales with varying numbers of channels In object detection models like YOLOv3, the detection process often involves analyzing features at multiple scales to detect objects of different sizes. Features from deeper layers with larger receptive fields are better suited for detecting larger objects, while features from shallower layers with smaller receptive fields are more suitable for detecting smaller objects. The choice of having more channels in features from smaller scales and fewer channels in features from larger scales is often driven by the need to capture finer details in smaller objects. Smaller objects may require more spatial information and feature channels to be accurately detected, whereas larger objects may be adequately represented with fewer channels. By passing inputs from multiple scales with varying numbers of channels to the Detect layer, the model can effectively detect objects across a wide range of sizes. The model combines features from different scales and channels to generate bounding box predictions and class probabilities for objects present in the image.

C. Datasets

1) Dataset 1: The dataset comprises a curated collection of 150 high-resolution images, capturing three different fruits: apple, orange and banana. Each image is annotated with bounding boxes outlining the regions of interest containing the fruits. No preprocessing and Augmentation were performed on this dataset. Every model is trained for 100 epochs with a batch size of 13 and input image size of 640 x 640.

Fig. 5. Dataset 1.

2) Dataset 2: It contains images of mixed fruits categorized into four distinct classes, namely apples, oranges, lemon, and pear. Preprocessing steps specifically, RGB images were converted to grayscale using established color conversion algorithms. This transformation not only reduces computational complexity but also emphasizes shape and texture features essential for fruit detection, thereby enhancing model discriminative power. Augmentation techniques like random horizontal and vertical flips were introduced to simulate variations in fruit orientation, ensuring the model's ability to detect fruits irrespective of their spatial orientation. Additionally, rotation augmentation was employed, allowing images to be rotated by ±15 degrees around their center. This augmentation strategy introduces angular diversity, enabling the model to better generalize to fruits positioned at varying angles within the image frame. Here each model is trained for 20 epochs with a batch size of 13 and an input image size of 640 x 640.

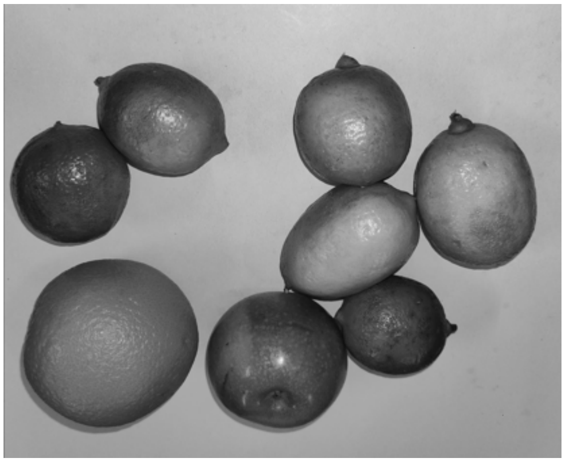


Fig. 6. Dataset 2.

3) Dataset 3: The dataset utilized is MinneApple, designed specifically for apple detection and segmentation within orchard environments, aiming to push the boundaries of fruit detection [6] technology. It focuses solely on a single class object detection: Apples. Acquired from Roboflow, an open-source computer vision tool, MinneApple has a split of 670 training images and 331 testing images totaling over 41,000

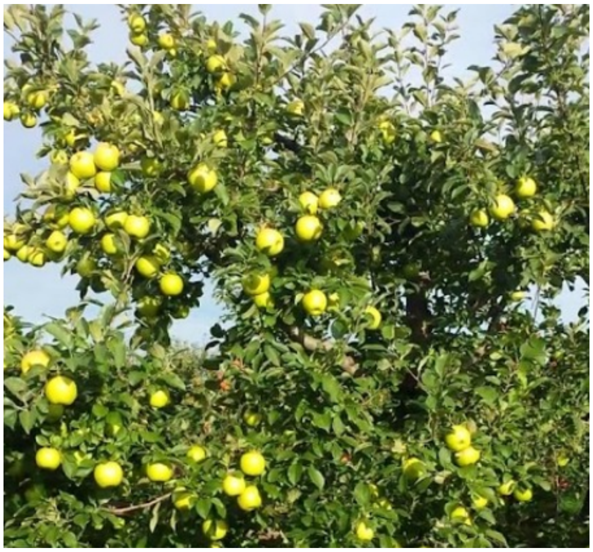


Fig. 7. Dataset 3.

