

INTELLIGENT SYSTEM FOR AUTOMATIC CLASSIFICATION OF FRUIT DEFECT USING FASTER REGION-BASED CONVOLUTIONAL NEURAL NETWORK (FASTER R-CNN)

^aHasan Basri, ^bIwan Syarif, ^{c,1}Sritrusta Sukaridhoto, ^dMuhammad Fajrul Falah

^aDepartment of Informatics Management, Politeknik Negeri Fakfak, West Papua, Indonesia

^{b,c,d}Department of Informatics and Computer Engineering, Politeknik Elektronika Negeri Surabaya,
Indonesia

E-mail: hasanbasri@polinef.id, iwanarif@pens.ac.id, dphoto@pens.ac.id

Abstract

In 2018, the Indonesian fruit exports increased by 24% from the previous year. The surge in demand for tropical fruits from non-tropical countries is one of the contributing factors for this trend. Some of these countries have strict quality requirements – the poor level of quality control of fruit is an obstacle in achieving greater export yield. This is because some exporters still use manual sorting processes performed by workers, hence the quality standard varies depending on the individual perception of the workers. Therefore, we need an intelligent system that is capable of automatic sorting according to the standard set. In this research, we propose a system that can classify fruit defects automatically. Faster R-CNN (FRCNN) architecture proposed as a solution to detect the level of defect on the surface of the fruit. There are three types of fruit that we research, its mangoes (sweet fragrant), lime, and pitaya fruit. Each fruit divided into three categories (i) Super, (ii) middle, (iii) and fruit defects. We exploit join detection and video tracking to calculate and determine the quality of fruit in real-time. The datasets are taken in the field, then trained using the FRCNN Framework using the tensor flow platform. We demonstrated that this system can classify fruit with an accuracy level of 88% (mango), 83% (lime), and 99% (pitaya), with an average computation cost of 0.0131 m/s. We can track and calculate fruit sequentially without using additional sensors and check the defect rate on fruit using the video streaming camera more accurately and with greater ease.

Keywords: Deep Learning, Faster R-CNN, Object Detection, Tensor Flow, Tracking Detection.

¹ Corresponding author.

INTRODUCTION

The fruit is widely cultivated in Indonesia. National fruit production has significantly increased in the past years. In 2016, national fruit production amounted to 17.711,548 tons, which subsequently increased by 7.4% to 19.021,099 tons in 2017. This growth causes Indonesia to become one of the top tropical fruit exporters in several export destinations. In 2017, Indonesia exported 33.68 thousand tons of fruit worth of 19.95 million dollars in the US[1]. Mangoes, pitayas, and limes are some of the fruits exported from Indonesia to several countries. Besides, we want to focus on researching these fruits because mangoes, pitayas, and limes have become a commodity in recent years due to the increased production of said fruits in Indonesia and high demand from the domestic and foreign markets.

The type of fruits planted by farmers is very dependent on the harvest season and some are produced throughout the year. Sundry varieties of fruits produced in Indonesia are mango, pineapple, malacca, pitaya and limes which are scattered throughout different provinces. Additionally, each region has a different commodity value for different fruits. Harvesting mangoes is carried out if it has a maturity level of a minimum of 80% in a periodic range of 110 to 120 days after sprouting mango blossoms with green color and red stalk [2]. Characteristics of ripe mangoes can be observed from the surface of the skin, where green spots and yellowish speckles are present. Furthermore, a pitaya fruit is ripe when the fruit shows red spots that were initially green. Farmers can harvest it when the fruit appears 32 days after flowering [3]. Ripe limes have the characteristics of green, shiny skin. They must be free of defects or have very slight defects at most, provided these do not affect the general appearance, quality, keeping quality and presentation of the produce. Limes of diameter below 42 mm are excluded if the fruit has a fundamental difference between small and large fruits shown in diameter sizes not exceeding 7 mm[4]. The limes that have fulfilled the above characters are ready for harvest. The quality of fruit after harvest must be maintained throughout the process of sorting, packaging, storing, and fruit distributed systems to the supermarket.

The majority of Indonesian farmers are currently sorted fruit using manual methods. However manual sorting involves workers having

to perform sensory tasks in large capacity and for longer working hours, which renders such method as unreliable. Besides, this process also requires a long time and a greater number of employees. A large number of employees will increase the cost of production and can lead to uncertainty and inaccuracy because individual judgments are subjective and inconsistent with fruit objects and work done repeatedly can cause saturation[5]. To overcome these limitations, it requires the help of computer image sensors to be more effective and efficient. Therefore, we propose a method for detecting multi-fruit using the AI-based smart cameras approach that can classify several varieties of fruit at one time based on color characteristics to make it easier for farmers in the process of fruit sorting. Artificial intelligent technologies have been using research to classify mangos teen defects on the surface fruit [6]. However, the method used in the traditional deep learning approach required 15 seconds to detect one fruit. It would be inefficient if this method was to be used in processing very large amounts of data. The same problem is acknowledged by K.N. Rajit et al. In identifying and classification fruit diseases [7].

