

A dataset for computer-vision-based fig fruit detection in the wild with benchmarking you only look once model detector

Adi Izhar Che Ani¹, Mohammad Afiq Hamdani Mohammad Farid¹, Ahmad Shukri Firdhaus Kamaruzaman¹, Sharaf Ahmad², Mokh Sholihul Hadi³

¹Centre for Electrical Engineering Studies, College of Engineering, Universiti Teknologi MARA Cawangan Pulau Pinang (UiTM CPP), Penang, Malaysia

²Department of Applied Sciences, Universiti Teknologi MARA Cawangan Pulau Pinang (UiTM CPP), Penang, Malaysia

³Department of Electrical Engineering, Universitas Negeri Malang, Malang, Indonesia

Article Info

Article history:

Received Jan 6, 2023

Revised Oct 13, 2023

Accepted Feb 12, 2024

Keywords:

Agricultural automation

Deep learning

Fig fruits

Image dataset

Object detection

ABSTRACT

The image datasets that are most widely used for training deep learning models are specifically developed for applications. This study introduces a novel dataset aimed at augmenting the existing data for the identification of figs in their natural habitats, specifically in the wilderness. In the present study, researchers have generated numerous image datasets specifically for object detection focus on applications in agriculture. Regrettably, it is exceedingly difficult for us to obtain a specialized dataset specifically designed for detecting figs. To tackle this issue, a grand total of 462 photographs of fig fruits were gathered. The augmentation technique was utilized to substantially increase the size of the dataset. Ultimately, we conduct an examination of the dataset by doing a baseline performance study for bounding-box detection using established object detection methods, specifically you only look once (YOLO) version 3 and YOLOv4. The performance obtained on the test photos of our dataset is satisfactory. For farmers, the capacity to identify and oversee fig fruits in their natural or developed environments can be highly advantageous. The detecting device offers instantaneous data regarding the quantity of mature figs, facilitating decision-making procedures.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Mokh Sholihul Hadi

Department of Electrical Engineering, Universitas Negeri Malang

Malang, Indonesia

Email: mokh.sholihul.ft@um.ac.id

1. INTRODUCTION

Fruit identification is the cornerstone of agricultural automation, upon which fruit positioning, automated harvesting, and yield estimation are developed. Human visual inspection is the most typical method for identifying fruits in the wild or an orchard environment. This conventional technique of detection would need a lot of labour and time. Due to the rapid development of algorithms [1] for artificial intelligence and image sensor technology, online automatic fruit recognition has lately emerged as a new trend [2]. They allow farmers to manage and maximize their resources while making wise decisions during harvest [3]. Artificial intelligence systems used for fruit detection on digital images offer several advantages, including a high degree of accuracy, user-friendliness, and real-time performance that is both efficient and effective [4], [5]. The development of technology has driven the idea of fig detection [6].

As we all know, figs are famous due to their excellent nutritional, health, and therapeutic properties. As a result, figs have been actively collected worldwide in recent years. In 2013, the fig production area was

predicted to be 358,494 hectares, with a yield of 1,117,452 t. Fig trees are mostly found in Turkey around the Black Sea, the Marmara area, the Aegean, and Mediterranean coasts. As the market is excitingly growing, the size of fig production has increased further [4]. Fig fruits are less distinguishable in color from the background, with thicker branches and leaves than the other fruits. Thus, effective recognition and placement of fig fruits may provide significant technical support for intelligent orchard technologies such as yield predictions and automated picking [7].

The detection of fruit has been undertaken by researchers with a broad spectrum of sensor technologies and algorithms; however, cameras and computer vision techniques are the most effective combination [8]. Unfortunately, using computer vision technology in outdoor orchard settings comes with its challenges, including the following: i) varying brightness conditions and ii) occlusion of fruits by other leaves, branches, or other fruits. As a result, detecting fruits has become increasingly challenging, leading to the development of deep learning algorithms to automate processes. However, because there are no standardized benchmark datasets or testing standards in precision agriculture, it is hard to compare different methodologies directly. Benchmark datasets have received much attention and are driving computer vision research [5]. ImageNet, pascal visual object classes (VOC), and the common items in context (COCO) dataset are well-known datasets in computer vision that contain many images organized into various categories. However, there is a shortage of resources for datasets relevant to fig fruits, and most datasets developed to identify figs are in general contexts.

The key reason leading to the lack of research activities is the lack of publicly available data annotated with information on the ground truth. It has become a severe bottleneck in developing fig fruit recognition, particularly in deep learning models, which depend heavily on massive training data. In this work, we provide a dataset that attempts to overcome this limitation. Furthermore, a significant gap in the academic community must be filled by datasets of fig images captured in natural settings and supported by standardized analysis methods. We have great expectations that this dataset will be a critical step in advancing the agriculture field.

2. RELATED WORK

Datasets have consistently been crucial in advancing image-processing research. They offer a technique for instructing, assessing algorithms, and propelling study in novel and demanding domains. The majority of computer vision approaches depend on extensive datasets for the purpose of training, testing, and assessing different solutions to issues [9]. They offer the resources to educate and assess novel algorithms, facilitating a direct comparison of the outcomes [10], [11]. Ultimately, they enable researchers to address novel and increasingly complex research problems [12]. ImageNet [13] and COCO [14] are widely recognized as the most popular image datasets. These datasets were made freely accessible by their respective developers, the ImageNet large scale visual recognition challenge (ILSVRC) and Microsoft. ImageNet, COCO, and pascal VOC [15] are three datasets that have facilitated the advancement of image classification and object segmentation by providing a vast collection of annotated photos to the public.

