Edge Detection of Strawberries Ripeness Based on Model Optimization Using Intel OpenVINO Toolkit

Yosef Adhitya Duta Dewangga(✉), Agus Bejo, and Eka Firmansyah

Department of Electrical Engineering and Information Technology, Faculty of Engineering, Universitas Gadjah Mada, Jl. Grafika No. 2 Kampus UGM, Yogyakarta, Indonesia
yosef.adhitya.d@mail.ugm.ac.id

Abstract. The improvement of automation and big data analytics with technology is giving big benefits to the agriculture sector. In the open field crops, farming robot technology helps farmers to spray fertilizer. Soil data analysis can help to determine plant treatment. Harvesting robots for picking fruits with computer vision and robotics can support the productivity and quality of crops. To implement this, appropriate hardware and algorithms are important to define. One of the most premium fruits that can be cultivated by using high technology is strawberries. We need to declare the best trade-off between system design (software & hardware) with implementation especially in agricultural sector. In this experiment, the YOLOX algorithm is running to detect the ripeness of strawberries. The algorithms run in two modes: GPU and CPU only. The best results show that the YOLOX-S algorithm, which runs in GPU mode, is 95.75% in precision and 59 fps in throughput. It will be difficult to accommodate a harvesting robot processor that has a GPU. The algorithm is now run in CPU only mode and it gives only 11.31 fps in throughput. Then the proposed model, which is already optimized by Intel OpenVINO, gives better results in throughput, showing 37.15 fps. So, with the proposed optimized model, we can choose CPU-only hardware for affordable hardware implementation.

Keywords: Big Data · YOLOX · Intel OpenVINO · Fruit Ripeness Detection

1 Introduction

In a nation like Indonesia, agriculture is the backbone of the economy and a major source of jobs. The second-largest economic support sector in Indonesia, this industry contributes 13.28% of the country's GDP [1]. A nation that ignores this industry will weaken as a result of losing its sustainable economic base. It is important for both local and international trade in the majority of the nation. Today, however, there is a need to boost agricultural productivity due to the population expansion. Therefore, we must enhance agriculture sector by finding more effective ways to boost crop output while spending less money and making better use of the resources already at our disposal.

Most of farmers in Indonesia still use the traditional way of farming. They reluctant to use advanced technologies because of low access in knowledge, high cost investment, and

J. Sumantyo et al. (Eds.): ICoSIA 2022, ABSR 29, pp. 46–54, 2023.
https://doi.org/10.2991/978-94-6463-122-7_5

they are unfamiliar with the big advantages of using technologies. Various problems and challenges in agriculture fields, nowadays can be solve by using blockchain technology such as cloud computing, internet of things, machine learning and also deep learning. Building neural networks to replicate the human brain for analytical learning is the goal of deep learning, a subset of machine learning. Applications of computer vision and machine learning can be used to solve complex problems like image segmentation, object detection, image recognition and data analysis for crops and cultivation process.

As one of the most consumed berries worldwide, strawberries may be cultivated in either open fields or enclosed structures with controlled environments, such as greenhouses and polytunnels [2]. The most labor and time-intensive stage of strawberry production is harvesting. More than 75 percent of the overall manufacturing expenses are labor-related, and this percentage keeps rising each year [3]. Since harvesting is a labor-intensive process, saving time and money via the use of robotic harvesters is an important goal of this study. In harvesting process the key activities or factors that can solved by using technologies are fruit/crop sizing, detect skin color and maturity stage, fruit detection and classification [4].

2 Literature Review

Several strawberry ripeness identification methods have been proposed and tested. This method leads to decisions in harvesting robot. In general, several methods proposed to add some pre or and post processing techniques, doing some compression techniques to reduce the computational workload, comparing several algorithms to meet their hardware setup, and modifying internal structure of a method so they can increase the computational speed.

For use with inexpensive SBCs like the Raspberry Pi 3B, Nikolas Lamb and colleagues developed a fruit detector for strawberries. This method achieves fast inference with good accuracy by experimenting with several optimization strategies. The dataset composed of BRG images with high resolution which contain of ripe strawberries. Manual annotation in bounding boxes was did to identify each piece of fruit in each image. About three minutes video extracted into 4550 images and containing 22.662 annotated strawberries. Then compressed from 1080 $\times$ 1920 pixel to 360 $\times$ 640 pixel and color masking was applied. The dataset is randomized, with 60% used for training, 20% for validation, and the remaining 40% for testing. Successful execution on a Raspberry Pi 3B resulted in a detection rate of 1.63 frames per second with an average accuracy of 0.842 [5].