More recently, Deep Learning is implemented by [8] to accurately detect fruit. They found that the application of a faster R-CNN framework (FRCNN) in this research could detect paprika fruits and achieved impressive accurate results. Susovan Jana et al. conducted fruit classification using the method of color segmentation and edge detection [9]-[12]. The proposed approach is carried out using the image acquisition process. However, they did not consider that the approach taken was not only detection-estimating the quality of fruit was also important in this case. Quality and quantity of fruit are the two main factors of farmers' success in distributing fruit to several countries.

Furthermore, they developed a system that can calculate the amount of harvest obtained by farmers. Their approach uses the radial symmetry transformation method to detect grapes. The detection results are then processed to produce estimates [13], [14]. However, this method only detects grapes that appear in front of them, whereas the blocked fruit cannot be detected. L Agilan Deswari also researched determine the quality of mangoes before reaching the mango market [15]. The author's approach was not optimal because algorithm optimization was not used in the

selection of features to improve the accuracy obtained.

Other researchers have designed an application for image processing and color transformation. The method used in this study was color transformation to detect and classify the maturity level of bananas. Images of bananas in extraction are based on the values of red, green, and blue and are then converted to this [16]. From the experiment, the authors show that the approach taken can obtain 85% accuracy. Mc.Cool proposed of pepper detection and segmentation [17]. they used the LBP (Local Binary Pattern) as a method for extracting feature series [18], histogram [19] HSV color gradient and auto-encoder features. Then to segmentation, it is necessary to process random data to create vector features. This method has a good performance similar to human image processing, but its detection performance is surpassed by Deep Fruits.

Recently, Deep Fruit was proposed by Sa et al. This approach uses the FRCNN object detection framework [20] which uses deep learning to enter the proposal region and classify the proposed region. They achieved impressive results by applying this technique to several plants and showing that such an approach can be trained quickly and used. Another deep learning approach was suggested by Chen et al [21] to perform segmentation and detection using DCNN. From here they estimate results only from the image. Two aspects that are not considered from one of his works are the potential for estimating maturity and how to calculate of fruit the conveyor runway using video Tracking (or real-time data). This case is our inspiration to develop a system that can be to evaluate the quality and quantity of fruit in the field.

Three necessary steps should be done to develop an automatic classification system of fruit defect. The first stage is to divide each fruit into three categories; a) super, b) middle, c) defect. Secondly, using two network architectures and joint detection as a method to detect the quality of fruit and defects simultaneously (i) multi-class detection and (ii) layer parallel task. Third, ensure that the fruit will only be counted once using the tracking-via-detection approach. It can track and calculate fruit sequentially without using additional sensors and makes it easier for us to check the defect level on the fruit accurately using a video streaming camera

MATERIAL AND METHODS

In this research, we propose a system that can automatically classify mangoes, pitaya, and limes and detect defects in the fruit's surface. There are three approaches that we propose in this study, first by classifying each fruit as one of the three classes, 'a', 'b' and 'c'. Second, we create a system capable of estimating the level of maturity and defects based on the class of fruit. Third, we use video tracking. Methods for estimation of maturity and detection of fruit defects were inspired by Deep Fruits [8], however, we made an additional improvement by modifying previously trained network structures [14] (based on VGG-16 Architecture) to estimate fruit maturity and detect defects found on fruit surfaces. In the video tracking system, we count the number of fruits using the camera and simultaneously classify the quality of the fruit as good, medium, or defect. We also create a simple framework to ensure that the camera can classify fruit accurately and count the number of fruits in sequence without using a three-dimensional (3D) camera.

Design System

We develop a system that can classify and estimate fruit quantity on the conveyor runway. The design system of the sorting machine is shown in the following Fig.

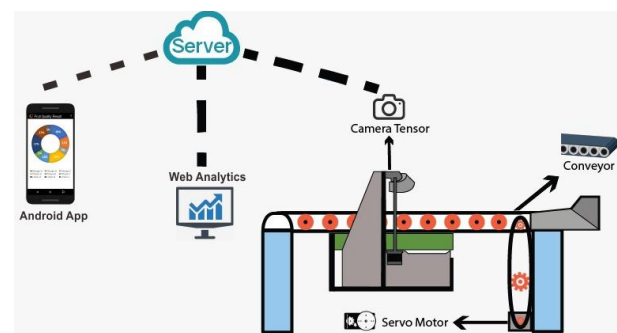


Fig 1. The architecture of sorting machine proposed

Fig. 1 shows a master plan of the system that will be created. the conveyor uses a camera to detect fruit. The results from the camera will be stored on the database server. Next, we create a real-time analysis chart that can be accessed using web applications and android app. At an early stage, the focus of our discussion is the detection system and fruit quantity estimation. The system

design that we propose can be seen in the picture below.