Over the past few years, the ImageNet dataset [8], consisting of a large number of images, has significantly enhanced the precision of research utilizing neural networks [4], [16], [17] for the purposes of image categorization and object recognition. In the future, the release of the COCO database, which aims to identify non-iconic objects, could enable researchers to do more precise object recognition, instance segmentation, image captioning, and human keypoint localization [14]. Google has released open images V4 [13] a newly available dataset. The dataset comprises nine million photos that have been annotated at the image level and include corresponding item bounding boxes. The datasets listed contain photos that are intended for widespread utilization. This means that the objects, perspectives, and applications depicted in the images are representative of typical circumstances.

The initial stage in estimating yield or harvesting fruit involves the detection of the fruit. Several studies have been published that focus on fruit detection, using different datasets. Yamparala *et al.* [18] introduced a method for automatically classifying fruit diseases in order to facilitate identification. The experiments are conducted using a dataset consisting of 200 photographs of various fruits. Specifically, there are 50 images of apples, 50 images of mangoes, 50 images of oranges, and the remaining 50 images depict grapes. Neural networks are trained using the size, color, and form of the fruits. In addition, the researchers of reference [19] developed a deep learning model to classify mango and pitaya fruits. The input consists of authentic data provided by fruit producers, which includes a total of 700 photographs of mangoes and 700 photographs of pitaya fruits. In their publication, Jian *et al.* [20] introduced an algorithm for the optical detecting system used in agricultural robots for fruit harvesting. This algorithm is capable of identifying and locating different types of fruits in diverse environments. The dataset comprises photos depicting apples, oranges, and bananas, with detection accuracies of 96.5%, 97.6%, and 82.3%, correspondingly.

In their study, Zhao and Qu [21] propose techniques for identifying both sound tomato fruits and those affected by prevalent physiological disorders. The dataset underwent several improvements to optimize network performance: initially, the picture datasets were enriched with additional data to mitigate the risk of overfitting. As a result, augmented datasets were acquired, which encompassed 1000 photographs of tomato fruits. Furthermore, a grayscale processing module and a foreground extraction module were created to assess the importance of picture data type. This report advocated for the employment of data augmentation in our project. Yijing *et al.* [4] introduced a deep learning approach to identify fig fruits. Their method utilized a dataset consisting of 913 photos. The training set consisted of 70% (639 photos), while the test set consisted of 30% (274 images). Nevertheless, the dataset utilized is confidential.

3. METHOD

In this method, a dataset of 462 fig fruit images was gathered in Tasik Gelugor, Penang, capturing variations in fruit quantity and shading using a Nikon DSLR camera. The dataset was meticulously annotated using Labellmg software [22], creating bounding boxes for ‘buah tin’ (fig fruit) and saving annotation information in PascalVOC [23] format “.xml” files. Data augmentation was employed to enhance dataset diversity and prevent overfitting, which included 90° rotations, resizing to 416×416, brightness and noise variations, resulting in 1110 augmented samples. The dataset was then split into a 70/30 training/testing ratio, allocating 70% (972 images) for training and 30% (138 images) for testing, setting the stage for robust deep learning applications and performance assessment.

3.1. Dataset acquisition

The fig fruit photos were gathered in Tasik Gelugor, Penang, with a sample size of 462 images, as depicted in Figure 1. The sample collection comprises photos that vary in terms of fruit quantity and shade intensity, hence augmenting the diversity of the samples. The fig fruit photographs were captured with a Nikon DSLR camera with a resolution of 4608×3072 pixels, resulting in a spatial resolution of 300 dpi. Consequently, these photos are suitable for applications that demand great resolution and quality. Every image is unique in terms of the quantity of fruits, perspective, and level of shading. In addition, certain photos depict fig fruits that are half concealed, shrouded by another object, or just partially perceptible. To enhance identification during testing and training, the dataset’s diversity is augmented by incorporating a complex backdrop environment.



Figure 1. Sample of fig fruit images collected

3.2. Dataset annotation

The process of manually annotating the photos of fig fruits is crucial in order to obtain precise data parameters once a significant amount of photographs has been collected in the dataset. The image annotation tool utilized for this task was the Labellmg software (refer to Figure 2). The smallest external rectangular box of the fig fruit was chosen as the actual box for annotation in order to minimize the number of background pixels within the box. This stage employs Python programming to provide a function for selecting a frame. Every fig fruit image in the collection has been labeled as ‘buah tin’. To clarify, all the bounding boxes of the figures seen in the image were recorded in the PascalVOC format, as depicted in Figure 3. Following the annotation stage, the application produces .xml files for every annotated image, containing data such as the coordinate values for the bounding boxes of each lesion on the fig fruits, including their height and width.

