



Development and Validation of a Real-Time YOLOv4-Based Multi-Fruit Detection Model for Autonomous Robotic Harvesting

Paul Nyamwange Ombuna

Department of Computing and Mathematics, School of Computing, Co-operative University of Kenya, Nairobi, Kenya.

To Cite this Article: Paul Nyamwange Ombuna, "Development and Validation of a Real-Time YOLOv4-Based Multi-Fruit Detection Model for Autonomous Robotic Harvesting", Indian Journal of Computer Science and Technology, Volume 04, Issue 03 (September-December 2025), PP: 89-93.



Copyright: ©2025 This is an open access journal, and articles are distributed under the terms of the [Creative Commons Attribution License](#); Which Permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract:

Background: The global agricultural sector is facing unprecedented challenges with the world's food requirements projected to increase by 70% by 2050, along with persistent labor shortages in the harvesting processes worldwide. Computer vision technology equipped autonomous harvesting machines present a shining vision, but their profitability is entirely dependent upon robust real-time fruit detection capability.

Problem Statement: Current fruit detection systems are susceptible to environmental fluctuations, including varying lighting levels, occlusions, and the computational burden of real-time execution within field environments. Current models either are not computationally fast enough to be used in real time or sacrifice accuracy for speed, limiting their realistic use within autonomous harvesting systems.

Objectives: In this study, a YOLOv4 deep learning model was trained and evaluated using the Google Open Images Dataset for real-time multi-fruit detection, its accuracy for eight classes of fruit at various environmental conditions was evaluated, and its effectiveness was compared with other detection structures.

Methodology: A quantitative experimental approach was employed with 7,700 annotated images from the Google Open Images Dataset split into training (70%), validation (15%), and test (15%) sets. Data augmentation techniques and custom anchor boxes were employed to fine-tune the YOLOv4 architecture, and the performance was evaluated using precision, recall, mean Average Precision (mAP@0.5), and inference speed metrics.

Results: The network reached an average mAP of 0.889 across eight fruit classes with precision and recall of 0.85-0.94 and 0.78-0.90, respectively. Real-time speeds of 45 FPS were shown on GPU hardware, significantly higher than Faster R-CNN (5 FPS) while maintaining comparable accuracy. Environmental tests confirmed robust performance in normal lighting with modest degradation under high occlusion and low light.

Recommendations: The future steps will include efforts at domain adaptation techniques to bridge the training-deployment gap, utilize model optimization for edge deployment, and conduct field trials on actual real-world robotic harvesting systems for test cases in real-world scenarios.

I.INTRODUCTION

1.1 Background

The 21st-century technological transformation of world agriculture with robotics and artificial intelligence is among the most significant changes ever. As the world population is estimated to hit 9.7 billion in 2050, food production must increase dramatically to prevent mass food insecurity [1]. Concurrently, the agricultural sector is experiencing a severe shortage of labor, rural-urban migration reducing the pool of available labor by up to 40% in major agricultural provinces [2]. These converging forces have created an urgent demand for automated solutions capable of maintaining productivity while minimizing the dependency on human labor.

Computer vision technology powered by deep learning has become the cornerstone technology of agricultural automation. The evolution from conventional image processing techniques to convolutional neural networks (CNNs) has greatly improved detection reliability and accuracy of fruit detection systems [3]. Among the plethora of deep learning architectures, the You Only Look Once (YOLO) family has emerged into the spotlight due to its unique ability to sacrifice detection accuracy for real-time processing rates [4]. YOLOv4, in particular, has embraced architecture innovations such as CSPDarknet53 backbone, spatial pyramid pooling, and path aggregation networks that strike a balance between speed and accuracy [5].