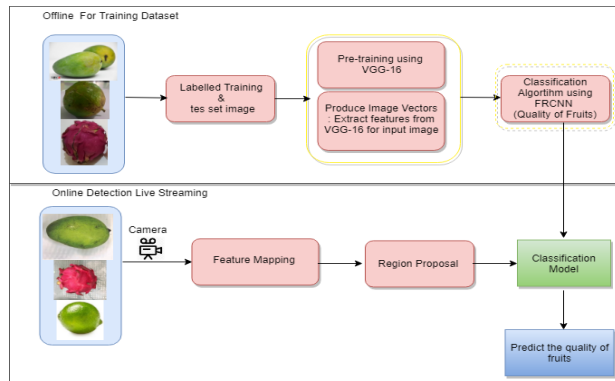


Fig 2. System Design

Datasets

In this study, we collect data real from farmers located in Gresik regency with 11,820 images that are further divided into 3 categories, A, B, and C or defect categories. Category A is characterized by a fruit that is shiny, green and has no defects. The B category fruit has the characteristics of a dull, green fruit and its surface has a rather yellowish color, while the defect fruit is yellow. Mangoes with category A, are those that are green and have white faded spots. A fruit in category B is a fruit that is too mature, making its shelf life in the market shorter. A Category B mango is a fruit that has a dull green color mixed with yellow and when held feels soft. A Category C mango is a fruit that has a defect on the surface of its fruit. This fruit is mostly used for bird feed so the price of the fruit in the market is very cheap. The three fruit categories above also have different prices, a fruit in Category A has the most expensive price and the fruit in the reject category has the cheapest price. Categories from all the datasets that we have collected can be seen in the table 1.

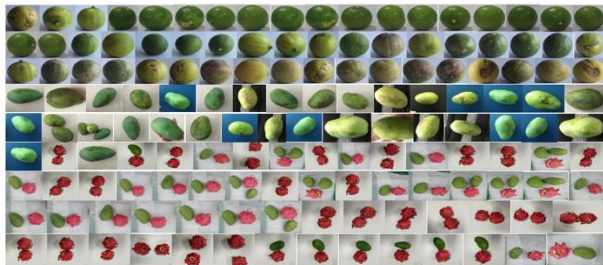


Fig 3. Sample of Fruits datasets

Table1. Row Images of Fruit Datasets to be Extracted

Feature Master	Class Label	Data Training	Data Testing	Total Dataset
Limes fruits	1. limes_A	800	160	3.768
	2. limes_B	1340	268	
	3. Jeruk Defect	1000	200	
Mango fruits	4. mango_A	350	70	3.012
	5. mango_B	1300	260	
	6. mango Defect	860	172	
Pitaya fruits	7. Pitaya_A	1300	260	5.040
	8. Pitaya_B	800	160	
	9. Pitaya_Defect	2100	420	

Image Labelling

The next stage after collecting images is an annotation process to label the segmentation process. We crop the image to 300x300 pixels and apply a label from one of 9 classes (limes A, limes B, limes defect, mango A, mango B, mango C, pitaya A, pitaya B, and pitaya defects). The output of this process will be the Region of Interest (ROI). We use Imglabelling tools to annotate images by labeling them one by one. The form of labeling can be seen in the picture below.

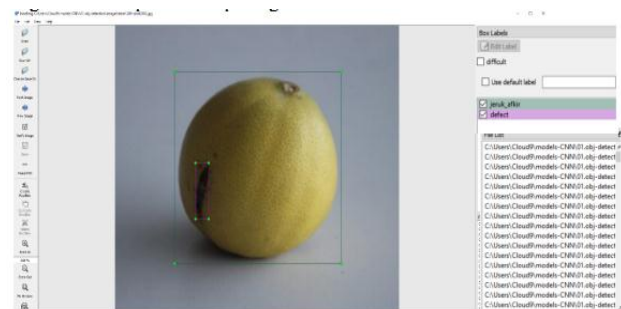


Fig 4. Sample of Fruits datasets

After the image is labeled one by one, labeled the data is stored in the form of an XML file. This XML file contains information on a fixed spatial distance from (h = high; w = width) while h and w are fixed hyper-parameters of each particular ROI. To easily process data, the contents of the XML file are converted into files that have the extension .csv.

Fruit Detection Module Using Faster RCNN

The framework that we propose is similar to the approach used by Deep Fruits, but we propose two new models to estimate the quality of the fruit and its detection system. Our initial stage detects each fruit based on their class sequences to determine super, medium, and defect fruit categories using multiclass Faster R-CNN. The next stage uses parallel layers (Parallel FRCNN) as one method to detect fruit and group fruit qualities together. More details can be seen in Fig. 5.

In this step, we carried out the detection of fruits using the FRCNN architecture by utilizing the Network Region proposal as a method to detect objects in real-time. We modified it using 8 convolution layers with a 300x300 image size and filter window 3x3 tread 2 on each filter. This technique is a refinement of the VGG-16

architecture [14] previously trained by Image Net which then produces a fully-connected layer output that contains a collection of aggregate scores. We executed this because the Image Net has 1000 classes, in the VGG16 architecture it consists of 4096-4096-4096 neurons in the hidden layer and 1000 neurons in the output layer, whereas in our case, we have 9 classes (limes-A, limes-B, limes-defect, mango-A, mango-B, mango-defect, Pitaya-A, pitaya-B, and pitaya defect). This is the main reason we made modifications to the FC layer of the VGG-16. Subsequently, in the Multi Class FRCNN and parallel-FRCNN, the same parameters are used to make it easier for us to adjust the initial weight on the 8 convolution layers that have been made.