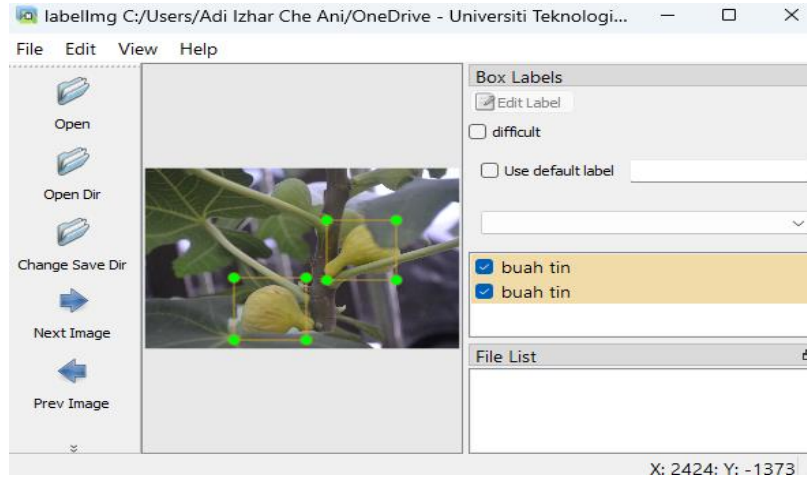


Figure 2. Image annotation process using Labelimg



Figure 3. Detailed information of a labelled fig fruit image. The green, blue, and red rectangles indicate the size of the image, fruit location, and the object name based on the image on the left, respectively

3.3. Dataset augmentation

Data augmentation is one method for increasing the dataset by changing the diversity of the image. Besides, it is also overcoming the problem of overfitting in training [21]. Overfitting happens when random noise or mistakes are reported rather than the underlying relationship. In this process, several techniques are used for data augmentation operations which are 90° rotation transformations clockwise, counterclockwise, and upside-down. Next, the images are resized to 416×416 and contain disturbances of brightness and noise. Furthermore, the brightness is set between -25% and +25%; meanwhile, the noise is up to 5% of pixels. Due to that reason, three new fig fruit images are generated from each image through the operations stated, as shown in Figure 4. In Figure 4(a) the original image serves as a reference point. Figure 4(b) demonstrates the resized image with both reduced brightness and low noise, while Figure 4(c) showcases the image after a clockwise rotation, combined with low brightness and heightened noise. Finally, Figure 4(d) illustrates the resized image post-clockwise rotation, exhibiting increased brightness and noise levels. These augmentation techniques serve to diversify the dataset, helping the deep learning algorithm learn from a range of variations, ultimately enhancing its robustness and ability to handle different real-world scenarios.

Therefore, new 1110 augmented dataset samples were obtained. With additional images following data augmentation, it would benefit the algorithm as it may learn as many irrelevant patterns as possible throughout the training step, avoiding overfitting and achieving better performance. Then, the dataset will be split into the most commonly adopted deep learning applications, a 70/30 ratio of training and testing, respectively. Hence, 70% (972 images) were selected for training, and 30% (138 images) were selected for testing. Figure 5 shows the train images, which consist of the raw image, the augmented image with ground

truth and the heatmap. Figure 5(a) reveals the raw image, offering insight into the original visual data. Figure 5(b) displays an augmented image with ground truth, which incorporates the results of the data annotation process, enhancing the image's informative value. Finally, in Figure 5(c), we observe a heatmap that highlights a single annotation, showcasing the spatial distribution of important features in the image. This figure serves as a valuable reference for understanding the stages of data processing and annotation, illustrating how raw images are transformed into informative, annotated representations with ground truth information.

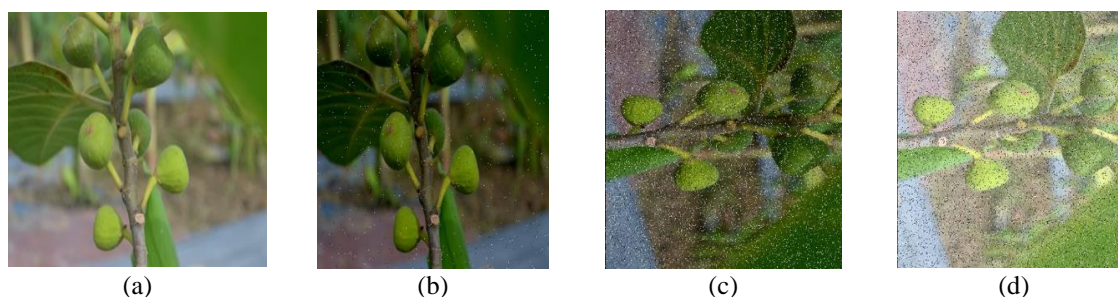


Figure 4. Fig fruit image augmentation: (a) original image; (b) resized picture with reduced brightness and noise; (c) resized image with clockwise rotation, reduced brightness, and increased noise; and (d) adjust the size of the image by rotating it in a clockwise direction and increase the brightness level with a high amount of noise

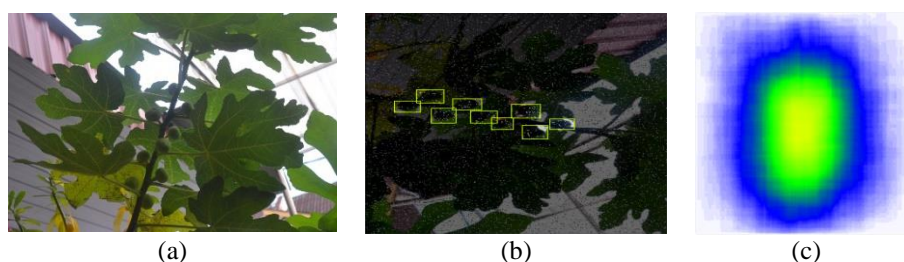


Figure 5. Sample of train image; (a) raw image; (b) augmented image with ground truth; and (c) heatmap of a single annotation

