Jurnal Elektronika dan Telekomunikasi (JET), Vol. 23, No. 1, August 2023, pp. 37-46
Accredited by KEMDIKBUDRISTEK, Decree No: **158/E/KPT/2021**
doi: 10.55981/jet.525

Determinants of Pepper Quality Based on the Percentage of Foreign Objects Based You Only Look Once (YOLO)

Indra Dwisaputra*, Siti Barokah, Muhammad Erfani Ramadhani, Ocsirendi

Electrical and Informatic Engineering
Politeknik Manufaktur Negeri Bangka Belitung
Kawasan Air Kantung
Sungailiat, Indonesia

Abstract

The presence of foreign objects in pepper seeds is one of the things that affect the quality of pepper seeds. Farmers in Bangka sell pepper to pepper collectors. The collectors in this area still inspect the pepper using manual methods without the help of other tools, so there are still foreign objects such as dry leaves or pepper stalks. This method is often inefficient because the precision of each person is different. In this case, we propose to determine the quality of pepper based on the percentage of foreign objects automatically in accordance with the determination of pepper quality standards regulated in the national quality standard (SNI). The authors use YOLOv3 for object detection which is one of the fastest and most accurate object detection methods, outperforming other detection algorithms. However, YOLOv3 requires a heavy computer architecture. Therefore, YOLOv3-tiny, a lighter version of YOLOv3, can be a solution for smaller architectures. This study found that YOLOv3-tiny model has a reasonably high network performance value: precision value of 0.99, recall value above 70%, and F1 score above 80%. While determining the quality of pepper according to the standard quality of pepper (SNI) the value obtained must be below 2%. Then a comparison was made between the detection system and the manual calculation of objects. It was found that in the sample of 26 pepper seeds, the system detected 8.97 seconds faster than manual calculation.

Keywords : Pepper, YOLOv3, TinyYOLOv3, image detection

I. INTRODUCTION

Pepper (Piper nigrum L) is one type of spice that is quite important in terms of its role as a contributor to foreign exchange and its unique uses that cannot be replaced by other types of spices. Indonesia is one of the world's largest producers of pepper. The commodity is mainly exported in the form of black pepper and white pepper, and in small quantities in the form of pepper powder and pepper oil. In the world market, black pepper is known as "Lampung black pepper" and white pepper is known as "Muntok white pepper" [1]. Muntok white pepper is located in one region of Indonesia, namely Bangka Belitung province, West Bangka district, Mentok city. Collectors in this area still inspect pepper and foreign objects, such as leaves and stalks, using traditional methods or manually. This method is often inefficient because the precision of each person is different, while the presence of foreign objects in pepper is one of the things that affects the quality of pepper. The determination of pepper quality standards has been regulated in national quality standards (SNI). Pepper seed commodity standards in SNI-01-0004-1995 and SNI-01-0005-1995 are two categories of standards that the National Standardization Agency has made. The standard is based not only on foreign object content but also on moisture content, blackish pepper content, fungal contamination content, light seed content, animal contamination content, and color. Among the above pepper quality standards, we discuss foreign object content in pepper. For pepper quality standard I, the level of foreign objects found in pepper is 1%, while for pepper quality II, the level is 2%. However, it is rather difficult to manually identify foreign objects in pepper from the many pepper seeds processed. Usually, collectors take samples from the pepper and examine them, but there are no other tools to examine the pepper seeds. Furthermore, it is more difficult to determine the quality of the processed pepper when the amount of pepper increases. Deep learning is a solution to detect pepper seeds and foreign objects in pepper so that it can be an option for the development of automatic pepper inspection while determining the quality of the pepper.

The field of object recognition and detection is advancing from time to time according to the development of science and technology. The development of object recognition and detection can be divided into two phases [2]. In the traditional phase, humans play an important role in advising the system on what needs to be done in the detection process. The process of recognizing and detecting objects is still primarily done manually. Then, the deep learning

* Corresponding Author.
Email: dwisaputra.indra@gmail.com
Received: January 2, 2023 ; Revised: April 26, 2023
Accepted: June 21, 2023 ; Published: August 31, 2023

phase allows the system's algorithms to learn and evolve autonomously by using the presented information and previously acquired knowledge without requiring human involvement [3], [4]. Recently, deep learning has been a trending topic that is more frequently used to develop object, face, and other types of detection. The preferred detectors for object and face detection include You Only Look Once (YOLO), Fast-RCNN, and Faster-RCNN. These detectors have advanced precision yet lightweight detection in various fields. Each of these methods has its advantages and disadvantages. However, YOLO is one of the fastest and most accurate object detection methods, outperforming other detection algorithms [5].

YOLO is a branch of Convolutional Neural Network (CNN). YOLO uses convolutional neural network as an object detector [5]. YOLO is one of the methods developed to detect objects in realtime with fairly good accuracy. This method is often used in detecting vehicles, people, fruits, and other objects. YOLO is considered to have a fast and highly accurate architecture. Although the detection speed is fast, YOLO does not have a prior detection phase, so the error in positioning the object is large. In addition, YOLO also has difficulty detecting small objects that are close together. YOLO also has a heavy computer architecture, so the training process will take a long time. The YOLO-tiny version was created for a lighter, faster, and more efficient architecture than YOLO. The YOLO-tiny network has fewer convolutional layers than YOLO so that the training process is faster and can be used to classify and recognize objects [6].