annotated object instances across 1001 images. The data collection process spanned more than a year at the University of Minnesota's Horticultural Research Center, employing a standard Samsung Galaxy S4 cell phone to ensure real-world representativeness. Footage was captured at a controlled speed of 1 m/s to minimize motion blur, with images extracted at regular intervals to encompass diverse lighting, angles, and fruit ripeness stages. This diverse dataset, encompassing various fruit varieties, ripeness stages, and illumination conditions, is pivotal for training robust machine learning models capable of generalizing effectively. No data pre-processing or augmentation was applied to this dataset. All models in Dataset 3 were trained for 30 epochs with a batch size of 13 and input image size of 640 x 640.

IV. Results

In this study, we evaluated the performance of three object detection models **(YOLOv3, YOLOv51, and Our Model)** on three diverse datasets: **(Dataset 1, Dataset 2, and Dataset 3),** as displayed in Table I. We employed standard metrics including precision, recall, mAP@50, and mAP@50-95 to assess their detection accuracy.

TABLE I. Performance Metric Comparison

Dataset	Model	Precision	Recall	mAP	
				@50	@50-95
Dataset 1	YOLOv3	**0.715**	0.617	0.695	0.367
	YOLOV51	0.371	0.546	0.529	0.249
	Our Model	0.692	**0.664**	**0.747**	**0.392**
Dataset 2	YOLOv3	0.971	0.967	0.982	0.725
	YOLOV51	0.949	0.939	0.971	0.782
	Our Model	**0.978**	**0.968**	0.981	0.771
Dataset 3	YOLOv3	**0.700**	0.566	0.638	0.293
	YOLOV51	0.642	0.477	0.528	0.233
	Our Model	0.679	**0.594**	**0.643**	**0.301**

In terms of model size, we found that our model has less parameters than the YOLOv5l variant and the YOLOv3 version, as can be seen in Table II.

TABLE II. Model Details

Model	Layers	Parameters
YOLOV3	262	61,497,430
YOLOV51	368	46,119,048
Our Model	**225**	**45,403,880**

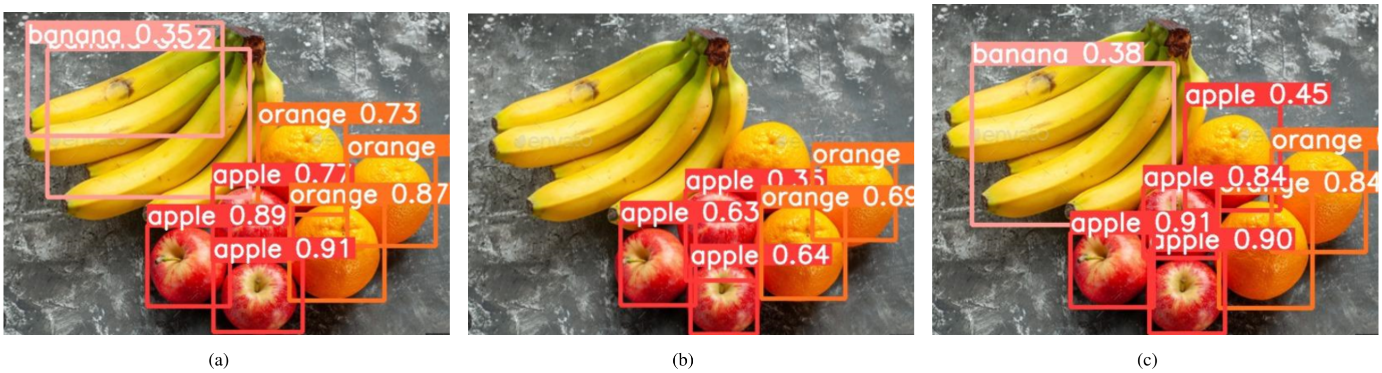
1) Dataset 1: Our model exhibits superior recall and mAP scores (both mAP50 and mAP50-95) compared to YOLOv3. While YOLOv3 has a slight edge in precision, our model's overall performance is better. When compared to YOLOv5, our model surpasses it in all measured metrics: precision, recall, mAP50, and mAP50-95. Thus, our model demonstrates a notable improvement over YOLOv3 in recall and mAP scores, and an overall superior performance across all metrics when compared to YOLOv5.