A study comparative of traditional computer vision and deep learning for detection of strawberries was proposed by Pierre Huseb et al. This research was explore 3 different method of object detection of strawberries in image. The first proposed method was a traditional computer vision and uses primarily a segmentation algorithm. Deep learning framework that uses a single pass of neural network was presented for the two other ones. This research concludes the more suitable methods to detect strawberries in images was the deep learning methods rather than the traditional computer vision implemented. The best result in this research showed a mAP of 87.1% and average segmentation accuracy of 86.6% for YOLOv3-strawberry. The speed performance was tested in the Google Colab and gave 0.1 fps when run in CPU mode and 16.6 fps when in GPU mode [6].

Implementation SSD-MobileNet convolutional network module on a harvesting robot powered by SBC Jetson Nano as object detection quality of strawberries was proposed by Muh Fauzan Irwan et al. in 2021.The capacity of a robot harvester to identify strawberry quality in real time is the topic of this study. The RGB photographs were separated into two dataset folders, each containing images with a 1280 × 720 pixel aspect ratio. There were 114 training photos, 28 validation images, and 36 test images in the first folder. There were 136 training photos, 36 validation images, and 42 test images in the second folder. Thus, the detector achieved over 90% accuracy in recognizing and discriminating strawberry quality at 40-60fps in actual camera streaming [7].

Yanchao Zheng et al. suggested a study in 2021 with the stated goal of enhancing the performance of lightweight deep neural networks for use on embedded platforms. The information was gathered by flying a drone at a height of 3 m over the 67 by 6 m experimental area with 5 rows of strawberry plants. Using MIT-developed labelling software, three more classes were manually added to the labelling process: mature, immature, and bloom. The training environment consisted of an AMD Ryzen 5 3600 with 6 cores and an NVIDIA RTX 3070 graphics processing unit. Then the performance of 4 network were evaluated in a Jetson Nano ARM Cortex A57 with MaxWell 128-core GPU. The purpose of this study was to find a way to speed up the detection inference without sacrificing the detection accuracy of strawberries, therefore a modified version of the lightweight YOLOv4-tiny was developed, with decreased layers and a modified structure known as RTSD-Net. In a head-to-head comparison with YOLOv4-tiny, RTSD-Net was shown to be less accurate by 0.62% while also being faster by 25 fps, or 25.93% more so than YOLOv4-tiny. Smart strawberry harvesting equipment and edge computing are two areas where the projected RTSD-Net has been said to shine [8].

3 Methods

The experiment was carried out in 3 steps. First step, from raw video we make label to get our dataset. After that, the dataset then trained with YOLOX algorithms to get model. Evaluation are made by running in GPU mode and CPU only then evaluated for computational speed, precision and recall rate. The experimental model then optimized with the OpenVINO on Intel Dev Cloud workbench.

3.1 Intel OpenVINO Model Optimizer

OpenVINO (Open Visual Inference and Neural Network Optimization) plays an important role in this research. This toolkit is a cross platform deep learning toolkit that developed by Intel. OpenVINO toolkit can covers computer vision and natural language processing application. OpenVINO can accelerate AI workloads and speed up throughput by doing pre-optimized on Intel hardware platform. As we can see on Fig. 1 the first step that we should do in this research is train the dataset to make a model. We can do this by own or take pre-trained model on Intel open model zoo. After that model optimizer will processing your model and produces an Intermediate Representation (IR) of the network.

Fig. 1. OpenVINO is an open-source toolkit for optimizing and deploying AI inference [10]

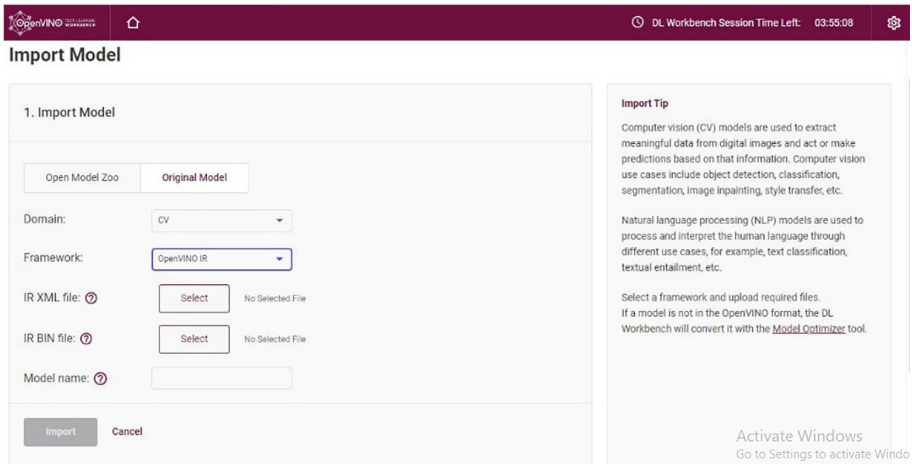

Fig. 2. DL Workbench user interface with option upload our own model and Intel Open Model Zoo.

