Automatic Detection and Calculation of Palm Oil Fresh Fruit Bunches using Faster R-CNN

Novian Adi Prasetyo[a], Pranowo[b] and Albertus Joko Santoso[b*]

[a]Departement of Informatics Engineering, Institut Teknologi Telkom Purwokerto, Jalan D.I. Panjaitan, Indonesia
[b]Magister Teknik Informatika, Universitas Atma Jaya Yogyakarta, Jalan Babarsari, Yogyakarta, Indonesia

Abstract: Indonesia is one of the countries with the largest industry of crude palm oil (CPO) in the world. During 2013-2017, the growth of the area of oil palm plantations in Indonesia decreased -0.52%, the decline is expected not to affect the amount of CPO production. One of the things that affect CPO production is the primary raw material availability of palm oil fresh fruit bunches (FFB). Raw material requirements can be predicted by several forecasting methods, but the methods only predict the raw material requirements FFB, not the availability. The development of deep learning eases humans in doing things. Deep learning can be used to calculate FFB automatically using the faster R-CNN algorithm. This study presented a system of automatic detection and calculation of FFB. The evaluation is carried out by comparing 4 network architectures; resnet inception V2, inception V2, resnet 50, and resnet 101. The results of this study indicate success in calculating FFB. The success is indicated by the results of evaluating the four network models with the average F1 scores above 80%.

Keywords: Palm oil fresh fruit bunches (FFB); Faster R-CNN; computer vision; object detection.

1. Introduction

Indonesia is the largest industry of crude palm oil (CPO) center in the world after Malaysia and Thailand [1]. Based on data from the Ministry of Agriculture 2018 [2], Indonesia produced 27.78 million tons CPO in 2013 and increased to 37.81 million tons in 2017, with an average growth of 2.13% per year in the period 2013-2017. In 2013, the value of Indonesia's CPO exports to the world amounted to USD 17.67 million or 59.97% of Indonesia's total plantation commodity exports and increased to USD 21.25 million or 66.81% in 2017 with an average growth of 26.41% per year. CPO is a source of non-oil foreign exchange for Indonesia, so the enhancement of CPO production in Indonesia is expected to improve the welfare of the nation.

During 2013-2017, the growth of the area of oil palm plantations in Indonesia decreased -0.52% [2], the decline is expected not to affect the amount of CPO production. One of the things that affect CPO production is the primary raw material availability of palm fresh fruit bunches (FFB). The raw material availability of FFB can be predicted by several methods such as Fuzzy Rule-Based Time Series Method [3] and linear regression [4]. Both forecasting methods only calculate the needs of raw material and cannot predict the availability of real raw material. Farmers can do calculations from close range and long distance. The close range calculation of the uneven and large FFB has been done manually and it is quite difficult and spends a lot of time to do. Long distance calculations certainly have a higher level of difficulty. In addition to limited vision, loss of concentration also increases the difficulty of distance calculation.

*Corresponding author; e-mail: albjoko@staff.uajy.ac.id

doi:10.6703/IJASE.202005_17(2).121
Received 5 December 2019
Revised 29 February 2020
Accepted 15 March 2020

Therefore there is a need for technology that can facilitate the calculation of availability of FFB associated with palm oil production in Indonesia.

Computer vision is a topic that is included in the field of deep learning that can be used to solve problems in various fields of human life. Computer vision can be used for classification [5, 6], face detection [7], semantic segmentation [8], object detection [9-11], and calculations [10].

There are 2 approaches to classification and detection, namely the machine learning approach in the form of Support Vector Machine (SVM) and the deep learning approach in the form of Convolutional Neural Network (CNN). The results of the Large Scale Image Classification and Recognition Challenge (ILSVRC) showed that there are many researchers use CNN to overcome object recognition and classification problems. Progress in this field is supported by computational capabilities and datasets [6]. Convolutional Neural Network is one of the deep learning methods resulting from the development of Multi-Layer Perceptron (MLP) which is designed to process two-dimensional data.

CNN has been used in the aviation field [9] to detect foreign objects on the airfield's sidewalks. In this study, the use of CNN from each network model has a level of precision above 89%. In the field of transportation, CNN is used by the automotive industry to provide pedestrian detection to provide information about the number of people on the street. In research T. Liu and T. Stathaki [8], the semantic segmentation framework that they developed was evaluated and compared with several existing frameworks and produced a 5.7% miss rate. In the field of security and defense, CNN is can be used as face detection to prevent crime. The study S. W. Cho et al. [7] proposed face detection in dark conditions with the help of a visible-light camera sensor and faster R-CNN development. The final results showed a 3.36% precision increase from the previous method. In the field of health, CNN is used to carry out food classifications as semi-automatic monitors of the daily diet. Research G. Ciocca et al. [5] evaluated food classifications based on available datasets and provided recommendations for improving larger datasets to maximize the results obtained. In the field of fashion, CNN is used to detect hair models on someone. The results in the study U. R. Muhammad et al. [12] showed that the level of accuracy achieved was around 90% and higher than in previous studies.