2) Dataset 2: Our model outperforms YOLOv3 in precision, mAP50, and mAP50-95, indicating superior accuracy and overall performance. However, YOLOv3 has better recall, likely due to its higher sensitivity in detecting a broader range of objects. Compared to YOLOv5, our model excels in all evaluated metrics: precision, recall, mAP50, and mAP50-95. Thus, our model demonstrates a more balanced and accurate performance overall, particularly in precision and mAP scores, while consistently surpassing YOLOv5 across all metrics.

3) Dataset 3: Our model surpasses both models, YOLOv3 and YOLOv5, in all metrics except precision, where YOLOv3 has a slight advantage. Specifically, our model demonstrates superior recall, mAP50, and mAP50-95 compared to YOLOv3.Our model excels in all evaluated metrics, including precision, recall, mAP50, and mAP50-95 compared with the YOLOv5 model.

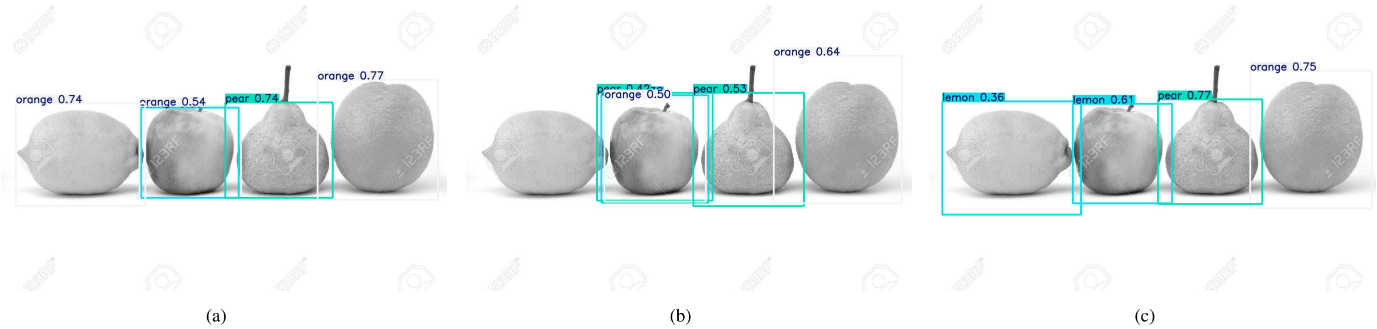
V. Discussion

The results across the datasets, when compared with standard detectors, provide valuable insights into the practical efficacy of our proposed system. For Dataset 1, the superior recall demonstrates the system's ability to detect a high number of fruit instances—apple, banana, and orange. This is particularly significant in dense orchard environments, where fruits are often obscured by leaves, twigs, and other elements within the tree canopy. The higher mAP@50 and mAP@50-95 values further highlight the model's superior performance compared to YOLOv5 and the base YOLOv3 variant in both accurately localizing and classifying fruits, particularly when predicted bounding boxes overlap with ground truth. In a multi-class dataset, this strong performance indicates the model's robustness in detecting multiple types of fruits. While standard YOLOv3 demonstrates a slight advantage in precision due to its effectiveness in reducing false positives, our model achieves a more balanced trade-off between precision and recall, which enhances the detection and classification of multiple fruits (apple, orange, banana) within a single image. The purpose Dataset 1 served was to evaluate the model's ability to detect large objects with occlusion. By using large fruit images, we can also evaluate the model's performance on objects that occupy a significant portion of the image. To perfect the system, a learning rate of 0.01 was applied for all the 100 epochs from scratch with no pre-trained weights. A split of 130 and 20 was chosen for train and validation respectively.