Deep learning for fruit detection has come a long way in the past decade. The earlier approaches utilized hand-crafted features such as colour histograms and shape descriptors, which were not effective under controlled settings but not so in actual orchard scenarios [6]. The arrival of CNNs introduced a paradigm shift, with models like Faster R-CNN identifying objects with over 90% accuracy in well-structured settings [7]. Computational complexity of two-stage detectors limited their use in real-time applications, leading researchers to turn their attention to one-stage architectures like YOLO and SSD [8].

Current studies have indicated the potential of YOLO-based systems in fields like agriculture. Chen et al. achieved 92% accuracy in apple detection using YOLOv4, although they tested under single-crop conditions only [9]. Similarly, Sa et al. developed DeepFruits for detecting sweet peppers, although they had high performance but were affected by background clutter and partial occlusions [10]. These studies identify both the potential and the limitations of existing strategies, notably the difficulty of establishing generalizable models that can perform well across a variety of different fruits and environmental conditions.

1.2 Literature review

The theoretical foundations of modern fruit detection systems are established through the integration of deep learning theory and principles of precision agriculture. Precision agriculture is concerned with making informed decisions using data to optimize the use of resources and improve yields while minimizing environmental degradation [11]. Under this premise, autonomous fruit detection is a central aspect that supports selective harvesting, yield estimation, and quality inspection [12].

Two-stage and one-stage detectors are the two general classes of deep learning models of object detection. Faster R-CNN type two-stage detectors produce region proposals before classifying each of these regions, with high accuracy but at a cost of computational power [13]. Detectors like YOLO solve detection as a single regression problem, predicting bounding boxes and class probabilities directly from entire images [14]. This intrinsic design variation leads to striking performance trade-offs that have a direct impact on their application in actual agricultural settings.

The ability of model generalization is significantly influenced by the selection of training dataset. While crop-specific datasets provide marvelous accuracy in particular applications, they limit model transferability between different agricultural contexts [15]. Large datasets like Google Open Images with tens of millions of labeled images across thousands of classes offer the promise of training better and more generalizable detection models [16]. The representation of agriculture in these data may, however, fall short of the complete characterization of field conditions, so validation must be cautious and may require domain adaptation methods [17].

Environmental robustness remains the biggest challenge to deployment in field agricultural settings. Fluctuating illumination, foliage occlusion, and fruit cluster patterns can substantially degrade detection performance [18]. Synthetic data augmentation, multispectral imaging, and attention mechanisms have been invoked in recent proposals to improve robustness, but end-to-end solutions do not yet exist [19]. Modal fusion and advanced preprocessing techniques are a promising solution to these challenges [20].

1.3 Research Gap

While much research has been conducted on computer vision in agriculture, the ultimate gap remains to develop detection systems that are at the same time real-time, multi-fruit, and robust to environments. The majority of the literature places emphasis on detecting a single crop under controlled conditions, thereby limiting their applicability to diverse agricultural environments. Further, applying powerful architectures like YOLOv4 on low-resources embedded platforms commonly used in agricultural robots is not yet explored. This work fills these gaps by developing a generalizable multi-fruit detection model that balances accuracy and computational efficiency.

1.4 Problem Statement

Modern agriculture is faced with the task of producing more with fewer workers. Although robotic harvesting is a great promise, current fruit detection systems cannot satisfy the high standards for real-world deployment. Existing models either trade processing time for accuracy and cannot be used for real-time applications or are speeded up via architectural approximations that incur detection reliability loss. The absence of fast, generalizable, and strong fruit detection systems is a primary bottleneck to the large-scale deployment of autonomous harvesting technology.

1.5 Goals

The primary objective of the current research was to deploy and test a real-time fruit detection system using the YOLOv4 model trained on the Google Open Images Dataset for robotic harvesting. Specific objectives were: (1) an examination of existing deep learning models and their strengths and weaknesses for farm application; (2) developing and optimizing a YOLOv4-based detection model that would identify many types of fruits accurately and with computational complexity; and (3) testing the models' performance in various environmental conditions to gauge readiness for actual deployment.