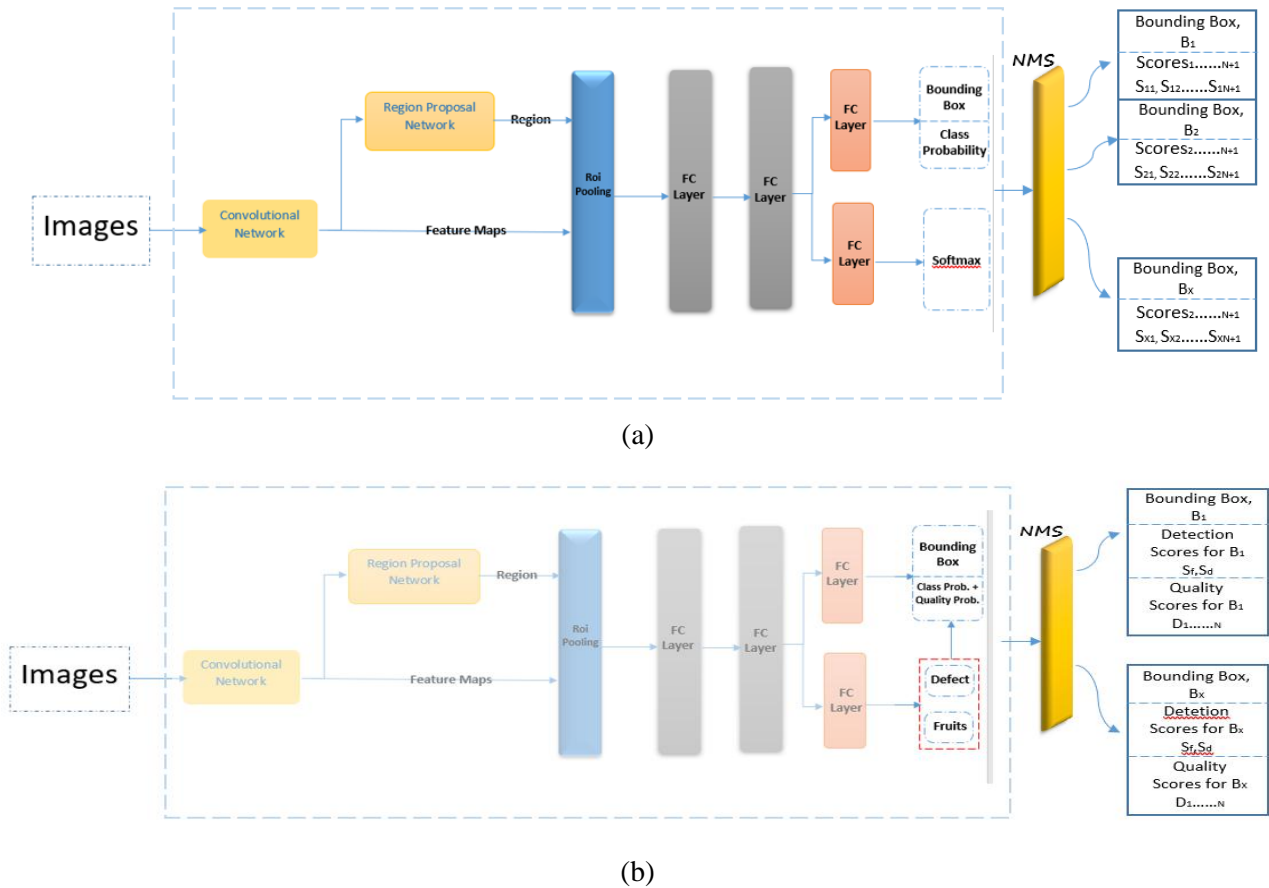
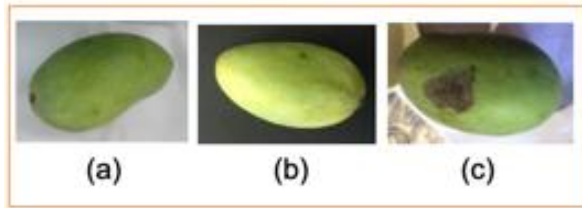
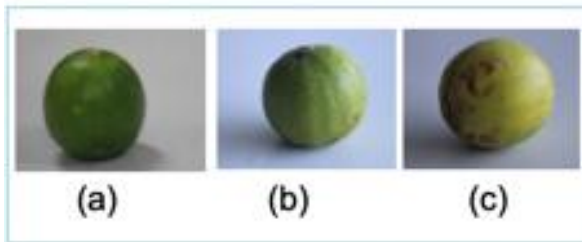


Fig 5. (a) Network Structure for Training Image using Multi-Class Faster R-CNN, (b) Network Structure Using Layer Parallel-Task Faster R-CNN

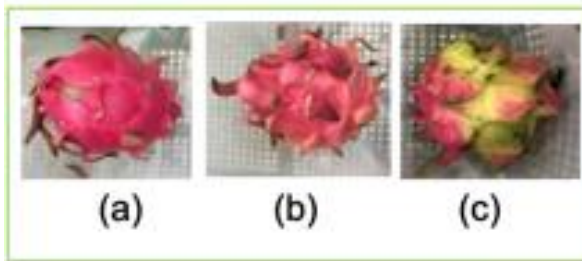
In this case, a forward propagation and backpropagation process is carried out in each layer, where each layer will be trained one by one until it is stable. When a layer is stable, we move on to the next layer and repeat the entire process using the same method. Once all the layers have been trained, the output will produce a Map feature.