Research on using the YOLO method has been widely conducted. Research conducted by Oktaviani Ella Karlina and Dina Indarti [7] discusses the recognition of fast food objects on video and realtime webcams using the YOLO method. The results show 100% mAP validation accuracy and an average loss of 4.6% for the implementation of the YOLO algorithm in fast food object recognition. Thus, it can be said that YOLO successfully recognizes things in fast food photos. Mawaddah Harahap et al. [8] discussed YOLOv3. Six objects are classified in the results: a car, a bus, a moving vehicle, a person, and a truck, all of which are affected by the camera angle and the size of the object when playing the video. Thus, the object detection results are often inaccurate. Similar to what is said in the bus description, buses have the same large interior dimensions as automobile vehicles. The large-scale shape of a truck is almost the same as a bus. Fix CCTV has the highest mAP (mean Average Precision), which is 97%, while PTZ CCTV has the highest mAP, which is 99%. Adam Fahmi Fandisyah et al. [9] discussed the detection of ships in the Indonesian sea using YOLOv3. The findings show that the Model 2 method, which uses a trained model and approximately nine anchors through a k-means procedure, has the best classification performance for determining the type of cap on a satellite vehicle. The results show that Model 2 has an average increase in profitability over Model 1 with a null value of 0.1978. In addition, Model 2's mAP score is 95.06% and Model 1's score is 94.85%. Based on the average accuracy results of the test data, each category has a value greater than Model 1, which means that Model 2's mAP score is, respectively, 50.41% and 41.88% higher than Model 1. The creation of a module for mobile robots that uses the YOLO approach to detect human objects was covered by Khairunnas et al. [10]. The findings showed that the human object detection module in this investigation successfully identified human items. According to the YOLOv4 performance test results, a total of 904 photos were processed in 116 seconds with an mAP value of 87.03%. This module can detect single object and multiple objects in the test data to determine the detection angle of human objects. The test is performed to see if an object can be recognized by correctly distinguishing between objects inside and outside a certain angle range. When the object is still in the frame, object recognition by ID successfully classifies objects based on their ID. Calvin Geraldy and Chairisni Lubis [11] discussed the detection and recognition of car types using YOLO and CNN algorithms. The findings show the effectiveness of YOLO and CNN methods for car detection and identification. 88.1% of the tests for car detection and recognition were successful. The most effective test for car recognition used an epoch of 100 and a learning rate of 0.000005. During testing, the system was able to identify several types of cars on a single test image. Detection and recognition results can vary depending on the amount of light present, the number of cars present, and how the exterior of the car has been altered. However, pepper detection research is still very little. Research presented by Abdur Rohim and Reiva Marizka Harmie [12] has made a pepper fruit detection system using the CNN method which can only detect one pepper in the form of a stalk of pepper, not in the form of pepper seeds.

In this study, the authors developed a pepper detection system while determining the quality of pepper based on the percentage of foreign objects using the YOLO method. By making this system, it is expected that the system can detect more than one pepper in the form of granules. In addition to detecting pepper, this system can also detect foreign objects in pepper, such as pepper leaves and stalks, so that it can determine the quality of pepper based on the percentage of foreign objects in pepper.

II. METHODOLOGY

This study uses digital image processing techniques to determine the quality of pepper based on the percentage of foreign objects. The research process of determining the quality of pepper images based on the percentage of foreign objects uses the YOLO algorithm for realtime detection. The initial stage of this research starts with preparing images for datasets in the form of pepper seeds, leaves, and stalks. Furthermore, the images data are collected in one folder named dataset, which will then be preprocessed. In this preprocessing, image data size changes and data labeling are carried out. Then the data that has been processed will be trained using the YOLO algorithm. The results of this training model will be evaluated first to determine the quality of the model and how much accuracy is obtained before being applied to the detection system. If the training model is as desired, the model is ready to be implemented for realtime object detection. The research scheme of the image processing process of determining the quality of pepper based on the percentage of foreign objects can be seen in Figure 1 below.

A. You Only Look Once (YOLO)

You Only Look Once (YOLO) is an object detection system, which is targeted for realtime processing and frames object detection as a single regression problem, where from image pixels to bounding boxes and associated class probabilities [13]. Currently, there are various object detection methods that are extensions of the Convolutional Neural Network (CNN) artificial neural network architecture [14], such as Faster R-CNN [15], Single Shot Detector (SSD) [16], and You Only Look Once (YOLO) [17]. Each method has its own advantages and disadvantages. The YOLO method is one of the fastest and most accurate methods in object detection. It can even exceed up to two times the ability of other algorithms. YOLOv3 is able to detect objects with a higher frame rate than Faster R-CNN [17], and also YOLOv3 has a mean average-precision (mAP) parameter value on the intersection over union (IoU) metric of 0.5, which is better than SSD that has been trained on the COCO dataset [18], which means it has a better level of accuracy in recognizing objects. YOLOv3 uses the Darknet-53 architecture which has 53 convolutional layers. This architecture is superior to YOLOv2 and also has shortcut connections. However, YOLOv3 has a large architecture and feature extractor so the training process will take longer and require high computer specifications. Thus, the YOLOv3-tiny version, which is a lighter version of YOLOv3, is the solution. The convolutional layer in the tinyYOLO architecture is reduced so that the training process can be faster and can be applied to computers that have adequate specifications [6]. The YOLOv3 architecture can be seen in Table 1 and the YOLOv3-tiny feature extractor can be seen in Table 2.