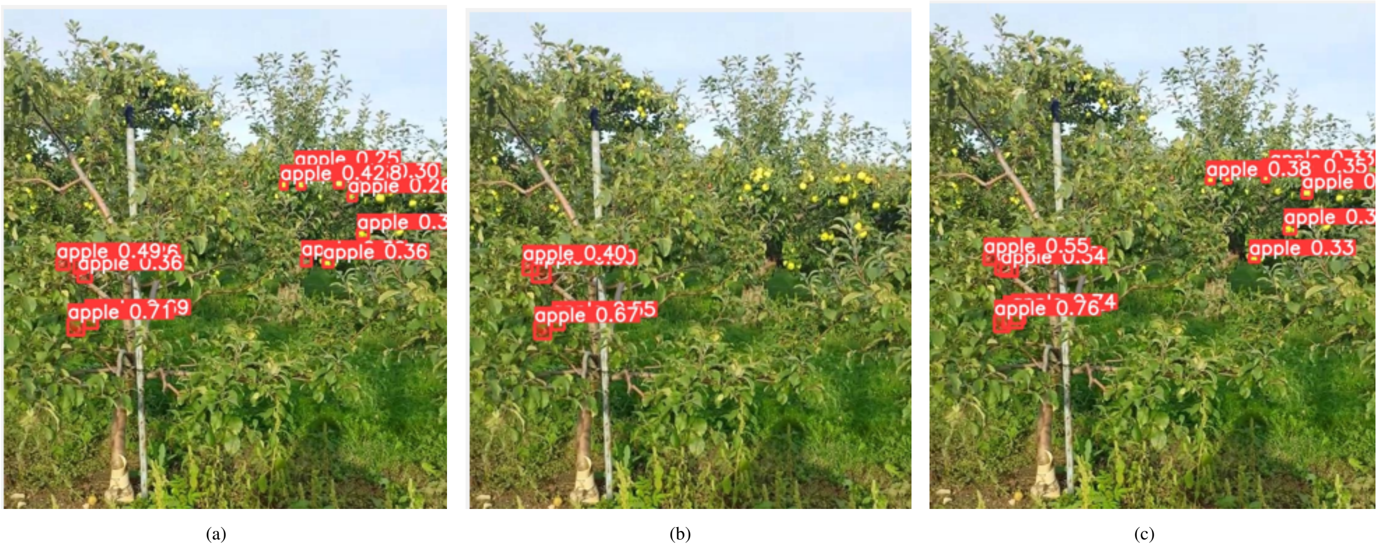
(a) (b) (c)

Fig. 8. Prediction on Dataset 1 using models YOLOv3 model(a) ,YOLOv5 model(b) and our model(c).



(a) (b) (c)

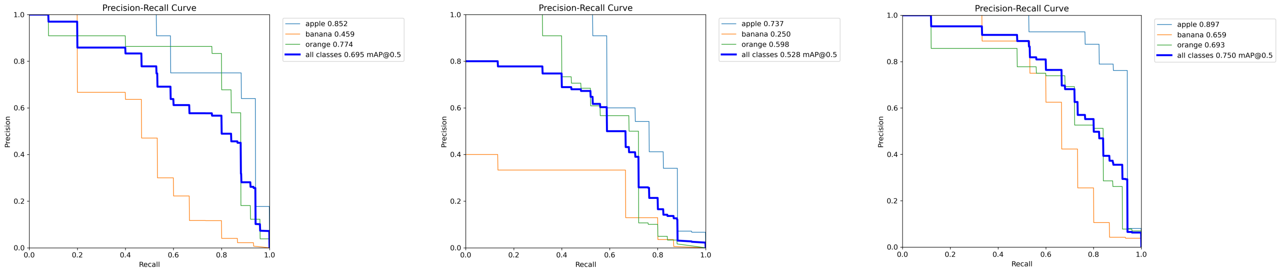
Fig. 9. Prediction on Dataset 2 using models YOLOv3(a), YOLOv5 model(b) and our model(c).



(a) (b) (c)

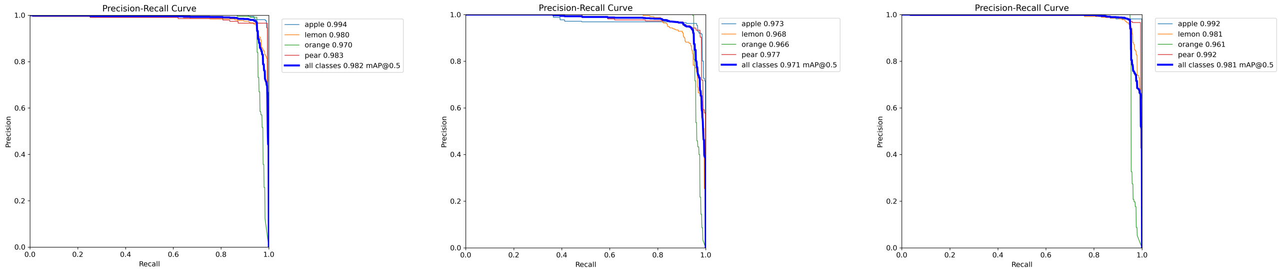
Fig. 10. Prediction on Dataset 3 by using models YOLOv3 model(a), YOLOv5 model(b) and our model(c).