This study will utilize CNN in agriculture as has been done in previous studies [13] which segmented apples to estimate apple orchard yields to make it easier for farmers to plan harvests. The study C. Zhang et al. [14] made modifications to the resnet 50 network to improve the detection and classification of tomato flowers, ripe tomatoes, and unripe tomatoes using faster R-CNN. The results showed that there was an increase in mAP of 5.5%. Automatic fruit counting is also done as in research W. Maldonado and J. C. Barbosa [15] which calculated green oranges and tomatoes [10], counting apples, mangoes, and almonds [16], counting mangoes [17, 18], and strawberry [19]. Literature review on deep learning in agriculture [17, 20] has not shown the use of this technology in FFB detection or calculation.

Based on the description above, research on oil palm will be carried out by using the convolutional neural network (CNN) as a solution to carry out FFB detection and calculation automatically.

2. Materials and Methods

2.1 Hardware and Software

This research used a computer with an Intel Core i7 3770K processor, 16GB Ram, and NVIDIA GTX980. It used Linux Ubuntu 16.04 OS, python 3.6 programming language, and tensor flow framework. To support the research dataset's annotation, this research used COCO Annotation [21].

2.2 Dataset

Collection of datasets was carried out through scrapping FFB images on the internet in jpg format with 100 images by 181 x 278 pixels to 1300 x 956 pixels. After the dataset has been collected, the annotation was carried out with the COCO dataset format [22]. It resulted in 536 FFB annotations. The annotated images can be seen in Figure 1.

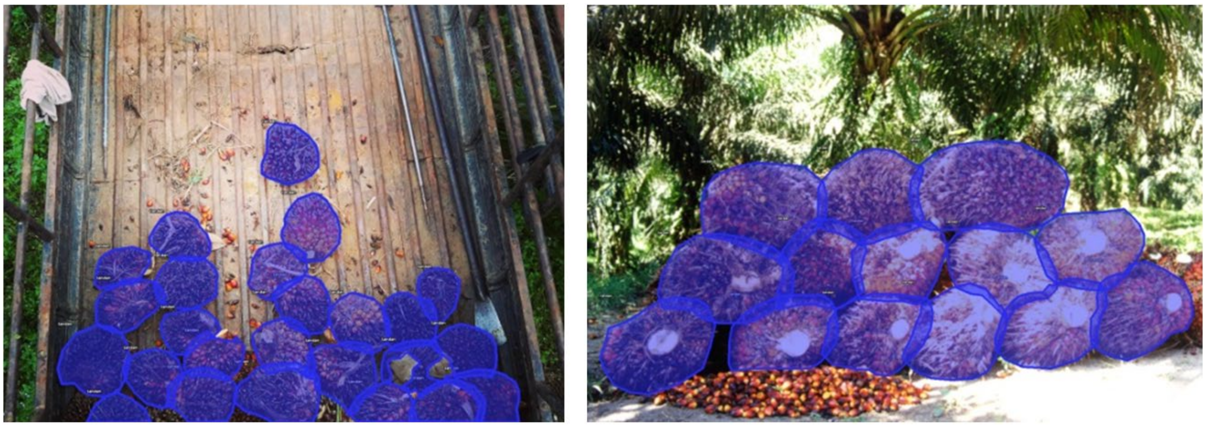


Figure 1. Example FFB with annotation.

5-fold cross-validation method was used to avoid the overfitting effect [23] in evaluating the results of network model training. The dataset was divided into 5 partitions with 20 random images in each partition. Furthermore, 5 experiments were carried out on each network model with a combination of 4 partitions as training and 1 partition as validation. The 5-fold cross-validation design can be seen in Figure 2.

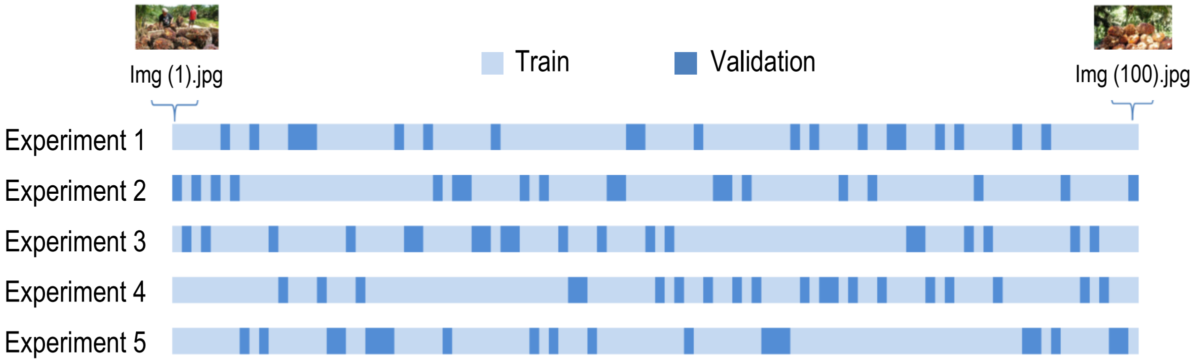


Figure 2. 5-Fold cross validation design.