The models optimized with several option techniques such as quantization, freezing, fusion and more.

This research begins with setting up the DL workbench on Intel Dev Cloud by creating new project. We can visualize, adjust, and compare the performance of deep learning models on various Intel architecture configurations using the web-based graphical environment known as DL Workbench. If we want to make project with our own model, several files need to be uploaded. There are 2 options while importing model. First, we can choose from Intel OpenVINO model zoo, or we can originally input our model. Supported model formats for this DL workbench are ONNX, TensorFlow, Pytorch, MXNet, Caffe, Kaldi, and OpenVINO IR (Intermediate Representation). The user interface of DL workbench on Intel Dev Cloud shown in Fig. 2. After the model uploaded successfully, then we can run the model optimization and get the result.

3.2 Dataset

In this research, the experiment used strawberry dataset from dataset [6] for training and testing. The dataset was consisting from 352 frame photos, ground truth shows 90 ripe

Table 1. Experiment Performance Result While Running in CPU only mode.

	YOLOX-X	YOLOX-S
CPU	29, 38, 94, 14, 72, 63, 18, 82	77, 42, 62, 33, 39, 73, 42, 84
GPU	0 GB	0 GB
RAM	6.04 GB	5.03 GB
FPS	2	18

strawberries and 101 unripe strawberries. The dataset then labelled into 2 categories, mature strawberry and immature strawberry using Label Studio [12]. Ubuntu 18.01 served as the OS for the training environment, the CPU is an 11th-gen Intel Core i5 8-core processor, and an NVIDIA GTX 1650 GPU was employed to speed up model training for comparison purposes. Version 11.6 of CUDA is currently installed in this environment. Models were constructed and trained using the YOLOX [13] framework in this experiment. YOLOX is the anchorless version of YOLO, simpler in design but with better performance exceeding yolov3 v5 with OpenVINO supported. Their goals is to bridge the gap between research and industry. Maximum batch size was 8, learning rate was le-2, and epochs in training were set at 1000.

3.3 Performance Evaluation

Mean average precision or mAP is a universal metric used to find artificial neural networks. mAP is the average value of the mean value of precision (AP). Although good at evaluating object detection, mAP is not suitable for evaluating segmentation because it requires confidence scores and bounding boxes. The following are some of the metrics that will be used to evaluate segmentation, IoU, accuracy and precision. IoU is the intersection over unity and is a measure of how much overlap there is between the prediction and the underlying truth. It is a metric that combines accuracy and precision in one. Accuracy is the intersection of the underlying truth and indicates how much of the underlying truth is covered by the prediction. Precision is the intersection of predictions and measures how much of the prediction is covered by the ground of truth. Last performance evaluation method is latency. It will measure the inference time (in milliseconds) required to process one input.

4 Experimental Result

The experiment was taken in 3 steps. First, two YOLOX algorithm are running in GPU mode then evaluated the computational speed. Second step, the algorithms then running in CPU only mode then evaluated for the computational speed and the precision and recall rates. Third step, the model of this experiment then optimized by OpenVINO toolkit by choosing a configuration does not compromise on decreasing accuracy. Performance assessment was given by the toolkit.

Table 2. Experiment Performance Result While Running in GPU mode.

	YOLOX-X	YOLOX-S
CPU	62, 4, 7, 5, 42, 14, 18, 22	73, 42, 49, 42, 30, 67, 59, 83
GPU	1.431 GB	1.013 GB
RAM	7.19 GB	7.10 GB
FPS	10	58

Table 3. Confusion matrix of YOLOX-X.

	Predicted unripe	Predicted ripe
Actual: Unripe	99	8
Actual: Ripe	2	82

Table 4. Precision and Recall Rate of YOLOX-X.