4. RESULTS AND DISCUSSION

4.1. Related object detectors

Object identification techniques based on deep learning can be categorized into two types: region-based two-stage target detection methods, exemplified by the region-based convolutional neural network (R-CNN) series and regression-based one-stage target detection algorithms, such as you only look once (YOLO) and single shot detector (SSD). R-CNN employs the selective search method to extract around 2000 region proposals from the uppermost to the lowermost parts of the image. Subsequently, it retrieves characteristics for every suggested region and categorizes them employing support vector machines (SVM). Subsequently, it conducts boundary regression on the proposed regions. This strategy is segmented into multiple sections, and the process of preparing the area proposal is excessively time-consuming [7].

Faster-R-CNN improved upon the R-CNN by replacing the original selective search method with region proposal network (RPN) to generate region proposals. This change reduced the amount of region proposals from approximately 2,000 to 300, resulting in improved overall quality. In order to enhance performance and optimize operations, faster-RCNN utilizes shared convolutional layers with RPN and Fast R-CNN [7]. An example of this is the approach devised by [19] to classify mango and pitaya fruits using Faster R-CNN. The input consists of authentic data provided by fruit producers, which includes a total of 700 photographs of mangoes and 700 photographs of pitaya fruits. The proposed approach has an accuracy score of around 99%. This methodology is suitable for developing a systematic process for efficiently categorizing a large quantity of fruits in real-time, with the aim of preserving the fruits' overall quality.

Several neural networks, including YOLO, can extract the bounding boxes of multiple kinds of items from an image and proceed to the next stage. YOLO is an all-in-one network that does feature

extraction, localization, and classification. Therefore, the YOLO series demonstrates exceptional velocity and is better suited for real-time detection in comparison to the R-CNN series. As an illustration, YOLOv3 exhibited a considerably higher detection speed, achieving a frame rate of eight times that of faster R-CNN. YOLOv4 outperforms other detectors in terms of both speed and accuracy, surpassing the quickest and most accurate ones [4], [24], [25]. Consequently, this work utilizes YOLOv3 and YOLOv4 to build fig fruit identification models for real-world scenarios.

4.1.1. YOLOv3

YOLOv3 is an upgraded iteration of the YOLO technique that divides the input image into a grid of $S \times S$ dimensions, where S is equal to 13. Every grid is assigned the responsibility of identifying an object that is situated within it. Each grid consists of three border boxes with distinct initial sizes. K-means clustering determines the initial dimensions of each anchor box. The YOLO v3 model produces five predictions for each bounding box: t_x , t_y , t_w , and t_h represent the coordinates of the bounding box, while confidence is the score indicating the likelihood of an object being present [26]. The YOLO v3 model employs logistic regression to forecast objectness scores for every bounding box. If the previous bounding box overlaps with a ground truth object and has a greater size than the other prior bounding box, it is given a score of 1. Bounding box priors that do not possess the best quality but nevertheless exhibit an overlap with a ground truth object surpassing a specific threshold will be discarded. Before allocation, a bounding box is assigned to each object to accurately represent the ground truth. The YOLOv3 network architecture is seen in Figure 6.

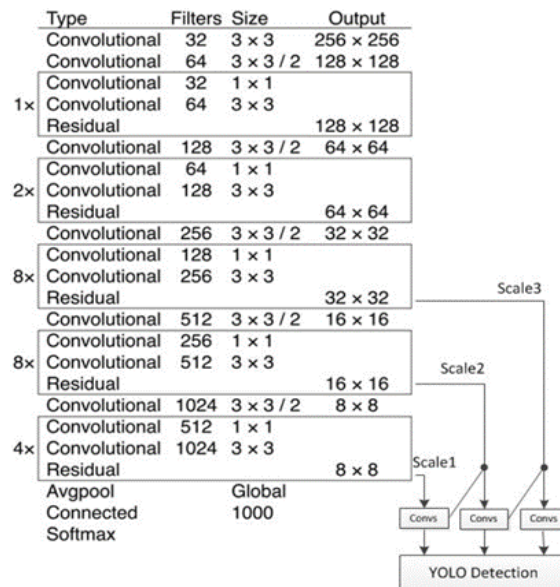


Figure 6. YOLOv3 network architecture

The YOLO v3 model currently produces a three-dimensional tensor that contains encoded data for bounding boxes, objectness scores, and class predictions. The tensor is of size $NN[3(4+1+C)]$, where NN denotes the number of bounding box offsets, 4 denotes the number of bounding box coordinates, 1 is the objectness prediction, and C denotes the number of class predictions. The utilized feature extraction network is darknet-53, which consists of a total of 53 convolutional layers. The darknet-53 employs a convolutional neural network structure comprising of 33 and 11 convolutional layers, in addition to a few shortcut connections. The YOLOv3 algorithm utilizes convolution and batch normalization procedures to predict boxes at three different scales based on the input image. Object detection is performed on scales of 13×13 , 26×26 , and 52×52 when the supplied image has a size of 416×416 . YOLOv3 integrates three scales using a comparable approach to feature pyramid network (FPN). Multi-scale detection enables the acquisition of features from different resolutions, providing an advantage in recognizing small target objects [24].