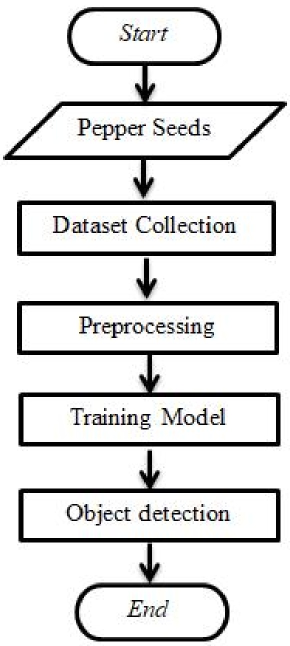


Figure 1. Schematic of Research

B. Dataset

The image dataset in this study is an image of pepper seeds and foreign objects in the form of leaves and stalks of pepper. Image data is taken manually using an Android camera. The number of images as a dataset is 222 images in jpg format. Pepper seeds amount to 100 images, pepper here is in the form of granules. While the leaves amounted to 50 images, the stalk amounted to 50 images, and 22 images are a combination of pepper, leaves, and stalks. Furthermore, image data preprocessing is carried out which consists

TABLE 1
YOLOV3 ARCHITECTURE

Type	Filters	Size	Output	
Convolutional	32	3x3	256x256	
Convolutional	64	3x3/2	128x128	
Convolutional	32			
Convolutional	64	1x1		1x
Residual		3x3	128x128	
Convolutional	128	3x3/2	64x64	
Convolutional	64	1x1		
Convolutional	128	3x3		2x
Residual			64x64	
Convolutional	256	3x3/2	32x32	
Convolutional	128	1x1		
Convolutional	256	3x3		8x
Residual			32x32	
Convolutional	512	3x3/2	16x16	
Convolutional	256	1x1		
Convolutional	512	3x3		8x
Residual			16x16	
Convolutional	1024	3x3/2	8x8	
Convolutional	512	1x1		
Convolutional	1024	3x3		4x
Residual			8x8	
Avgpool		Global		
Connected		1000		
Softmax				

TABLE 2
YOLOV3-TINY FEATURE EXTRACTOR

Layer	Type	Filters	Size/ Stride	Input	Output
0	Convolutional	16	3x3/1	416x416x3	416x416x16
1	Maxpool		2x2/2	416x416x16	208x208x16
2	Convolutional	32	3x3/1	208x208x16	208x208x32
3	Maxpool		2x2/2	208x208x32	104x104x32
4	Convolutional	64	3x3/1	104x104x32	104x104x64
5	Maxpool		2x2/2	104x104x64	52x52x64
6	Convolutional	128	3x3/1	52x52x64	52x52x128
7	Maxpool		2x2/2	52x52x128	26x26x128
8	Convolutional	256	3x3/1	26x26x128	26x26x256
9	Maxpool		2x2/2	26x26x256	13x13x256
10	Convolutional	512	3x3/1	13x13x256	13x13x512
11	Maxpool		2x2/1	13x13x512	13x13x512
12	Convolutional	1024	3x3/1	13x13x512	13x13x1024
13	Convolutional	256	1x1/1	13x13x1024	13x13x256
14	Convolutional	512	3x3/1	13x13x256	13x13x512
15	Convolutional	255	1x1/1	13x13x512	13x13x255
16	YOLO				
17	Route 13				
18	Convolutional	128	1x1/1	13x13x256	13x13x128
19	Upsample		2x2/1	13x13x128	26x26x128
20	Route 19, 8				
21	Convolutional	256	3x3/1	26x26x384	26x26x256
22	Convolutional	255	1x1/1	26x26x256	26x26x255
23	YOLO				

of changing the image size and labeling. Image data from the dataset is resized to 1000 x 750. Next, labeling will be done. Image labeling is the initial stage where each image in the dataset is labeled to convey image information. The labeling process is done by giving each image object the bounding box and class name. Labeling consists of two classes: the first class is labeled with the class name "pepper", and the second class consists of leaves and stalks of pepper is labeled with the class name "Benda_asing" with the YOLO file.txt format. The labeling process is done manually using MakesenseAI software. The training model dataset can be seen in Table 3, and the labeling process results can be seen in Figure 2 and Figure 3.

TABLE 3
DATASET TRAINING MODEL

Object Name	Sample Dataset	Class Name	Number of Datasets
Pepper		Pepper	104
Leaves		Foreign Objects	57
Stalk of Pepper		Foreign Objects	31
Pepper, leaves, and stalks		Pepper, Foreign Object	20

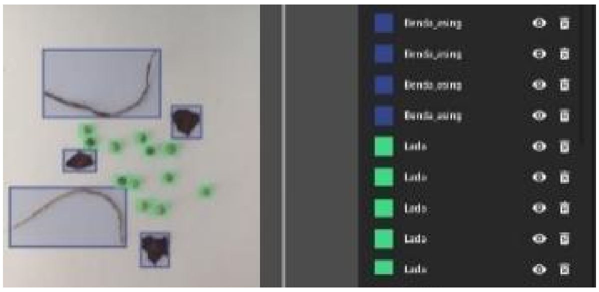


Figure 2. Dataset Labeling Process

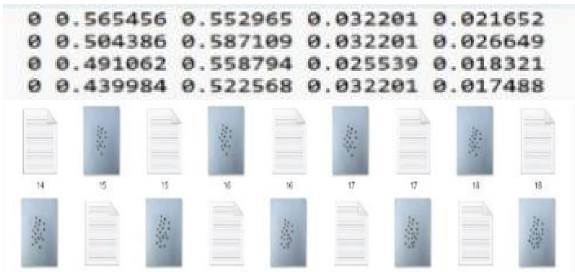


Figure 3. Result of Labeling Process

C. YOLOv3-tiny Training Model

In the training stage, the model is carried out using a cloud notebook, namely the Google Colaboratory platform. This training process utilizes the Darknet open-source library available in the online repository

TABLE 4
CONFIGURATION OF YOLOV3-TINY

Type of Configuration	Value
Class	2
Max_Batch	4000
Filters YOLO	21

Figure 4. Saved Weight Files

(AlexeyAB) which functions as a load model for training YOLOv3-tiny data. The hyperparameter configuration can be seen in Table 4.