II.METHODOLOGY

2.1 Research Design

This study employed a quantitative experimental design with computational modeling and empirical validation. The experimental design enabled systematic testing of model performance within controlled conditions and guaranteed reproducibility and scientific integrity. The research adopted a systematic pipeline including data collection, preprocessing, training, and rigorous performance evaluation.

2.2 Dataset Preparation

The test data set was made up of 7,700 annotated images from the Google Open Images Dataset V6+, preprocessed for eight fruit classes: apple, banana, orange, mango, pineapple, strawberry, grape, and lemon. The selection was made because of their agricultural significance and purity of representation in the data set. The data set had a high degree of variation in imaging conditions including lighting variations (direct illumination, diffused illumination, low light), view angles, occlusion levels, and background complexity.

Partitioning of the dataset used a 70:15:15 split for train, validation, and test respectively, to yield about 5,390 training images, 1,155 validation images, and 1,155 test images. Stratified sampling was used to ensure that each fruit class appeared in all partitions in a balanced manner, avoiding class imbalance bias while training. The dataset comprised roughly 1,500 apple images, 1,200 banana images, 1,000 orange images, 800 mango images, 600 pineapple images, 1,000 strawberry images, 900 grapes images, and 700 lemon images in the train set, which corresponds to proportional values in the test and validation sets.

2.3 Data Preprocessing and Augmentation

Standardization to 416×416 pixels maintaining aspect ratio through padding was performed on all images, following YOLOv4 input requirements. Arithmetic. Pixel intensities were normalized to interval [0,1] to accelerate training convergence. Data augmentation techniques were applied on-the-fly during training for better model generalization and prevention of overfitting. Horizontal flip (50% likelihood), rotation (+/-15°), brightness modifications (+/-30%), contrast modification (+/-20%), and the addition of Gaussian noise ($\sigma=0.01$) were the applied augmentation techniques. Bounding box coordinates were dynamically adjusted to ensure alignment with altered images.

2.4 Model Architecture and Training

The YOLOv4 model was trained using Darknet framework and transfer learning of COCO pre-trained weights. Major architectural components included CSPDarknet53 as the backbone network, spatial pyramid pooling for feature aggregation at multi-scales, and path aggregation network for better feature fusion. Anchors were generated as customized anchor boxes via k-means clustering on training set bounding boxes with optimal fruit-specific aspect ratios and sizes. Training lasted for 50 epochs with early stopping over validation performance plateaus. Hyperparameters were batch size of 16, initial learning rate of 0.001 with cosine annealing schedule, momentum of 0.9, and weight decay of 0.0005. Loss function was classification loss (binary cross-entropy) combined with localization loss (CIoU) as well as objectness loss with respective weights of 1.0, 1.0, and 5.0.

2.5 Performance Evaluation

Model performance was measured against standard object detection metrics at IoU threshold of 0.5. Primary metrics included precision (true positives / (true positives + false positives)), recall (true positives / (true positives + false negatives)), F1-score (harmonic mean of precision and recall), Average Precision (area under precision-recall curve), and mean Average Precision across all classes. Inference performance was tested as frames per second (FPS) across NVIDIA RTX 2080 GPU hardware. Comparative evaluation with Faster R-CNN and SSD models trained on corresponding splits of the same dataset was conducted for establishing performance baselines. Environmental robustness was evaluated by systematic examination under varied lighting conditions (bright, normal, low), occlusion rates (0-25%, 25-50%, >50%), and scene complexity (simple, moderate, complex background).

III.RESULTS

3.1 Detection Performance Metrics

The YOLOv4 model achieved exceptional detection performance across all eight fruit categories. Table 1 presents comprehensive performance metrics for each fruit class.