In Dataset 2, our model exhibits superior performance in correctly identifying fruits without mislabeling background objects. This precision is particularly critical in controlled environments for distinguishing between similar objects or fruits of varying sizes. Although the mAP at a threshold of 50 reflects strong localization capabilities, YOLOv5l demonstrates a marginally better performance, likely due to the high similarity in shape between certain fruits, such as lemons and apples, or lemons and oranges. Despite standard YOLOv3 achieving higher recall, it compromises accuracy, as evident from the labels on the bounding boxes. This dataset had fruits of medium to large size. Further, the inclusion of grayscale



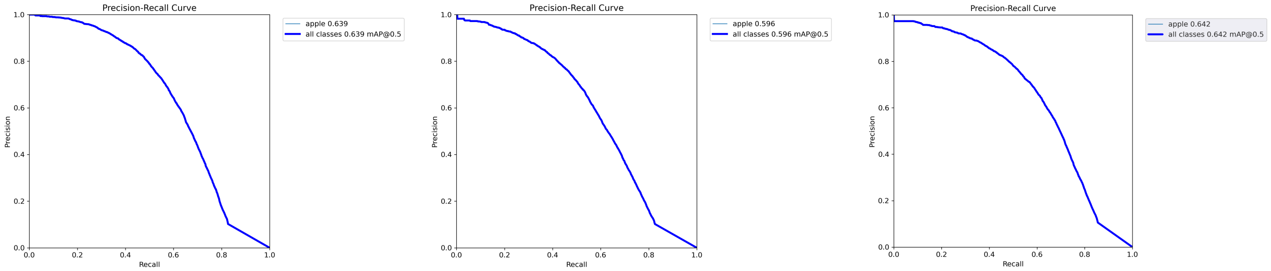
(a) (b) (c)

Fig. 11. Precision recall graph of Dataset 1 applied on YOLOv3 model(a), YOLOv5(b) model and our model(c).



(a) (b) (c)

Fig. 12. Precision recall graph of Dataset 2 applied on YOLOv3(a) model, YOLOv5 model(b) and our model(c).



(a) (b) (c)

Fig. 13. Precision recall graph of Dataset 3 applied onc YOLOv3 model(a), YOLOv5 model(b) and our model(c).

images and augmentation techniques (horizontal/vertical flips and rotation) helps increase the dataset's diversity and improve the model's generalization capabilities. A learning rate= of 0.01 was applied for all the 20 epochs with no pre-trained weights, and a split of 1450 and 230 for train and validation was chosen respectively.

Dataset 3 presents a slightly different trend. In dense orchard settings, precision is key in reducing false positives, such as branches, leaves, or occluded apples being incorrectly classified as apples. While standard YOLOv3 shows a slight advantage in minimizing these misclassifications, our model outperforms in terms of recall and mAP@50 and 90. This indicates that it detects a higher number of apples, even under challenging conditions where fruits may be partially hidden or closely packed. In such densely populated environments, the superior recall of our model is crucial, ensuring comprehensive apple detection, even when some fruits are obscured by branches. Furthermore, the higher mAP scores underscore the model's ability to accurately and consistently localize apples, particularly in cases where apples are tightly clustered, which is essential for effective yield estimation in dense orchards. We chose dataset 3 to specifically evaluate the model's performance on detecting very small objects. The MiniApple dataset provided a challenging benchmark for object detection tasks involving tiny objects. We applied a learning rate=0.01 over all 30 epochs, no pre-trained weights, and the 1001 images were kept at a split of 670 and 331 for train and validation respectively. It is noteworthy that high mAP@50 and mAP@50-95 scores proved powerful localization capabilities specially over the Minneapple dataset where similar research such as [23] which put to use MHT with YOLO and Faster RCNN, suffered at counting the fruits and had to rely on high velocity algorithms such as DeepSORT to improve performance while our system could correctly identify a very high number of tiny apples off the dense branches.