Datasets that have undergone the preprocessing stage will continue in the training process. The dataset will be included in the train set in the file "obj.data" and uses the YOLOv3-tiny configuration specified in the yolov3_training.cfg file which functions as a load weight and requires the downloaded YOLOv3-tiny pretrained weight. This process is called a transfer learning technique.

This training process takes about five hours to train the model. After the training process is successful, the training results will be saved in the form of a weights file. The process of saving weights will start at the first 1000 iterations until the final weight file named "yolov3_training_final.weight". This final weight file will be used for detection on static images and realtime detection. Figure 4 shows an example of a saved weight file.

D. YOLOv3-tiny Object Detection

After doing the training process on the Google Colaboratory platform and getting a weight file for detection, the next step is to perform a validation program on the YOLOv3-tiny detection process. The final result of the validation process is the object's bounding box in the image with the label of the class name and its confidence value. The bounding box can be seen in Figure 5 below.

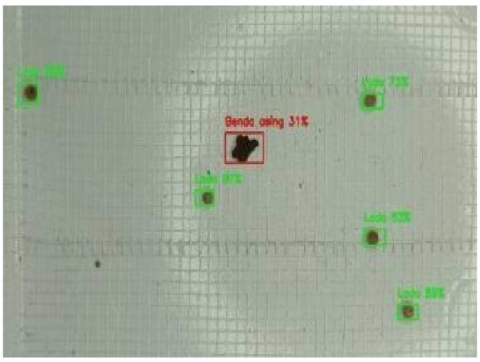
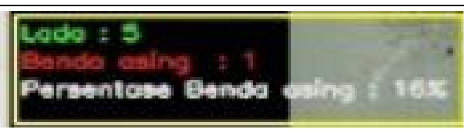
Figure 5. Bounding Box

Before carrying out the validation process, the final weight file of the YOLOv3-tiny model, the .cfg file containing the hyperparameter configuration of the YOLOv3-tiny network, and the preprocessed dataset file must be downloaded. This validation process uses Jupyter Notebook on Anaconda3 using the YOLOv3-tiny algorithm. This process is done to determine the quality of the model and how much accuracy is obtained.

III. RESULTS AND DISCUSSION

After the entire detection process is carried out, the YOLOv3-tiny network performance test is then carried out to determine the performance of the model in realtime detection of new images. The performance of the YOLOv3-tiny network in testing data will be evaluated by calculating the value of confusion matrix, recall, precision, and F1 score in detecting pepper objects and foreign objects. Performance testing will be carried out on 16 images of test data taken in realtime using DroidCam. The detection is carried out on a white background section with a size of 17.6 x 22.8 cm, where the pepper sample will be detected. In this test, we used a predetermined distance of 20 cm from the cross section which corresponds to the size of the cross section area. A ringlight lamp was used as a lighting aid during the test, and the test was conducted during the day and night. We conducted tests using lighting and without lighting. The test results can be seen in Table 5.

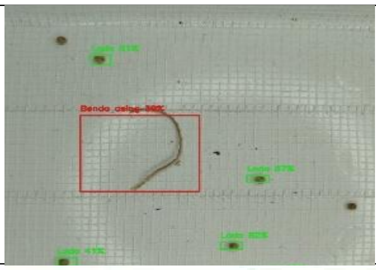
TABLE 5
REALTIME TESTING RESULTS

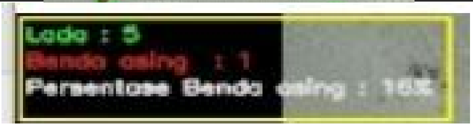
Actual	Results
Picture 3 (daytime) Pepper: 5 foreign object: 1	  In this detection, all objects were successfully detected.

Picture 9 (daytime)

Pepper : 7

Foreign Objects: 1



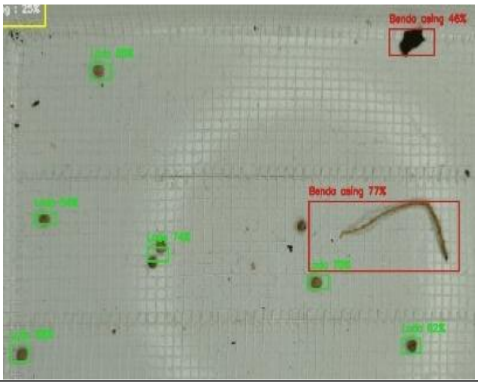


In this detection, there were two pepper seeds that were not successfully detected, due to the lack of quality of the camera, or the lack of training dataset.

Picture 46 (daytime)

Pepper : 8

Foreign Objects : 2

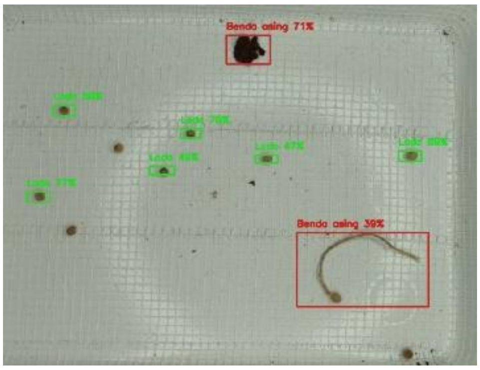


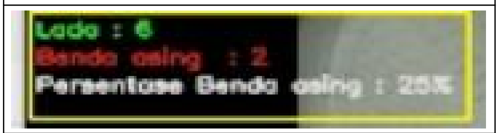
In this detection, there were two pepper seeds that were not successfully detected, this was due to the pepper seeds being close together, the lack of quality of the camera, or the lack of datasets that were trained.