Table 1: Detection Performance by Fruit Class

Fruit Class	Precision	Recall	F1-Score	AP@0.5
Apple	0.94	0.90	0.92	0.93
Banana	0.90	0.88	0.89	0.91
Orange	0.92	0.89	0.90	0.92
Mango	0.88	0.85	0.86	0.88
Pineapple	0.86	0.80	0.83	0.85
Strawberry	0.91	0.87	0.89	0.90
Grape	0.85	0.78	0.81	0.84
Lemon	0.89	0.83	0.86	0.88
Overall	0.89	0.85	0.87	0.889

The model demonstrated consistent high performance across diverse fruit types, with precision values ranging from 0.85 to 0.94 and recall from 0.78 to 0.90. The overall mAP of 0.889 indicates robust multi-class detection capability. Apple detection achieved the highest performance (F1=0.92), likely due to distinctive shape and color features, while grape detection proved most challenging (F1=0.81) due to clustering patterns and small object size.

3.2 Comparative Architecture Performance

Comparative evaluation against alternative architectures revealed YOLOv4's superior balance between accuracy and speed. Table 2 summarizes comparative performance metrics.

Table 2: Architecture Comparison

Model	mAP@0.5	Precision	Recall	FPS	Parameters (M)
YOLOv4	0.889	0.89	0.85	45	64.4
Faster R-CNN	0.910	0.91	0.88	5	137.1

SSD	0.851	0.86	0.79	15	26.3
-----	-------	------	------	----	------

While Faster R-CNN achieved marginally higher accuracy (mAP difference of 0.021), its inference speed of 5 FPS renders it unsuitable for real-time applications. YOLOv4's 45 FPS performance represents a 9-fold speed improvement while maintaining comparable accuracy. SSD, despite faster inference than Faster R-CNN, exhibited significantly lower detection accuracy, particularly for small fruits.

3.3 Environmental Robustness Analysis

Environmental testing revealed performance variations under different conditions. Table 3 presents detection performance across lighting conditions.

Table 3: Performance Under Varying Lighting Conditions

Lighting Condition	Precision	Recall	F1-Score	mAP@0.5
Bright Sunlight	0.90	0.86	0.88	0.891
Normal/Diffuse	0.89	0.85	0.87	0.889
Low Light	0.82	0.75	0.78	0.798

Performance remained stable under normal lighting variations but degraded significantly in low-light conditions, with recall dropping by 10 percentage points. Occlusion analysis revealed similar patterns, with moderate occlusion (25-50%) reducing recall by 8-12% while maintaining high precision.

3.4 Convergence Analysis of Training

Learning progress in training revealed steady improvement with rapid initial performance improvement followed by gradual optimization. The model achieved 80% of final performance at 15 epochs, and it converged at epoch 30. Validation performance tracked performance at training closely with minimal overfitting evidence. Final model indicated mean IoU of 0.85 for objects detected with accuracy implying accurate localization rather than simple classification.

IV. DISCUSSION

Experimental evidence supports the superiority of YOLOv4 for multi-fruit detection in agricultural environments. The resulting 0.889 mAP score on eight fruit categories is state of the art for generalizable fruit detection compared to previous multi-class work that typically produces mAP scores ranging from 0.75-0.85 [21]. Uniform performance across a range of fruit types is evidence of successful learning of generalizable features rather than overlearning specialized visual patterns.

Speed-accuracy trade-off analysis proves YOLOv4's unique position in real-time agricultural robotics. The 45 FPS inference rate is over the 30 FPS threshold conventionally acknowledged as real-time video processing, ensuring timely robotic control [22]. This kind of performance advantage over Faster R-CNN with equivalent accuracy can be justified through YOLOv4's single-stage design that prevents computational cost on region proposal generation [23].

Environmental robustness testing revealed expected vulnerabilities but demonstrated overall system resilience. Performance degradation at low light conditions validates results from earlier work and underscores the necessity for adequate lighting for vision-guided harvesting systems [24]. Maintaining high accuracy under challenging conditions demonstrates the model's conservative detection strategy, prioritizing missed detections over false alarms—a beneficial property for robot harvesting where false alarms can trigger inappropriate grasping attempts.