VI. Conclusion

Concerning model size, we observed that our model contained fewer parameters than both the YOLOv3 version and the YOLOv5l variant. Despite containing lower number of layers than YOLOv3, our model performed beter feature extraction. This shows that we can achieve good accuracy without a large and complex structure, making our model a great choice for tasks that need both speed and efficiency in object detection. This lightweight model can be effectively used in applications where efficiency is crucial, offering fast processing and reduced memory usage compared to larger models like YOLOv3 and YOLOv5. This certainly means we have got through with our first and third primary objectives listed at the start of this paper. In addition, the SPPF module working within our model provides a reliable solution to the issues of multi-scale fruit detection. Using pooling layers of varied sizes improves the model's capacity to detect fruits of vastly varying sizes, whether they are small in the Minneapple dataset's orchards or the large ones in the Mixed Fruit dataset. This method decreases sensitivity to variations in input resolution, resulting in consistent performance across different image qualities. Furthermore, by merging characteristics from several scales within a single feature map, SPPF enhances the fruit object representation with fine-grained details as well as global context, ultimately enhancing fruit recognition accuracy and reliability in a variety of situations. With this, we have achieved our second primary objective. In summary, our model showcases strong performance in multi-class fruit detection by taking a balanced approach and striking a worthy balance between precision and recall, as opposed to models like YOLOv3 and YOLOv5. The fact that we successfully tested over images with fruits of different classes hints at success with our fourth primary objective. Also, our model has superior overall detection accuracy, as indicated by the greatest mAP@50 score, suggesting its ability to recognize true positives despite minor differences in bounding box placement. Furthermore, our model performs consistently across IoU levels, achieving competitive mAP@50-95 values and assuring accurate fruit location pinpointing even under stringent bounding box overlap criteria. However, while our model demonstrates promising results, there is potential for further improvement. The model still does not perform well in cases of occlusion. Improvement in fusion techniques to better mix information from multiple layers is needed and for this, we can investigate sophisticated backbones such as EfficientNet. By exposing the model to a greater range of training data, data augmentation can also aid in enhancing the model's capacity for generalisation. Finally, testing the model against a variety of benchmarks can reveal the model's advantages and disadvantages as well as point out areas that still want improvement. We may improve our model's performance and attain even greater results in object detection tasks by implementing these strategies.

References

[1] Ritesh Kumar Singh, Rafael Berkvens, And Maarten Weyn, "AgriFusion: An Architecture For IoT And Emerging Technologies Based On A Precision Agriculture Survey", September 2021, IEEE Access, PP(99):1-1, DOI: 10.1109/ACCESS.2021.3116814

[2] "Study to determine Post Harvest Losses of Agri Produce in India",NABCONS. 2022

[3] Maheswari, Prabhakar and Raja, Purushothaman and Apolo-Apolo, Orly Enrique and Pérez-Ruiz, Manuel, "Intelligent Fruit Yield Estimation for Orchards Using Deep Learning Based Semantic Segmentation Techniques—A Review", Frontiers in Plant Science, Vol 12, 684328, 2021

[4] Redmon, J. and Farhadi, A. (2018) YOLOv3 An Incremental Improvement. Computer Science, arXiv 1804.02767.

[5] L. Fu et al., "Fast and Accurate Detection of Banana Fruits in Complex Background Orchards", IEEE Access, vol. 8, pp. 196835-196846, 2020, doi: 10.1109/ACCESS.2020.3029215

[6] O. M. Lawal, "YOLOMuskmelon: Quest for Fruit Detection Speed and Accuracy Using Deep Learning", IEEE Access, vol. 9, pp. 15221-15227, (2021), doi: 10.1109/ACCESS.2021.3053167

[7] Xiao, B., Nguyen, M. Yan, W.Q., "Fruit ripeness identification using YOLOv8 model", Multimed Tools Appl, 83, 28039–28056,2024. doi.org/10.1007/s11042-023-16570-9

[8] Lawal, O.M, "Study on strawberry fruit detection using light weight algorithm", Multimed Tools Appl., 83, 8281–8293,2024. doi.org/10.1007/s11042-023-16034-0