	Ripe strawberries	Unripe strawberries	Overall
Precision rate	91.11%	98.01%	94.56%
Recall rate	97.59%	92.52%	95.05%

From Table 1, the data shown that algorithm run in CPU only mode for YOLOX-X only give speed of 2 fps. While YOLOX-S can give for 18 fps which is 9 times faster than YOLOX-X (Table 2).

Target detection performance of the YOLOX algorithm was measured using the precision (P) and recall (R) rates:

$$P = \frac{TP}{TP + FP} \tag{1}$$

$$R = \frac{TP}{TP + FN} \tag{2}$$

In where TP is the total number of ripe items that have been accurately identified. The false positive rate (FP) is the amount of improperly matured cases. The fraction of instances that should have been marked as ripe but weren't denoted by the symbol FN. The experiment's confusion matrix was manually counted, and the results reveal that the ground truth for ripe strawberries is 90 pcs, whereas the ground truth for unripe strawberries is 101 pcs. It shown in Table 3 for YOLOX-X and Table 5 for YOLOX-S.

From given data in Table 3 we get the TP is 82 and the FP is 8. So, we can get the precision of YOLOX-X for the ripe strawberries is 91.11%. For the unripe strawberries, the TN is 99 and the FN is 2. So, we can get the precision for the unripe strawberries is 98.01%. The overall precision rate is 94.56%. Recall rate for the ripe and unripe

Table 5. Confusion matrix of YOLOX.S

	Predicted unripe	Predicted ripe
Actual: Unripe	98	5
Actual: Ripe	3	85

Table 6. Precision and Recall Rate of YOLOX-S

	Ripe strawberries	Unripe strawberries	Overall
Precision rate	94.49%	97.02%	95.75%
Recall rate	96.59%	95.14%	95.86%

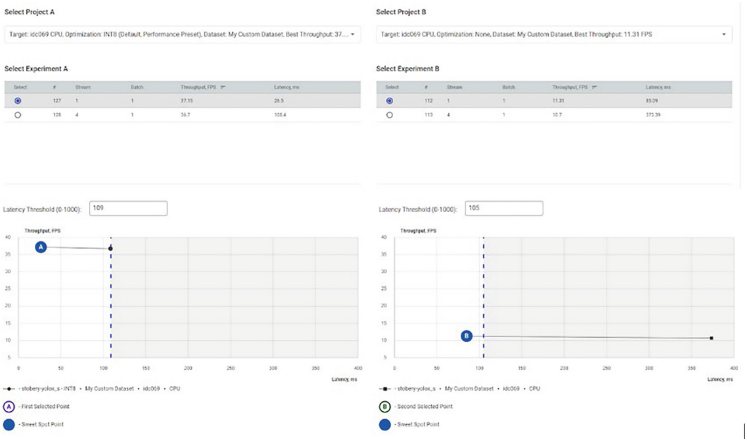

Fig. 3. Comparison of no optimization and optimized model run in Intel DL Workbench

strawberries can be calculate from the given confusion matrix. So, we get result overall recall rate for YOLOX-X algorithm is 95.05%. The others are completely shown in Table 4.

From given data in Table 5 we get the TP is 85 and the FP is 5. So, we can get the precision of YOLOX-S for the ripe strawberries is 94.49%. For the unripe strawberries, the TN is 98 and the FN is 3. So, we can get the precision for the unripe strawberries is 97.02%. The overall precision rate is 95.75%. Recall rate for the ripe and unripe strawberries can be calculate from the given confusion matrix. So, we get result overall recall rate for YOLOX-S algorithm is 95.86%. The others are completely shown in Table 6 (Fig. 3).

In the last step of this experiment, we tried to run in CPU only mode but concern in high computational speed. So with Intel Dev Cloud, we uploaded the same model with last experiment then doing optimization there. The basic and not compromise on

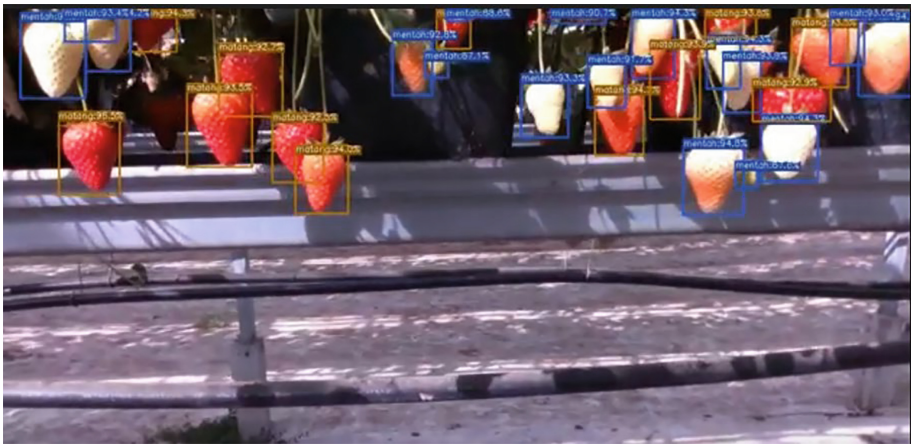

Fig. 4. Strawberries ripeness detection. In this image, the ripe and unripe strawberries can be distinguished by 2 boxes of different colors