4.1.2. YOLOv4

YOLOv4 is an enhanced iteration of YOLOv3 that successfully accomplishes target localization and target classification prediction. The one-stage technique transforms the challenge of determining the location

of the object bounding box into a regression problem, resulting in a substantial enhancement in detection speed while retaining a predetermined level of accuracy. Contrary to two-stage target identification techniques like faster RCNN, YOLO employs a solitary convolutional neural network to ascertain the category and location of the regression target over the entire image [25], [26]. The YOLO network comprises three primary components: backbone, neck, and head. The term “Backbone” refers to the convolutional neural network that collects and produces visual features at various degrees of detail in the image. The neck is a constituent of the network layer that consolidates and integrates visual attributes. The primary objective of this is to integrate feature information from feature maps of different sizes and supply image data to the prediction layer. The head is a neural network that possesses the capability to predict picture attributes, produce bounding boxes, and categorize images [4], [26]. Figure 7 illustrates the network structure of YOLOv4.

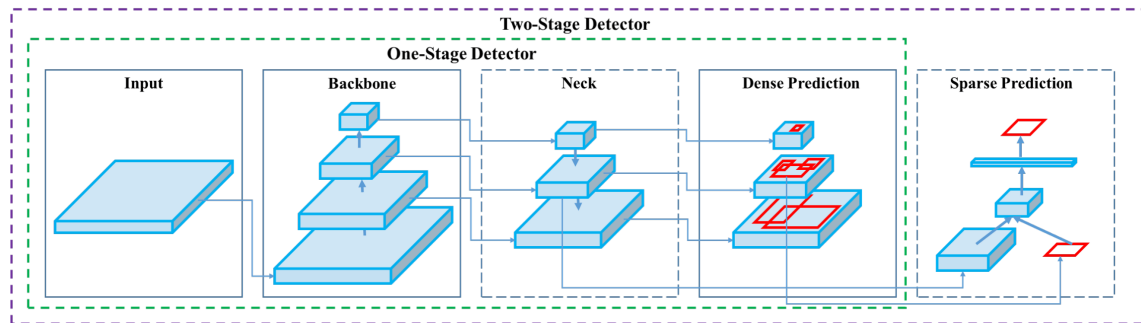


Figure 7. YOLOv4 network architecture

The YOLOv4 model utilized the CSPDarknet53 design as its core, effectively concealing unnecessary gradient information while optimizing the network. The incorporation of the cross-stage partial (CSP) module in YOLOv4 facilitates the assimilation of all gradient modifications into the feature map. This leads to a decrease in the number of parameters and computational requirements of the model, while yet preserving accuracy. The Mish activation function is utilized in the backbone to amplify the infiltration of input into the neural network, leading to enhanced accuracy and generalization capabilities. PANet functions as a module for merging features in the neck component of YOLOv4. The objective is to address the issue of one-way feature integration in YOLO v3's FPN by including bottom-up feature fusion. In addition, the spatial pyramid pooling (SPP) technique is used as an extra component to broaden the scope of information received, extract significant contextual data, enhance the overall and local accuracy of fruit identification in complicated environments, and enhance the efficiency of detection [4].

4.2. Training platform

The training procedure of the two proposed models will be conducted using Google Colab, with the programs written in the Python environment. Both variants of the proposed YOLO utilize the Darknet framework, which is an open-source neural network framework built in C and CUDA. Darknet is known for its speed, ease of installation, and support for CPU or GPU processing. The training aims to develop a model for fig fruit detection.

4.3. Performance evaluation

The assessment measures utilized in this study to assess the detection model include mean average precision (mAP), precision, recall, and F1 score. The measures used to evaluate performance are true positive (TP), true negative (TN), false positive (FP), and false negative (FN). True positive rate (TPR) is the ratio of correctly identified positive samples to the total number of samples with positive outcomes. The image contains a fig fruit, and the computer successfully identifies it as 'buah tin'. FP is the proportion of genuine negative samples with expected positive outcomes. There is no depiction of a fig fruit in the image, yet the system correctly identifies it as 'buah tin'. FN represents the count of genuine negative samples that resulted in negative predictions. Although the image contains a fig fruit, the algorithm fails to identify it as 'buah tin'. The TN value is not considered in our performance metrics calculation [25].