Picture 1 (daytime)

Pepper : 10

Foreign Objects: 2



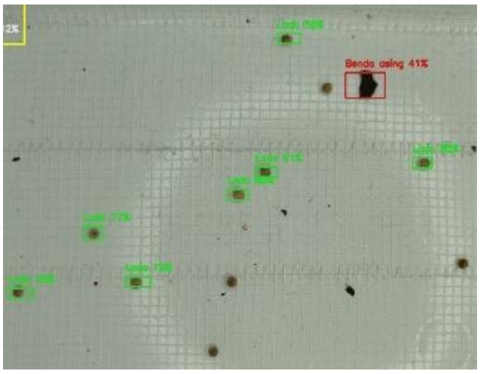


In this detection, there were four pepper seeds and one foreign object that were not successfully detected, this was due to the pepper being close to the stalk, the lack of quality of the camera, or the lack of training dataset.

Picture 37 (daytime)

Pepper : 11

Foreign Objects : 1

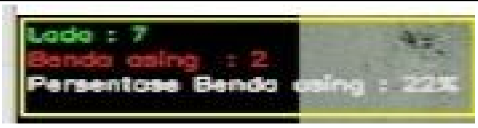


In this detection, there were four pepper seeds that were not successfully detected, this was due to the lack of quality of the camera, or the lack of a training dataset.

Picture 48 (daytime)

Pepper : 13

Foreign Objects : 2

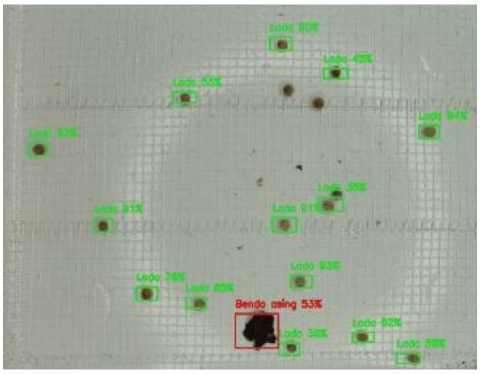


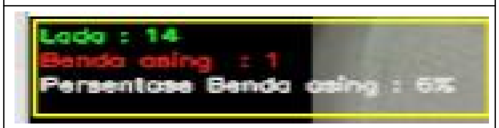
In this detection, there were six pepper seeds that were not successfully detected, this was due to the lack of quality of the camera, or the lack of datasets that were trained.

Picture 4 (daytime)

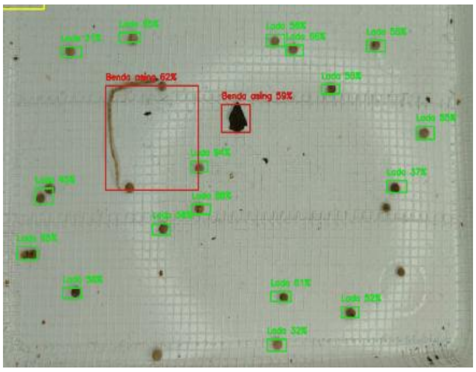
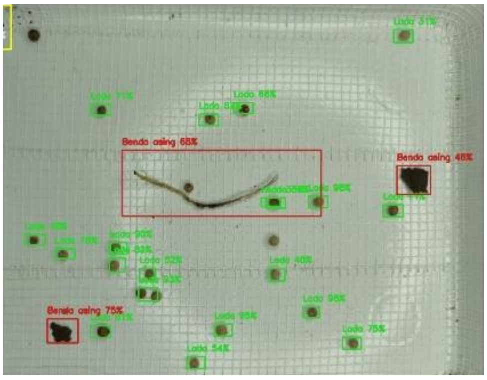
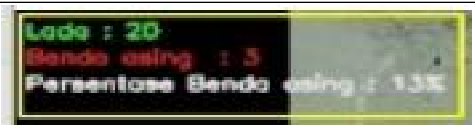
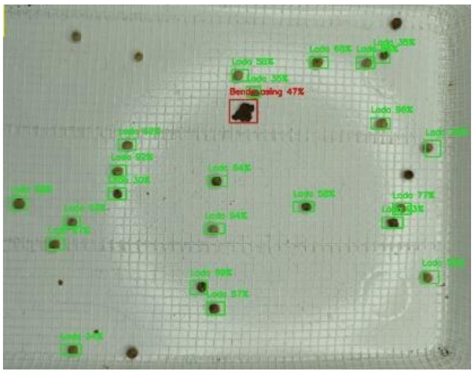
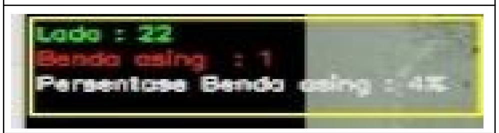
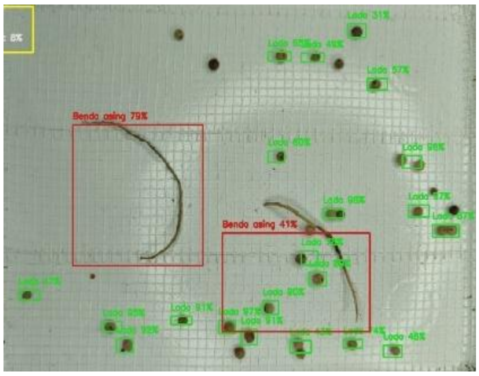
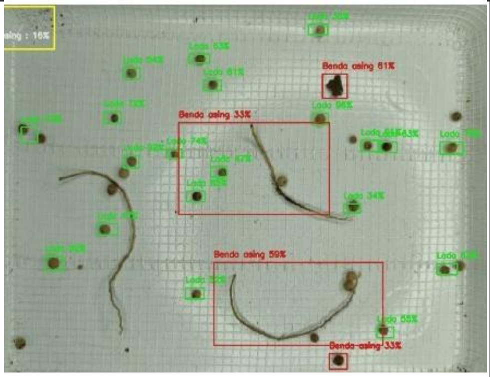
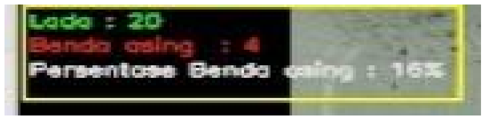
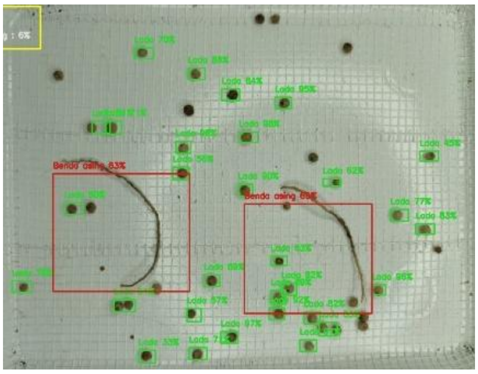
Pepper : 17