decreasing accuracy configuration was selected for the optimization method. The last experiment run in the Intel DL Workbench with same hardware specification and the computational speed get almost the same at 11 fps. The example of detection result can be shown in Fig. 4 that shows the computational speed of optimized model at 37.15 fps. It's get better than before for 3.37 times faster.

5 Conclusion

This paper compares two existing YOLOX algorithms. From the experiment, the YOLOX-X algorithm obtained a level of precision of 94.56% with a speed of 2 fps when running on a processor that has no support from the GPU. Meanwhile, when run with the GPU mode, this method provides a speed of 10 fps. In the second experiment using the YOLOX-S algorithm, it gave better results. For the mode without GPU assistance, this method obtained a precision level of 95.75% and a speed of 11 fps. When it runs in GPU mode, the speed increases to 59 fps. This increase in computational speed creates hardware costs that need to be spent on GPU accelerators. Then a new method was proposed with model optimization by using the Intel OpenVINO toolkit and choosing uncompromised on the level of accuracy. As a result, in CPU mode, the execution speed of the YOLOX-S algorithm can increase to 37.15 fps.

6 Limitations and Future Studies

Limitation for these research that conducted only for implementation based on two different CPU and GPU condition. Implementation based on algorithm that tested will be conduct for robotic automation harvest unit for future studies and implementation.

References

1. B. P. S. RI, Ekonomi indonesia triwulan ii-2022, 2022.
2. Y. Xiong, Y. Ge, L. Grimstad, and P. J. From, Journal of Field Robotics 37, 202–224 (2020).
3. J. Linnan et al., J. China Agricult. Univ. 24, 61–68 (2019).
4. R. Prange, Acta Horticulturae 933, 43–5003 (2012).
5. N. Lamb and M. C. Chuah, "A strawberry detection system using convolutional neural networks," in 2018 IEEE International Conference on Big Data (Big Data) (2018), pp. 2515–2520.
6. P. Huseb, "Automatic strawberry detection: A comparative study of traditional computer vision and deep learning for detection of strawberries." Master thesis, Norwegian University of Science and Technology (2019).
7. M. F. Ridho and Irwan, "Strawberry fruit quality assessment for harvesting robot using ssd convolutional neural network," in 2021 8th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI) (2021), pp. 157–162.
8. Y. Zhang, J. Yu, Y. Chen, W. Yang, W. Zhang, and Y. He, Computers and Electronics in Agriculture 192, p. 106586 (2022).
9. Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, Yolox: Exceeding yolo series in 2021, 2021.
10. OpenVINO, Model Optimizer Developer Guide, 2022. https://docs.openvino.ai/2021.3.html (accessed Nov. 15, 2022).
11. I. OpenVINO, Intel openvino docs, 2022..
12. M. Tkachenko, M. Malyuk, A. Holmanyuk, and N. Liubimov, Label Studio: Data labeling software, 2020–2022, open source software available from https://github.com/heartexlabs/label-studio.
13. Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, arXiv preprint arXiv:2107.08430 (2021).
14. Y. Yu, K. Zhang, H. Liu, L. Yang, and D. Zhang, IEEE Access 8, 116556–116568 (2020).
15. Demidovskij, Y. Gorbachev, M. Fedorov, I. Slavutin, A. Tugarev, M. Fatekhov, and Y. Tarkan, "Open-vino deep learning workbench: Comprehensive analysis and tuning of neural networks inference," in 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW) (2019), pp. 783–787.
16. Demidovskij, A. Tugaryov, A. Suvorov, Y. Tarkan, M. Fatekhov, I. Salnikov, A. Kashchikhin, V. Gol- ubenko, G. Dedyukhina, A. Alborova, R. Palmer, M. Fedorov, and Y. Gorbachev, "Openvino deep learning workbench: A platform for model optimization, analysis and deployment," in 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI) (2020), pp. 661–668.