[9] Xia, Y., Nguyen, M., Yan, W.Q., "A Real-Time Kiwifruit Detection Based on Improved YOLOv7", Yan, W.Q., Nguyen, M., Stommel, M. (eds), "Image and Vision Computing", IVCNZ 2022. Lecture Notes in Computer Science, vol 13836. Springer, Cham.,2023 https://doi.org/10.1007/978-3-031-25825-1_4

[10] Ranjan Sapkota, Dawood Ahmed, Manoj Karkee, "Comparing YOLOv8 and Mask RCNN for object segmentation in complex orchard environments", C.V.P.R., 2024. doi.org/10.48550/arXiv.2312.07935

[11] Lawal OM, Zhu S, Cheng K, "An improved YOLOv5s model using feature concatenation with attention mechanism for real-time fruit detection and counting", Front. Plant Sci., 14:1153505, 2023. doi: 10.3389/fpls.2023.1153505

[12] Liu Y, Ren H, Zhang Z, Men F, Zhang P, WuDandFeng R, "Research on multi-cluster green persimmon detection method based on improved Faster RCNN", Front. Plant Sci.14:1177114, 2023, PMID: 37346117; PMCID: PMC10279974

[13] P. Ganesh , K. Volle , T. F. Burks , S. S. Mehta (2019). "Deep Orange: Mask R-CNN based Orange Detection and Segmentation", IFAC-PapersOnLine,52-30,2019 70–75.Elsevier

[14] Zhenzhen Song, Longsheng Fu, Jingzhu Wu, Zhihao Liu, Rui Li, Yongjie Cui,"Kiwifruit detection in field images using Faster R-CNN with VGG16",IFAC-PapersOnLine,52-30, 2019,76-81.

[15] M. H. Junos, A. S. Mohd Khairuddin, S. Thannirmalai, "Automatic detection of oil palm fruits from UAV images using an improved YOLO model",*Visual Computer*, vol. 38, no. 11, pp. 2341–2355, 2022. [Online]. Available: doi.org/10.1007/s00371-021-02116-3

[16] Ahmad Aljaafreh et.al., A Real-Time Olive Fruit Detection for Harvesting Robots Based on YOLO Algorithms, Acta Technologica Agriculturae, 2023,26,3,121-132. https://doi.org/10.2478/ata-2023-0017 ·

[17] Wang L, Zheng H, Yin C, Wang Y, Bai Z, Fu W. Dense Papaya Target Detection in Natural Environment Based on Improved YOLOv5s. Agronomy. 2023, 13(8):2019. https://doi.org/10.3390/agronomy13082019.

[18] Diwan, T., Anirudh, G.,Tembhurne J.V., Object detection using YOLO: challenges, architectural successors, datasets and applications, Multimed Tools Appl, 82, 9243–9275.2023. https://doi.org/10.1007/s11042-022-13644-y

[19] O M Lawal et al,Ablation studies on YOLOFruit detection algorithm for fruit harvesting robot using deep learning, (2021) IOP Conf. Ser.: Earth Environ. Sci., 922 012001.

[20] R. Yang, Y. Hu, Y. Yao, M. Gao, and R. Liu, "Fruit target detection based on BCo-YOLOv5 model," *Mobile Information Systems*, vol. 2022, Article ID 8457173, 2022. [Online]. Available: https://doi.org/10.1155/2022/8457173

[21] Huang ML, Wu YS, GCS-YOLOV4-Tiny: A lightweight group convolution network for multi-stage fruit detection, Math Biosci Eng. 2023 Jan;20(1):241-268. doi: 10.3934/mbe.2023011

[22] Hulin Kuang and Cairong Liu and Leanne Lai Hang Chan and Hong Yan,"Multi-class fruit detection based on image region selection and improved object proposals",Neurocomputing, Vol 283, 241-255,2018, doi https://doi.org/10.1016/j.neucom.2017.12.057,

[23] Juan Villacrés, Michelle Viscaino, José Delpiano, Stavros Vougioukas, Fernando Auat Cheein , "Apple orchard production estimation using deep learning strategies: A comparison of tracking-by-detection algorithms", Computers and Electronics in Agriculture 204, [Online] Available: https://doi.org/10.1016/j.compag.2022.107513