Foreign Objects: 1





In this detection, there were three pepper seeds that were not successfully detected, this was due to the pepper seeds being close together, the lack of quality of the camera, or the lack of a training dataset.

Picture 37 (night)	
Pepper: 23 Foreign Objects: 2	
	In this detection, there were six pepper seeds that were not successfully detected, this was due to the pepper seeds being close together, the lack of quality of the camera, or the lack of training dataset.
Picture 18 (night)	
Pepper : 24 Foreign Objects: 3	
	In this detection, there were four pepper seeds that were not successfully detected, this was due to the pepper seeds being close together, the lack of quality of the camera, or the lack of a training dataset.
Picture 30 (night)	
Pepper : 27 Foreign Objects: 1	
	In this detection, there were five pepper seeds that were not successfully detected, this was due to the lack of quality of the camera, or the lack of a training dataset.
Picture 34 (night)	
Pepper : 31 Foreign Objects: 2	
	In this detection, there were ten pepper seeds that were not successfully detected, this was due to the pepper seeds being close together, the lack of quality of the camera, or the lack of a training dataset.
Picture 42 (night)	
Pepper : 32 Foreign Objects: 4	
	In this detection, there were eleven pepper seeds, one pepper seed detected as foreign object, and one stalk that was not detected, this was due to the pepper seeds being close together, the lack of quality of the camera, or the lack of dataset that was trained.
Picture 35 (night)	
Pepper : 44 Foreign Objects: 2	
	In this detection, there were fourteen pepper seeds that were not successfully detected, this was due to the pepper seeds being close together, the lack of quality of the camera, or the lack of a training dataset.

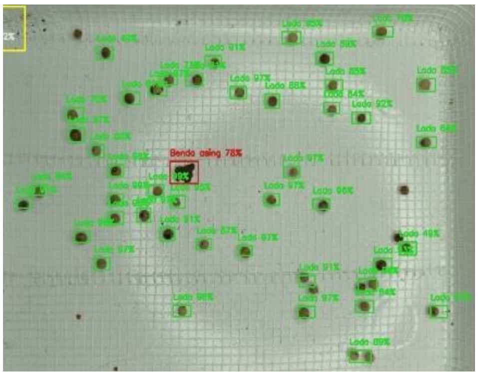
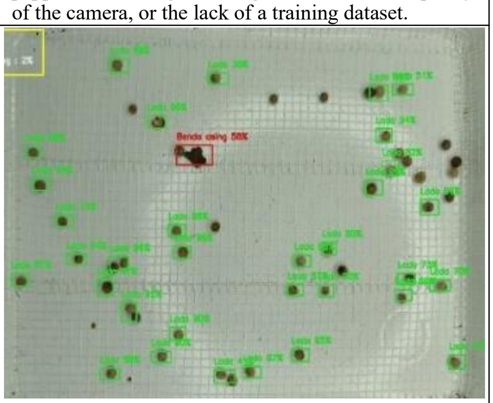
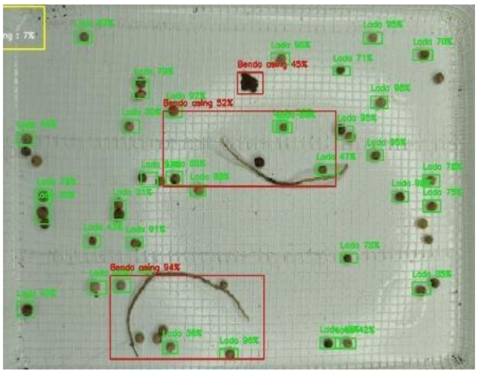
Picture 23 (night)	
Pepper : 49 Foreign Objects: 1	 In this detection, there were five pepper seeds that were not successfully detected, this was due to the pepper seeds being close together, the lack of quality of the camera, or the lack of a training dataset.
Picture 24 (night)	
Pepper : 50 Foreign Objects: 1	 In this detection, there were seventeen pepper seeds that were not successfully detected, this was due to the pepper seeds being close together, the lack of quality of the camera, or the lack of a training dataset.
Picture 27 (night)	
Pepper: 51 Foreign Objects: 3	 In this detection, there were sixteen pepper seeds that were not successfully detected, this was due to the pepper seeds being close together, the lack of quality of the camera, or the lack of a training dataset.
Total	
Pepper : 402 Foreign objects: 29	Pepper : 292 Foreign objects: 27