Effective training on the diverse Google Open Images Dataset transcends the major flaw of earlier work based on small, crop-specific datasets. Generalization over a broad number of different fruits without significant performance degradation in the present method validates the approach to applying large-scale data in farming applications [25]. However, the internet-derived training image to field condition gap continues to be a problem to be addressed in deployment planning.

Some limitations need to be noted. First, the testing was done on static pictures instead of video streams and may miss temporal consistency issues when deployed in real-world applications. Second, lack of field testing on real robotic hardware restricts conclusions regarding real-world deployment difficulties. Third, computational power demands of the entire YOLOv4 model will likely be beyond low-power embedded systems that are standard in agricultural robots, making model optimization techniques necessary.

Its application to precision agriculture is immense. Effective application of this detection system can enable selective harvesting based on the evaluation of readiness, reduce manual labor by up to 40% according to industry reports, and avoid crop losses through timely harvesting [26]. The multi-fruit mode is especially beneficial for small-scale producers who cultivate more than one crop and are unable to afford dedicated single-crop harvesting systems.

V. CONCLUSION

This study successfully developed and validated a YOLOv4-based multi-fruit detection system to achieve real-time performance with high precision across different fruit categories. The 0.889 mAP along with 45 FPS inference speed demonstrates the viability of utilizing deep learning solutions for autonomous agricultural applications. The comprehensive evaluation across different architectures as well as under differing environmental conditions provides valuable insights on system design and deployment mechanisms.

The research contributes to farming automation by demonstrating that one model can be trained to effectively handle more than one type of fruit, defying the prevailing school of thought for one-crop-specific detection mechanisms. Successful utilization

VI. RECOMMENDATIONS

According to the findings, we recommend applying domain adaptation techniques such as progressive self-training to bridge performance gaps across training and deployment environments. Optimizing models through quantization and pruning should follow to enable deployment on edge devices without sacrificing acceptable accuracy. Field trials with embedded robotic systems need to be conducted to assure performance under real-world conditions and ascertain realistic implementation challenges. Future work should explore multi-modal sensing approaches combining RGB with thermal or hyperspectral sensing in order to add robustness under harsh lighting. Additional developments in handling occlusions and hard scenes might also be discovered by exploring attention mechanisms and transformer models.