(a)



(b)



(c)

Fig 6. (a) From left to right, mangoes classified as Super, Middle, and Defect. (b) From left to right, limes classified as Super, Middle, and Defect. (c) From left to right, pitayas classified as Super, Middle, and Defect.

Produce optimal performances, the optimization of each layer is required using a momentum optimizer with several 0.9, the weight of 0.0006, and a learning rate of 0.00001. Next, we rotated, flipped and cropped in the image to multiply the dataset without losing the core or essence of the data.

- Multi-Class Faster R-CNN used Multiple Object class for detection and estimation quality of fruits: fruit quality estimation is given the symbol N where the fruit can be classified according to fruit type (S_f) and quality of fruit (S_d) is given the formula $N+1$. For more details, see in Fig. 2 (a). This is a simple strategy to provide information on fruits that have defects, based on type. However, if the number of fruit datasets in each type is different, it is difficult to estimate the quality of fruit that has defects.

Parallel (layer) Faster R-CNN: in this case, we divide into two stages of the process. The first layer serves to classify the variety of fruit. The second layer estimates the quality of fruits that have defects, where the estimated quality of the fruit is given a symbol of class N . The layer will be activated when there is fruit on the conveyor runway. Each layer will back-propagation and forward propagation to produce a stable feature. When the fruit is moving on a conveyor runway, the first and second layers will produce a cross-entropy class or loss L_d (fruit detection layer) and L_q (defect detection layer). Then, we calculated the total loss of both using formula:

$$L_{tot.} = L_d + L_q .$$

If the fruit is found detected. However, this strategy does not apply when there is no fruit running on the conveyor runway.

Object Tracking System

After the fruit is picked by the farmer, it is then stored in the basket and put into the conveyor. Fruit that moves on the conveyor runway will be classified according to the type and quality of the fruit. However, we must ensure that the fruit that runs on the conveyor is detected only once and is activated periodically to prevent duplicate amounts of data. This approach greatly facilitates farmers because they only use cheap cameras to carry out classification and detection. The tracking system uses a detector to obtain information about the latest scale and pose of the object detected. IOU used to take sample videos at a ratio of 40 frames per second to objects running on the conveyor runway.

Two steps must be done. In the first frame, we initialize the fruit to be detected, the fruit image taken from each frame is unique and sequential. Then the image is saved and initialized as the initial track. This requires a basic knowledge of computers to activate the camera when they look at the fruit moving on the conveyor runway. The steps we take are as follows:

- We calculate the intersection over Union (IOU) to determine which activates track (3).
- If the camera detects a new fruit, it will be recording as a new fruit track by:
 - Stages to compare threshold value from IOU, y_{merge} that activated and
 - Boundary threshold $y_{boundary}$, compared to the active track to determine whether the object is a new track based on points (1).

Stage 1: data from frame f -th $[D_{b,1} \dots, D_{b,K_b}]$ then compared to new detection with the currently active track $K[T_1 \dots, T_k]$. The IOU of each active track (K) and new detection in each frame will be calculated to search the highest value between the two T_M , and $D_{t,k}$. When IoU threshold value is higher than the value: T_M , and $D_{b,k}$, then $D_{b,k}$ is considered the same as T_M . Consequently, the position of T_M and $D_{t,k}$ will activate the track and update simultaneously. detection k , and active track from T_M are then deleted from the upcoming calculation. This process (1) will be repeated until no track is detected, or until IOU that best matches less than y_{d_i} , if a track is not considered active on three frames then it will be set to inactive and removed from the inactive tracklist.

Stage 2: remaining detection values $J[D_{f,1} \dots, D_{f,J}]$ will be counted as the candidate who made the new track. However, to solve it if detection must be calculated as a new track. First, detection will be compiled with active tracks and their IOU will be measured. This IOU is above the y_{merge} , detection will be removed from further consideration. Second, the limit measurement of the next detection will be compared with the active track, $S_{bdry}(T_m, D_j)$ and is calculated (See picture 2). Therefore, if the limit size is greater than the threshold y_{bdry} , then the detection will be removed from further consideration. After step (2) is completed, the new track will be initialized using the rest of the previous detection. During our experiment, IoU was unable to resolve several

cases. One of them is if an image has a large size and is on a small track. This causes many IOU values from these images to not proceed to the next step. Therefore, we determine the upper limit by measuring the limit by calculating how many j -th, D_j contained in them m th track.