4.3.1. Precision

Precision refers to the proportion of correctly anticipated positive cases out of all the projected positive cases. It quantifies the level of precision of the model in performing the detection task. A low precision signifies a significant amount of FP. Precision can be defined as (1):

A dataset for computer-vision-based fig fruit detection in the wild with benchmarking ... (Adi Izhar Che Ani)

$$Precision = \frac{TP}{TP+FP} \times 100 \quad (1)$$

4.3.2. Recall

Recall indicates that the proportion of affirmative cases is expected to be positive. The recall metric quantifies the number of objects that the algorithm erroneously disregarded. A high recall value indicates a low amount of false negatives. Recall can be depicted as (2):

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (2)$$

4.3.3. F1-score

The F1-score is a metric that quantifies the accuracy of a model by considering both precision and recall, which have an inverse relationship. The model is deemed flawless when the F1-score equals one and may be computed using (3):

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \times 100\% \quad (3)$$

4.4. Detection result

The system was trained using a batch size of 32 and an epoch setting of 100. Figure 8 demonstrates that both versions of YOLO exhibit exceptional performance and are capable of detecting the majority of figs inside an image. The top section of the Figure 8 exhibits the output generated by YOLO v3, whereas the bottom section showcases the outcomes produced by YOLOv4. Figures 8(a)-(d) display four distinct sample photos, enabling a direct comparison of the detection capabilities between the two versions of the YOLO object identification system. According to Table 1, the detection results of YOLOv3 show a greater number of false negative values compared to YOLOv4. False negatives refer to instances that should have been detected but were not. This phenomenon is known as the potential for overlooking some detections when the fruit is extensively obscured by surrounding features, such as branches and leaves, as depicted in Figures 8(a) and (b). YOLOv4 exhibits a significant number of FP samples in its detection results, as the resulting value surpasses that of YOLOv3. The model's tendency to misidentify the backdrop, such as foliage, as fruit is significant. Moreover, if the background and the item share a similar color, it can potentially elevate the likelihood of a FP outcome. The model accurately identifies the presence of 'buah tin' in the image, as shown in Figures 8(c) and (d), even when there is no fig present. Nevertheless, YOLOv4 has a significant capability to accurately detect a greater number of genuine positive samples, even when faced with substantially occluded figures, surpassing the performance of YOLOv3. The greater TPR achieved by YOLOv4 compared to YOLOv3 is evident.

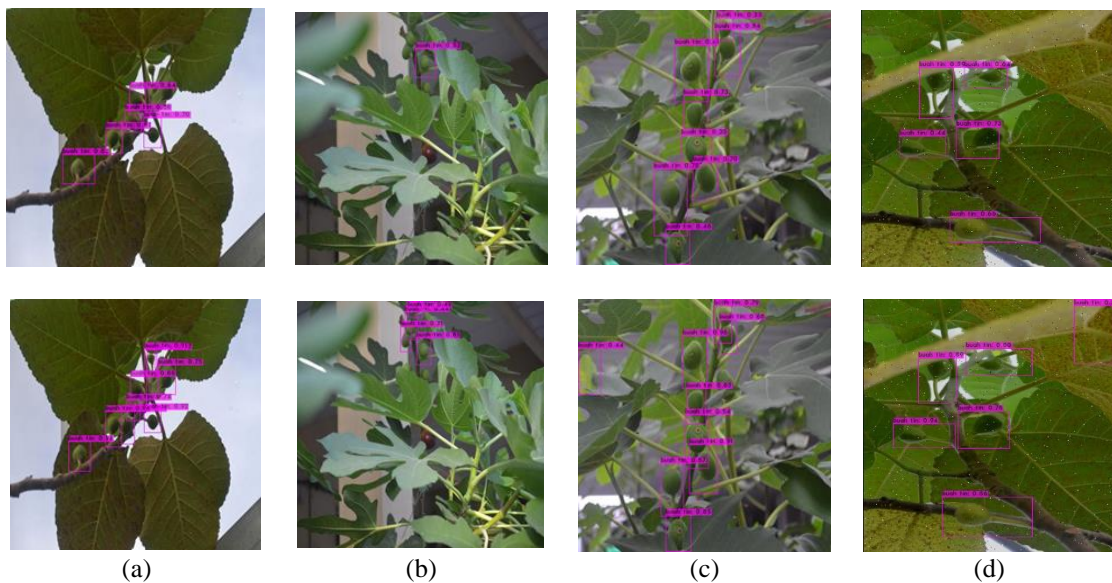


Figure 8. Comparison of the detection result from YOLO v3 (above) and YOLO v4 (below): (a) sample image 1, (b) sample image 2, (c) sample image 3, and (d) sample image 4

The test set comprises 30% of photos randomly selected from the dataset, totaling 138 images. The precision values obtained for both models can be deemed almost similar, with YOLOv3 being marginally higher by one percent compared to YOLOv4. However, according to the data, YOLOv4 performs better than YOLOv3 in terms of recall and F1-score, achieving 89% and 0.84, respectively. Furthermore, we evaluate performance using mAP, which refers to the accurate detection of test images. The mAP attained by YOLOv4 is the greatest, reaching 90.02%, followed by YOLO v3 with a mAP of 81.40%. Hence, the obtained performance results demonstrate that both models are capable of accurately detecting the location of the fig by utilizing our dataset.