TABLE 6
CONFUSION MATRIX

Confusion Matrix		**Aktual**	
		Lada	**Benda asing**
Prediction	Lada (pepper)	292	-
	Benda asing (foreign object)	1	27
	Tidak Terdeteksi (not detected)	109	1

In the testing stage, the output results are obtained on the monitor screen: if it is green then the pepper is detected, and if it is red then the foreign objects, such as leaves or stalks of pepper, are detected. The caption marked in green color indicates the number of detected pepper seeds, the caption marked in red color indicates the number of detected foreign objects, and the caption marked in white color indicates the percentage of detected foreign objects which also determines the pepper quality obtained. Based on the test results above, the confusion matrix of the detection results of pepper and foreign objects will be made as in Table 6.

Based on Table 6, the confusion matrix value can be found: true positive, false positive, false negative, precision, recall, and F1 score values. The following equation is used to find these values.

$$True\ Positive = 292 + 27 = 319$$

$$False\ Positive = 1$$

$$False\ Negative = 109 + 1 = 110$$

$$Precision = \frac{TP}{TP+FP} = \frac{319}{319+1} = 0{,}99$$

$$Recall = \frac{TP}{TP+FN} = \frac{319}{319+110} = 0{,}74$$

$$F1\ Score = 2\ x \frac{(Precision\ x\ Recall)}{(Precision+Recall)} =$$

$$2\ x \frac{(0{,}99\ x\ 0{,}74)}{(0{,}99+0{,}74)} = 0{,}84$$

The true positive value obtained a number of 319 objects of pepper and foreign objects that were successfully detected correctly, false positive has 1 object that cannot be recognized, false negative has a number of 110 objects that were not successfully detected, precision of 0.99, recall of 0.74, and F1 score of 0.84. Therefore, the value of realtime network performance testing has a high value.

After testing the network performance where the precision, recall, and F1 score values are obtained, calculations are then made to determine the quality of pepper based on the percentage of foreign objects. To get the quality of pepper based on the percentage of foreign objects, the total foreign objects is divided by the total of detected pepper seeds and detected foreign objects, then multiplied by 100 (total foreign objects/(total pepper seeds + total foreign objects) * 100). According to the pepper quality standards (SNI), the quality of pepper is determined by a value below 2%.

In addition to testing the network performance using 16 images from the realtime test data, we also calculated the length of the system detection time by manual calculation. To do the comparison, we took 10 trial samples. The following are sample data results of time comparison of automatic detection and manual calculation in Table 7.

The comparison results obtained can be seen in Figure 6. When comparing the detection system and the manual detection system (by humans), it was found that the amount of pepper greatly affects the difference in detection comparison time. It is more difficult to manually count when there are more pepper seeds and foreign objects. From the data obtained in Table 7, the system can detect 7 pepper seeds with a time of 0.27 seconds faster than the manual method and for 26 pepper seeds with a time of 8.97 seconds faster than the manual method. While the system can detect maximally if there are no network constraints at the time of detection.

TABLE 7
SAMPLE DATA RESULTS DETECTION TIME SYSTEM AND MANUAL

No.	Pepper Quantity (per seed)	Auto Time (seconds)	Manual Time (seconds)
1	7	2,5	2,77
2	10	2,91	3,42
3	12	1,89	4,77
4	19	2,66	7,31
5	17	3,73	6,82
6	20	2,15	8,8
7	22	2,82	9,98
8	26	2,79	11,76
9	23	3,74	10,31
10	24	1,52	10,73

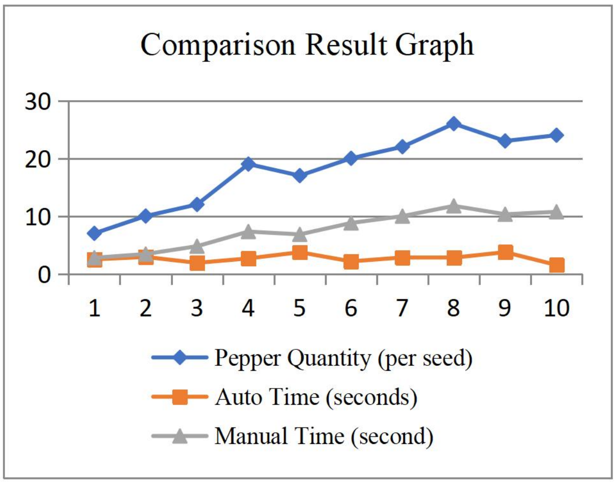


Figure 6. Comparison Result Graph

IV. CONCLUSION

In this study, determining the quality of pepper based on the percentage of foreign objects usingYOLO has good performance in detecting objects. The YOLO algorithm has a precision of 0.99 which means it has an almost perfect value (equal to 1), a recall of 0.74 which means that the overall detection rate of pepper seeds and foreign objects is 0.74, and F1 score is 0.84. Therefore, the value of realtime network performance testing has a high value. In this study, the system has been able to detect more than one object and determine the quality of pepper based on the percentage of foreign objects. However, the system is still difficult to detect if the targeted objects are close together. This happens because the size of the object is small, making it difficult to detect. This can be improved by adding more datasets or improving the image quality of the datasets. Then, the system performance will be even better.