$$S_{bdry}(T_m, D_j) = \frac{Area(T_m \cap D_j)}{Area(D_j)} \quad (1)$$

This method makes it easy for us to track new objects and add track based on detection results. Each track stores the following regression information output a region boundary with location coordinates (x, y, h, w) , the results of the classification on the area containing the object or not will determine the probability of 0 or 1 [22].

$$\begin{aligned} p^* &= 1 & \text{if } IoU > 0.7 \\ p^* &= -1 & \text{if } IoU < 0.3 \\ p^* &= 0 & \text{otherwise} \end{aligned} \quad (2)$$

$$\begin{aligned} t &= [(x - x^a)/w^a, (y - y_a)/h_a, \log w/w_a, \log h/h_a] \\ t^* &= [(x^* - x^a)/w^a, (y^* - y_a)/h_a, \log w^*/w_a, \log h^*/h_a] \end{aligned}$$

here w_a, h_a, x_a, y_a are with, height and center of anchor and w^*, h^*, x^*, y^* are the ground truth bounding box width, height, center.

RESULT AND DISCUSSION

The data presented herein were collected from one of the fruit farmer groups in the district of Gresik, East Java Province. The images we collected consist of 3 plant varieties, namely mango, pitaya, and lime. Each of the fruit varieties had a different harvest period. The image was captured using a Canon camera with a size of 300x300 pixels. Next, we annotate the images individually using labeling tools and store them in the form of data distribution.

Experiment 1: detection and quality estimation

Making a model that can recognize various classes is challenging, especially when large amounts of data and different variants are involved. Therefore, we try to use transfer learning to perform feature extraction in our case use transfer learning, we use the architectural model VGG-16 which had been trained before by Image Net. However, we made an architecture identical to VGG-16 without using the fully-connected-layer and downloading the weight. All weights from VGG-16 have been trained using the Image Net dataset and can recognize colors, shapes, textures, etc. Hence it is possible to use this model to perform feature extraction processes from all the images we have. Forward propagation and backpropagation processes in each layer will be trained individually until stable. When it is stable, we proceed to the next layer and repeat the training process using the same method. If all the layers have been retrained, then their output will produce a feature Map. This feature map is stored in file form and given the name train-feature.npy and Val-feature.npy. The file can be used to flatten or reshape the feature map into a vector, hence it can be used as input from the fully-connected layer.

Table 2. Computer Specification

Model Type	Specification
Processor	Chipset Intel®H110 Core™ i7 7700
RAM	24 GB
GPU	NVIDIA® GeForce GRT1060
GPU Memory	6 GB GDDR5
Clock Rate	3,60 GHz
Storage	500 GB SSD

Tensor board Graph Visualization makes it possible for us to see the learning process and see the debug of the scenario we are doing. Additionally, the Tensor board can graph up to thousands of nodes with a simple display. After getting the Feature Map from Feature extraction using the VGG-16 model, the feature map is then

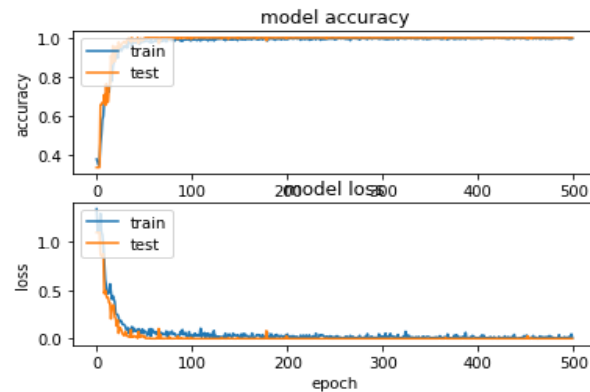


Fig 9. Model accuracy curve and Model loss curve Using Faster R-CNN.

By using the number of epoch 500 and a learning rate of 0.0001 the accuracy obtained in this scenario is 93.5% with loss cross-entropy of 0.02985. According to this scenario, we conclude that the feature extraction process is very influential on the accuracy of each learning and testing. Therefore, to produce maximum accuracy, more careful observation is needed to determine the model used in feature extraction. However, not all models that have good accuracy are suitable for use in the case of detection objects in real-time. We have used this model for object detection, but many objects are not even undetectable. However, many objects are incorrectly detected, for example, a mango-A, instead of being labeled as limes-C.

The accuracy obtained in the training process using the CNN Region-Based algorithm is very good. However, the task of solving the case of object detection in real-time is very slow. The experiment above is one of the implementations using Region-Based CNN(RCNN). From the observation process, the time needed to detect one image is 2 seconds. For this reason, the experiment applies the Faster R-CNN algorithm by utilizing a new method, namely the Region Proposal Network. The experiment also utilizes the same CNN, which requires very fast operating time. From our observations, this algorithm can detect video streaming objects by 5 frames per second. The following is a comparison table of the computational processes of several different methods.