Table 1. Result indicators of two models

Model	TP	FP	FN	IoU (%)	Precision (%)	Recall (%)	F1-score	mAP (%)
YOLOv3	330	78	95	58.56	81	78	0.79	81.40
YOLOv4	337	94	48	60.16	80	89	0.84	90.02

5. CONCLUSION

This paper offered a new dataset for fig fruit detection in the wild scenario. This compilation of annotated object instances is expected to facilitate the progress of object detection in complex and crowded surroundings. Regarding the benchmarks, we only conducted tests using the most cutting-edge object detectors to showcase the dataset's capacity and value for object detection. The results suggest that by employing our dataset, which portrays intricate background scenarios, a reasonable level of accuracy is achieved. To be more precise, YOLO v4 achieves an outstanding precision rate of 80%, a recall rate of 89%, an F1 score of 0.84, and a mAP of 90.02%. We anticipate that this dataset will address the current lack of available datasets pertaining to fig fruits, hence assisting computer vision researchers engaged in fruit detection tasks. In addition, we aim to expand our fig dataset by include information about the ripeness of the figs, which will be used for classification purposes. We also plan to enhance the efficiency of the dataset as we move forward.

ACKNOWLEDGEMENTS

The authors express their gratitude for the opportunity to conduct this research, which entails receiving technical assistance from Universiti Teknologi MARA Cawangan Pulau Pinang. The research is funded by the Universitas Negeri Malang Indonesia International Inbound Research Mobility (IIRM) program for the years 2022-2023, under the reference number 30.3.4/UN32.32/KM/2023.




REFERENCES

- [1] M. S. Hadi, Y. P. Sihombing, S. N. Mustika, M. A. Mizar, D. Lestari and C. A. A. Izhar, "Aquaponic Plant Control and Monitoring System Using Iot-Based Decision Tree Logic," *2022 6th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, Yogyakarta, Indonesia, 2022, pp. 705-710, doi: 10.1109/ICITISEE57756.2022.10057752.
- [2] A. Kamaruzaman, M. Farid, C. A. A. Izhar, M. Maruzuki, A. S., and M. A. Habibi, "Application of Deep Learning for Fig Fruit Detection in The Wild," *2022 2nd International Conference on Emerging Smart Technologies and Applications (eSmarTA)*, Ibb, Yemen, 2022, pp. 1-6, doi: 10.1109/eSmarTA56775.2022.9935356.
- [3] A. Amkor and N. El Barbri, "Classification of potatoes according to their cultivated field by SVM and KNN approaches using an electronic nose," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 3, pp. 1471-1477, 2023, doi: 10.11591/eei.v12i3.5116.
- [4] W. Yijing, Y. Yi, W. Xue-Fen, C. Jian, and L. Xinyun, "Fig fruit recognition method based on YOLO v4 deep learning," in *ECTI-CON 2021 - 2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology: Smart Electrical System and Technology, Proceedings*, 2021, pp. 303-306, doi: 10.1109/ECTI-CON51831.2021.9454904.
- [5] N. Häni, P. Roy and V. Isler, "MinneApple: A Benchmark Dataset for Apple Detection and Segmentation," in *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 852-858, April 2020, doi: 10.1109/LRA.2020.2965061.
- [6] N. E. Md Rosli, S. Setumin, A. Nugroho, A. I. Ch. Ani, M. I. F. Maruzuki, and M. S. Osman, "Fig Fruit Image Segmentation using Threshold, K-means Clustering, and Sharp U-Net Techniques," *2022 2nd International Conference on Emerging Smart Technologies and Applications (eSmarTA)*, Ibb, Yemen, 2022, pp. 1-6, doi: 10.1109/eSmarTA56775.2022.9935411.
- [7] L. Jiao *et al.*, "A survey of deep learning-based object detection," *IEEE Access*, vol. 7, pp. 128837-128868, 2019, doi: 10.1109/ACCESS.2019.2939201.
- [8] A. S. F. Kamaruzaman *et al.*, "Systematic literature review: application of deep learning processing technique for fig fruit detection and counting," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 2, pp. 1078-1091, 2023, doi: 10.11591/eei.v12i2.4455.
- [9] N. Jamil, A. N. Noralil, M. I. Ramli, A. K. M. K. Shah, and I. Mamat, "SiulMalaya: an annotated bird audio dataset of Malaysia lowland forest birds for passive acoustic monitoring," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 4, pp. 2269-2281, 2023, doi: 10.11591/eei.v12i4.5243.