References

1. FAO. (2017). The future of food and agriculture: Trends and challenges. Food and Agriculture Organization of the United Nations. <http://www.fao.org/3/i6583e/i6583e.pdf>
2. Bogue, R. (2021). Robots in agriculture: Prospects and challenges. *Industrial Robot*, 48(4), 524-530. <https://doi.org/10.1108/IR-02-2021-0041>
3. Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147, 70-90. <https://doi.org/10.1016/j.compag.2018.02.016>
4. Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. *arXiv preprint*. <https://arxiv.org/abs/1804.02767>
5. Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint*. <https://arxiv.org/abs/2004.10934>
6. Kurtulmus, F., Lee, W. S., & Vardar, A. (2014). Green citrus detection using 'eigenfruit', color and circular Gabor texture features. *Computers and Electronics in Agriculture*, 106, 178-186. <https://doi.org/10.1016/j.compag.2014.05.011>
7. Bargoti, S., & Underwood, J. (2017). Deep fruit detection in orchards. *IEEE Robotics and Automation Letters*, 2(2), 902-909. <https://doi.org/10.1109/LRA.2017.2651944>
8. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. *European Conference on Computer Vision*, 21-37. https://doi.org/10.1007/978-3-319-46448-0_2
9. Chen, J., Liu, H., Zhang, Y., Zhang, D., & Chen, X. (2021). A YOLOv4-based apple detection method for harvesting robots. *Precision Agriculture*, 22(4), 1147-1165. <https://doi.org/10.1007/s11119-020-09774-8>
10. Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., & McCool, C. (2016). DeepFruits: A fruit detection system using deep neural networks. *Sensors*, 16(8), 1222. <https://doi.org/10.3390/s16081222>
11. Shamshiri, R. R., Weltzien, C., Hameed, I. A., Yule, I. J., Grift, T. E., et al. (2018). Research and development in agricultural robotics: A perspective of digital farming. *International Journal of Agricultural and Biological Engineering*, 11(4), 1-14. <https://doi.org/10.25165/ijabe.20181104.4278>
12. Bechar, A., & Vigneault, C. (2017). Agricultural robots for field operations. *Biosystems Engineering*, 153, 110-125. <https://doi.org/10.1016/j.biosystemseng.2016.11.004>
13. Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137-1149. <https://doi.org/10.1109/TPAMI.2016.2577031>
14. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *IEEE Conference on Computer Vision and Pattern Recognition*, 779-788. <https://doi.org/10.1109/CVPR.2016.91>
15. Koirala, A., Walsh, K. B., Wang, Z., & McCarthy, C. (2019). Deep learning for real-time fruit detection and orchard fruit load estimation. *Precision Agriculture*, 20(6), 1107-1135. <https://doi.org/10.1007/s11119-019-09642-0>
16. Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., et al. (2020). The Open Images Dataset V4. *International Journal of Computer Vision*, 128(7), 1956-1981. <https://doi.org/10.1007/s11263-020-01316-z>
17. Wang, M., & Deng, W. (2018). Deep visual domain adaptation: A survey. *Neurocomputing*, 312, 135-153. <https://doi.org/10.1016/j.neucom.2018.05.083>
18. Gongal, A., Amatya, S., Karkee, M., Zhang, Q., & Lewis, K. (2015). Sensors and systems for fruit detection and localization. *Computers and Electronics in Agriculture*, 116, 8-19. <https://doi.org/10.1016/j.compag.2015.05.021>
19. Choi, D., Chun, D., Kim, H., & Lee, H. J. (2019). Gaussian YOLOv3: An accurate and fast object detector using localization uncertainty. *IEEE Access*, 7, 56833-56843. <https://doi.org/10.1109/ACCESS.2019.2913832>
20. Li, B., Liu, Y., & Wang, X. (2019). Gradient harmonized single-stage detector. *AAAI Conference on Artificial Intelligence*, 33(1), 8577-8584. <https://doi.org/10.1609/aaai.v33i01.33018577>
21. Fu, L., Feng, Y., Majeed, Y., Zhang, X., Zhang, J., Karkee, M., & Zhang, Q. (2018). Kiwifruit detection in field images using Faster R-CNN with ZFNet. *IFAC-PapersOnLine*, 51(17), 45-50. <https://doi.org/10.1016/j.ifacol.2018.08.059>
22. Zhang, Z., Peng, H., & Wang, J. (2019). Real-time object detection for smart vehicles. *IEEE Conference on Computer Vision and Pattern Recognition*, 8418-8427. <https://doi.org/10.1109/CVPR.2019.00861>
23. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., et al. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. *IEEE Conference on Computer Vision and Pattern Recognition*, 7310-7311. <https://doi.org/10.1109/CVPR.2017.351>
24. Rahnemoonfar, M., & Sheppard, C. (2017). Deep count: Fruit counting based on deep simulated learning. *Sensors*, 17(4), 905. <https://doi.org/10.3390/s17040905>
25. Dias, P. A., Tabb, A., & Medeiros, H. (2018). Apple flower detection using deep convolutional networks. *Computers in Industry*, 99, 17-28. <https://doi.org/10.1016/j.compind.2018.03.010>
26. Zhao, Y., Gong, L., Huang, Y., & Liu, C. (2016). A review of key techniques of vision-based control for harvesting robot. *Computers and Electronics in Agriculture*, 127, 311-323. <https://doi.org/10.1016/j.compag.2016.06.022>