Tabel 3. Computational average times with different methods

Methods	Running time per image(s)
FPN	0,1452
CNN	0,2121
Mask R-CNN	0,1022
FRCNN+MRP	0,1423
Faster R-CNN+ZF	0,0131

Table 4. The performance measure Recall Precision of Mango fruits

Fruits	Precision	Recall	F1-Score
Mango_A	0.95	0.84	0.80
Mango_B	0.80	0.88	0.77
Mango_C	0.79	0.90	0.80

Table 5. The performance measure Recall Precision of limes fruits

Fruits	Precision	Recall	F1-Score
Limes_A	0.90	0.78	0.89
Limes_B	0.82	0.67	0.78
Limes_C	0.95	0.85	0.88

Table 6. The performance measure Recall Precision of pitaya fruits

Fruits	Precision	Recall	F1-Score
Pitaya_A	0.97	0.81	0.90
Pitaya_B	0.94	0.66	0.77
Pitaya_C	0.89	0.75	0.89

Several methods even applied in our works to compare the accuracy of each experiment carried out. The results of the accuracy of several experiments that have been carried out can be seen in Fig. 11. Fig. 10 shows the optimal accuracy obtained used the Faster R-CNN Framework.

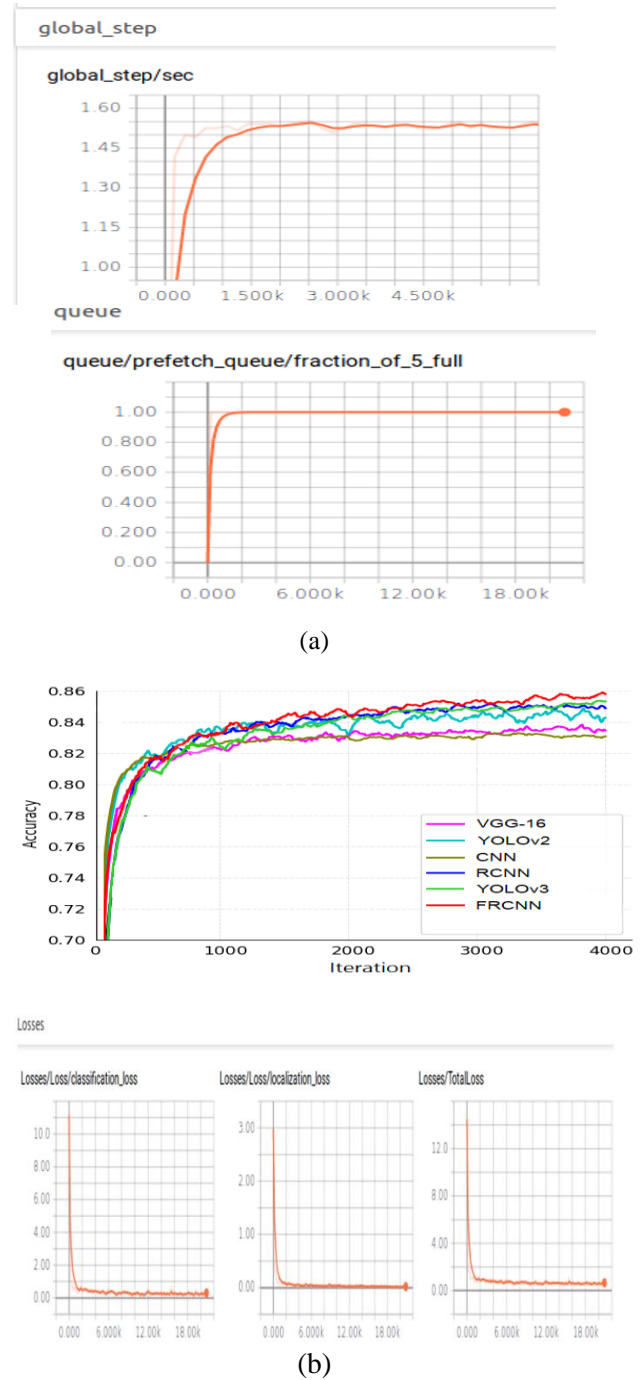


Fig 10. (a) The global step/sec is a performance indicator showing after 18.000 steps we processed as the training model. (c) show Total loss which trend to be the approach to zero after doing 18.000 steps, this shows performance better.

Experiment 2: Video Tracking

The table shows the result of a screenshot of our experiment in conducting a fruit detection process in real-time. From the results above, we conclude the process for object detection is remarkably constant, but there are shortcomings. When the fruit is inserted individually, the results of the detection area by the actual label. However, when fruit are entered simultaneously, some fruits do not match the correct label. This algorithm offers a solution for resolving cases of fruit quality detection in real-time managing many classes. In our future research, we aim to modify the architecture on feature extraction, so that it becomes capable of producing the best model and minimize errors in fruit quality detection.