This research also compares the detection time of the system with manual calculation. It was found that using the YOLOv3-tiny detection system is faster when compared to the manual method. Because the more objects that are detected, the longer the detection time will be. This will be difficult if done manually. However, this system cannot be compared with other methods because there has been no similar previous research on determining the quality of pepper using YOLO based percentage. Therefore, increasing the variety of datasets and improving the quality of datasets can be done for future research so that system performance can be improved for the better.

DECLARATIONS

Conflict of Interest

The authors have declared that no competing interests exist.

CRediT Authorship Contribution

Indra Dwisaputra: Conceptualization, Methodology, Supervision, Writing - Reviewing & Editing; Siti Barokah: Software Data Curation, Writing-Original Draft; Muhammad

Erfani Ramdhani: Validation, Resources, Ocsirendi: Formal Analysis, Writing - Reviewing & Editing.

Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

Acknowledgement

The authors would like to thank Politeknik Manufaktur Negeri Bangka Belitung that support the completion of this research.

REFERENCES

[1] Nurdjannah, "Perbaikan mutu lada dalam rangka meningkatkan daya saing di pasar dunia," *J. Perspekt.*, vol. 5, no. 1, pp. 13–25, 2006.

[2] N. O'Mahony *et al.*, "Deep learning vs. traditional computer vision," *Adv. Intell. Syst. Comput.*, vol. 943, no. Cv, pp. 128–144, 2020, doi: 10.1007/978-3-030-17795-9_10.

[3] Dina Gasong, *Belajar dan Pembelajaran*. Deepublish, 2018, 2018. [Online]. Available: https://www.bing.com/search?q=belajar+dan+pembelajaran+dina+gasong&qs=n&form=QBRE&sp=-1&lq=0&pq=belajar+dan+pembelajaran+dina+gasong&sc=0-36&sk=&cvid=6915769CAF6A41EE988E6EA9C6E24116&ghsh=0&ghacc=0&ghpl=#

[4] M. S. Dr. Yenny Suzana , M.Pd., Imam Jayanto, S.Farm., *Teori Belajar & Pembelajaran*. April 2021, 2021. [Online]. Available: https://www.bing.com/search?q=teori+belajar+dan+pembelajaran+yenny+suzana&qs=UT&pq=teori+belajar+dan+pembelajaran+yenny&sc=1-36&cvid=3EE15148639D4F01913D81B3197AEA72&FORM=QBRE&sp=1&ghc=1&lq=0#

[5] W. Fang, L. Wang, and P. Ren, "Tinier-yolo: a real-time object detection method for constrained environments," *IEEE Access*, vol. 8, pp. 1935–1944, 2020, doi: 10.1109/ACCESS.2019.2961959.

[6] J. Redmon and A. Farhadi, "YOLOv3: an incremental improvement," 2018, [Online]. Available: http://arxiv.org/abs/1804.02767

[7] O. E. Karlina and D. Indarti, "Pengenalan objek makanan cepat saji pada video dan real time webcam menggunakan metode you look only once (yolo)," *J. Ilm. Inform. Komput.*, vol. 24, no. 3, pp. 199–208, 2019, doi: 10.35760/ik.2019.v24i3.2362.

[8] M. Harahap *et al.*, "Sistem cerdas pemantauan arus lalu lintas dengan yolo (you only look once v3)," *Semin. Nas. APTIKOM*, p. 2019, 2019.

[9] A. F. Fandisyah, N. Iriawan, and W. S. Winahju, "Deteksi kapal di laut indonesia menggunakan yolov3," *J. Sains dan Seni ITS*, vol. 10, no. 1, 2021, doi: 10.12962/j23373520.v10i1.59312.

[10] K. Khairunnas, E. M. Yuniarno, and A. Zaini, "Pembuatan modul deteksi objek manusia menggunakan metode yolo untuk mobile robot," *J. Tek. ITS*, vol. 10, no. 1, 2021, doi: 10.12962/j23373539.v10i1.61622.

[11] C. Geraldy and C. Lubis, "Pendeteksian dan pengenalan jenis mobil menggunakan algoritma you only look once dan convolutional neural network," *J. Ilmu Komput. dan Sist. Inf.*, vol. 8, no. 2, p. 197, 2020, doi: 10.24912/jiksi.v8i2.11495.

[12] A. Rohim, R. M. Harmie, and M. I. Nugraha, "Prosiding seminar nasional sistem pendeteksi buah lada berbasis," 2021.

[13] R. Huang, J. Pedoeem, and C. Chen, "YOLO-lite: a real-time object detection algorithm optimized for non-gpu computers," *Proc. - 2018 IEEE Int. Conf. Big Data, Big Data 2018*, pp. 2503–2510, 2019, doi: 10.1109/BigData.2018.8621865.

[14] S. Mane and P. S. Mangale, "Proceedings of the 2nd international conference on intelligent computing and control systems, iciccs 2018," *Proc. 2nd Int. Conf. Intell. Comput. Control Syst. ICICCS 2018*, no. Iciccs, pp. 1809–1813, 2019.

[15] B. Liu, W. Zhao, and Q. Sun, "Study of object detection based on faster r-cnn," *Proc. - 2017 Chinese Autom. Congr. CAC 2017*, vol. 2017-Janua, pp. 6233–6236, 2017, doi: 10.1109/CAC.2017.8243900.

[16] W. Liu et al., "SSD: Single Shot MultiBox Detector," Computer Vision – ECCV 2016. Springer International Publishing, pp. 21–37, 2016. doi: 10.1007/978-3-319-46448-0_2.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.

[18] T. Y. Lin *et al.*, "Microsoft coco: common objects in context," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8693 LNCS, no. PART 5, pp. 740–755, 2014, doi: 10.1007/978-3-319-10602-1_48.