- [10] M. D. Yang, H. H. Tseng, Y. C. Hsu, C. Y. Yang, M. H. Lai, and D. H. Wu, "A UAV open dataset of rice paddies for deep learning practice," *Remote Sens (Basel)*, vol. 13, no. 7, p. 1358, Apr. 2021, doi: 10.3390/rs13071358.
- [11] Y. Wang, C. Wang, H. Zhang, Y. Dong, and S. Wei, "A SAR dataset of ship detection for deep learning under complex backgrounds," *Remote Sens (Basel)*, vol. 11, no. 7, p. 765, Apr. 2019, doi: 10.3390/rs11070765.
- [12] R. M. Jasim and T. S. Atia, "Towards classification of images by using block-based CNN," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 1, pp. 373-379, 2023, doi: 10.11591/eei.v12i1.4806.
- [13] J. Deng, W. Dong, R. Socher, L. -J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, USA, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.
- [14] T.-Y. Lin *et al.*, "LNCS 8693-Microsoft COCO: Common Objects in Context," *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, vol. 8693, pp. 740-755, 2014, doi: 10.1007/978-3-319-10602-1_48.
- [15] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010, doi: 10.1007/s11263-009-0275-4.
- [16] S. Gayathri, T. U. Ujwala, C. V. Vinusha, N. R. Pauline, and D. B. Tharunika, "Detection of Papaya Ripeness Using Deep Learning Approach," *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India, 2021, pp. 1755-1758, doi: 10.1109/ICIRCA51532.2021.9544902.
- [17] I. Kilic and G. Aydin, "Traffic Sign Detection and Recognition Using TensorFlow' s Object Detection API With a New Benchmark Dataset," *2020 International Conference on Electrical Engineering (ICEE)*, Istanbul, Turkey, 2020, pp. 1-5, doi: 10.1109/ICEE49691.2020.9249914.
- [18] R. Yamparala, R. Challa, V. Kantharao, and P. S. R. Krishna, "Computerized Classification of Fruits using Convolution Neural Network," *2020 7th International Conference on Smart Structures and Systems (ICSSS)*, Chennai, India, 2020, pp. 1-4, doi: 10.1109/ICSSS49621.2020.9202305.
- [19] H. Basri, I. Syarif, and S. Sukaridhoto, "Faster R-CNN Implementation Method for Multi-Fruit Detection Using Tensorflow Platform," *2018 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC)*, Bali, Indonesia, 2018, pp. 337-340, doi: 10.1109/KCIC.2018.8628566.
- [20] L. Jian, Z. Mingrui, and G. Xifeng, "A fruit detection algorithm based on R-FCN in natural scene," *2020 Chinese Control and Decision Conference (CCDC)*, Hefei, China, 2020, pp. 487-492, doi: 10.1109/CCDC49329.2020.9163826.
- [21] J. Zhao and J. Qu, "A Detection Method for Tomato Fruit Common Physiological Diseases Based on YOLOv2," *2019 10th International Conference on Information Technology in Medicine and Education (ITME)*, Qingdao, China, 2019, pp. 559-563, doi: 10.1109/ITME.2019.00132.
- [22] K. S. K. Patro, V. K. Yadav, V. S. Bharti, A. Sharma, and A. Sharma, "Fish Detection in Underwater Environments Using Deep Learning," *National Academy Science Letters*, vol. 46, pp. 407–412, 2023, doi: 10.1007/s40009-023-01265-4.
- [23] A. Y. Barrera-Animas and J. M. D. Delgado, "Generating real-world-like labelled synthetic datasets for construction site applications," *Automation in Construction*, vol. 151, 2023, doi: 10.1016/j.autcon.2023.104850.
- [24] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv preprint arXiv:2004.10934*, Apr. 2020.
- [25] N. Mamdough and A. Khattab, "YOLO-Based Deep Learning Framework for Olive Fruit Fly Detection and Counting," *IEEE Access*, vol. 9, pp. 84252–84262, 2021, doi: 10.1109/ACCESS.2021.3088075.
- [26] F. Miao, Y. Tian and L. Jin, "Vehicle Direction Detection Based on YOLOv3," *2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, Hangzhou, China, 2019, pp. 268-271, doi: 10.1109/IHMSC.2019.10157.

BIOGRAPHIES OF AUTHORS






Adi Izhar Che Ani    is a senior lecturer at the Centre for Electrical Engineering, Universiti Teknologi MARA, Cawangan Pulau Pinang (UiTM CPP), with a Master's degree in Engineering from Universiti Malaya Malaysia (2012). He obtained his Bachelor's degree in Electrical and Electronics Engineering from the University of Miyazaki (Japan) in 2007. His research interests are the fields of artificial intelligence. He can be contacted at email: adiizhar@uitm.edu.my.






Mohammad Afiq Hamdani Mohammad Farid    was born in Kedah, Malaysia in 1999. He is a final-year student with a Bachelor's degree in Electrical and Electronic Engineering from Universiti Teknologi MARA Cawangan Pulau Pinang (UiTM CPP). His research interests include deep learning and computer vision. He can be contacted at email: 2018297868@student.uitm.edu.my.






Ahmad Shukri Firdhaus Kamaruzaman    is a Master student at the Faculty of Electrical Engineering, Universiti Teknologi MARA Cawangan Pulau Pinang (UiTM CPP). He obtained his Bachelor's degree in Electrical and Electronics Engineering from Universiti Teknologi MARA Cawangan Pulau Pinang (UiTM CPP) in 2018. His research interests are in the fields of artificial intelligence and deep learning. He can be contacted at email: 2021459494@student.uitm.edu.my.



Sharaf Ahmad    received B.Sc. in Applied Physics from Universiti Kebangsaan Malaysia in 1993 and M.Sc. in Solid State Physics in 2007 from Universiti Sains Malaysia. He is now a senior lecturer at the Faculty of Applied Science, Universiti Teknologi MARA Penang branch. His research interest is solid-state applications, plant science, soil physics, and physics education. He can be contacted at email: sharaf@uitm.edu.my.



Mokh Sholihul Hadi    received his B.S. degree in Electrical Engineering, from Brawijaya University Indonesia in 2004. He received his M.S and Ph.D. degrees in Electronics and Applied Physics from Tokyo Institute of Technology, Japan, in 2010 and 2014. Since 2009, he has been worked as Associate Professor at the Department of Electrical Engineering, Faculty of Engineering, State University of Malang Indonesia. His current research interest is embedded IoT system, smart devices, robotics, semiconductor devices, and nano electronics. He can be contacted at email: mokh.sholihul.ft@um.ac.id.