Table 7. Testing on lemons




NO	DETECTION RESULT	TRUE LABEL	PREDICT RESULT
1		limes_A	limes_A
2		Limes_C / Defect	Limes_C/ Defect
3		limes_B	limes_B

Table 8. Testing on mangoes





NO	DETECTION RESULT	TRUE LABEL	PREDICT RESULT
1		Mango_A	Mango_A
2		Mango Defect/Afkir	Mango Defect/Afkir

Table 9. Testing on pitayas

NO	DETECTION RESULT	TRUE LABEL	PREDICT RESULT
1		Pitaya_B	Pitaya_B
2		Limes_B	Mango_B

CONCLUSION

In this paper, we present a system that classifies fruit with an accuracy level of 88% (mango), 83% (limes), 99% (pitaya) and with an average computation cost of 0.0131 m/s. We can track and calculate fruit sequentially without using additional sensors. additionally, we can also check the defect rate on fruit accurately using the video streaming camera. the datasets are taken in the field areas then trained using the frCNN framework using the tensor flow platform. Faster R-CNN is one model that proved possible in solving complex

computer vision problems with the same principle and shows incredible results at the beginning deep learning revolution. This algorithm provides a solution for solving cases of fruit quality detection in real-time using multi-classes. In our future research, we experiment need to be done and make android apps integrated with an analytics system. we also used the tracking video system to calculate the number of fruits on the conveyor runway. we can track and calculate fruit sequentially without using additional sensors and check the defect rate on fruit using the video streaming camera more accurately and with greater ease.

REFERENCES

- [1] In. Sub-directorate of Horticulture Statistics, *Statistics of Annual Fruit and Vegetable Plants Indonesia*. Indonesia: BPS-Statistics Indonesia, 2018.
- [2] E. B. Esguerra and R. Rolle, *Post-harvest management of mango for quality and safety assurance*. United Nations: Food and Agriculture Organization(FAO), 2018.
- [3] S. CODEX, *Standard for Pitahayas are classified in three classes defined*. United Nations: Codex Committee on Fresh Fruits and Vegetables (CCFFV), 2011.
- [4] S. CODEX, *Standard for Limes are classified in three classes defined*. United Nations: Codex Committee on Fresh Fruits and Vegetables (CCFFV), 2011.
- [5] H. Basri, I. Syarif, and S. Sukaridhoto, "Faster R-CNN Implementation Method for Multi- Fruit Detection Using Tensorflow Platform," 2018, no. October, pp. 1–4.
- [6] L. Ma, S. fadillah Umayah, S. Riyadi, C. Damarjati, and N. A. Utama, "Deep Learning Implementation using Convo-lutional Neural Network in Mangosteen Surface Defect Detection," 2017, no. November, pp. 24–26.
- [7] K. N. Ranjit, H. K. Chethan, and C. Naveena, "Identification and Classification of Fruit Diseases," vol. 6, no. 7, pp. 11–14, 2016.
- [8] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. McCool, "Deepfruits: A fruit detection system using deep neural networks," *Sensors (Switzerland)*, vol. 16, no. 8, 2016.
- [9] S. Jana and S. Basak, "Automatic Fruit Recognition from Natural Images using Color and Texture Features," *Devices Integr. Circuit*, pp. 620–624, 2017.
- [10] M. Kaur and R. Sharma, "Quality Detection of Fruits by Using ANN Technique," *IOSR J. Electron. Commun. Eng. Ver. II*, vol. 10, no. 4, pp. 2278–2834, 2015.
- [11] C. S. Nandi, B. Tudu, and C. Koley, "A machine vision-based maturity prediction system for sorting of harvested mangoes," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 7, pp. 1722–1730, 2014.
- [12] S. Basak, "An Improved Bag-of-Features Approach for Object Recognition from Natural Images," vol. 151, no. 10, pp. 5–11, 2016.
- [13] S. Nuske, S. Achar, T. Bates, S. Narasimhan, and S. Singh, "Yield Estimation in Vineyards by Visual Grape Detection (in process of publication)," *2011 IEEE Conf. Intell. Robot. Syst.*, pp. 2352–2358, 2011.
- [14] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv Technical Report*, 2014. [Online]. Available: <https://arxiv.org/abs/1409.1556>. [Accessed: 14-Feb-2019].
- [15] L. Agilandeewari, M. Prabukumar, and G. Shubham, "Automatic Grading System Mangoes Using Multiclass SVM Classifier," *Int. J. Pure Appl. Math.*, vol. 116, no. 23, pp. 515–523, 2017.
- [16] Indarto and Murinto, "Banana Fruit

- Detection Based on Banana Skin Image Features Using HSI Color Space Transformation Method,” *JUITA*, vol. 5, no. 1, pp. 15–21, 2017.
- [17] C. McCool, I. Sa, F. Dayoub, C. Lehnert, T. Perez, and B. Upcroft, “Visual Detection of Occluded Crop: For Automated Harvesting,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 2506–2512.
- [18] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution grayscale and rotation invariant texture classification with local binary patterns,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, 2002.
- [19] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 886–893.
- [20] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
- [21] S. Chen, “Counting apples and oranges with deep learning: A data driven approach,” *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 781–788, 2017.
- [22] Y. Lecun, Et All, Backpropagation Recognition, Applied To Handwritten , In *Proceedings - 30th IEEE Conference On Computer Vision And Pattern Recognition*, Cvp, 2015.