2.3 Regional-based Convolutional Neural Network

Convolution Neural Network (CNN) is part of deep learning which consists of several convolutional, ReLu, and pooling layers that resemble the human visual system [10]. Convolution networks consist of Convolution layers, pooling layers, and fully connected. CNN has made a lot of progress because it is supported by computing capabilities and datasets used for training [6].

Regional-based Convolutional Neural Network (R-CNN) is a combination of region proposals and CNN or referred to as Regional-based Conventional Neural Network (R-CNN). R-CNN is the development of CNN where classification only focuses on one object and is tasked with explaining the object. But when viewed from a broader perspective, it can be seen that there are many objects in the image, there are complex sights, overlapping objects, and diverse backgrounds. These problems cannot be solved through classification. As seen in Figure 3, the purpose of R-CNN is to detect and localize an object in the image [24]. In R-CNN, images are detected using simple detection feature techniques (such as edge detection and others) to obtain Regions of Interest (RoI), this process is also referred to as selective search [25].

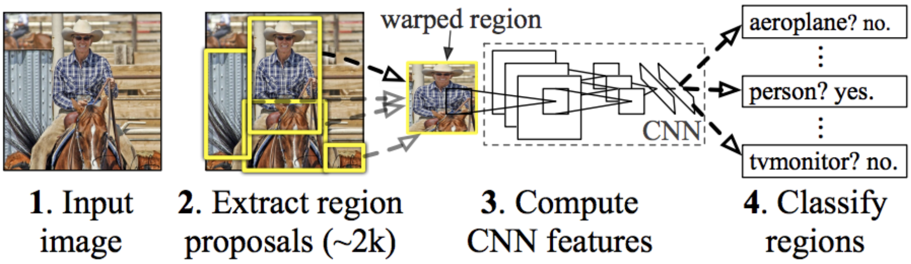


Figure 3. Process of detection and localization R-CNN [35].

R-CNN works quite well but is very slow. In 2015 found a solution to the problem that occurred in the R-CNN architecture [26], so that it is created Fast R-CNN. As shown in Figure 4, Fast R-CNN replaces the SVM classifier with a softmax layer above CNN to produce a classification. In addition, Fast R-CNN also added a linear regression layer parallel to the softmax layer to the output boundary box coordinates. In this way, all the output needed comes from a single network [26].

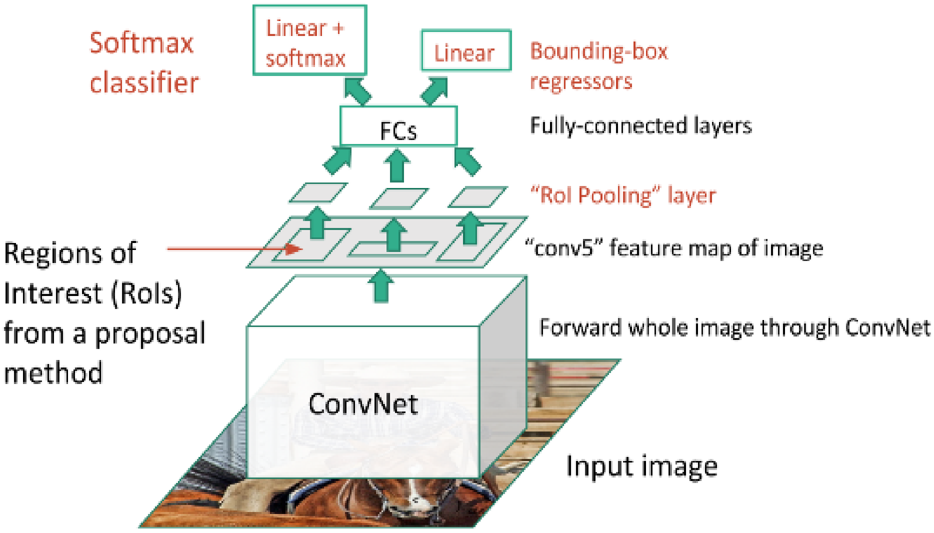


Figure 4. Architecture Fast R-CNN [26].

In 2015 the research team from Microsoft [26, 27] proposed development of Fast R-CNN namely Faster R-CNN. Fast R-CNN detects objects by making multiple detection boxes in the part that has the potential to be detected. This is a fairly slow process and hinders the entire process.

Faster R-CNN is a development of Fast R-CNN [26] where the architecture consists of two modules namely RPN and fast R-CNN detector [27]. RPN is a small network neuron that is in the last row of the convolution layers network and serves to predict the existence of the required object and also predict the bounding box on that object. Furthermore, detection is done by classifying the object results obtained by the RPN based on the object class. The detailed architectural details of Faster R-CNN can be seen in Figure 5.

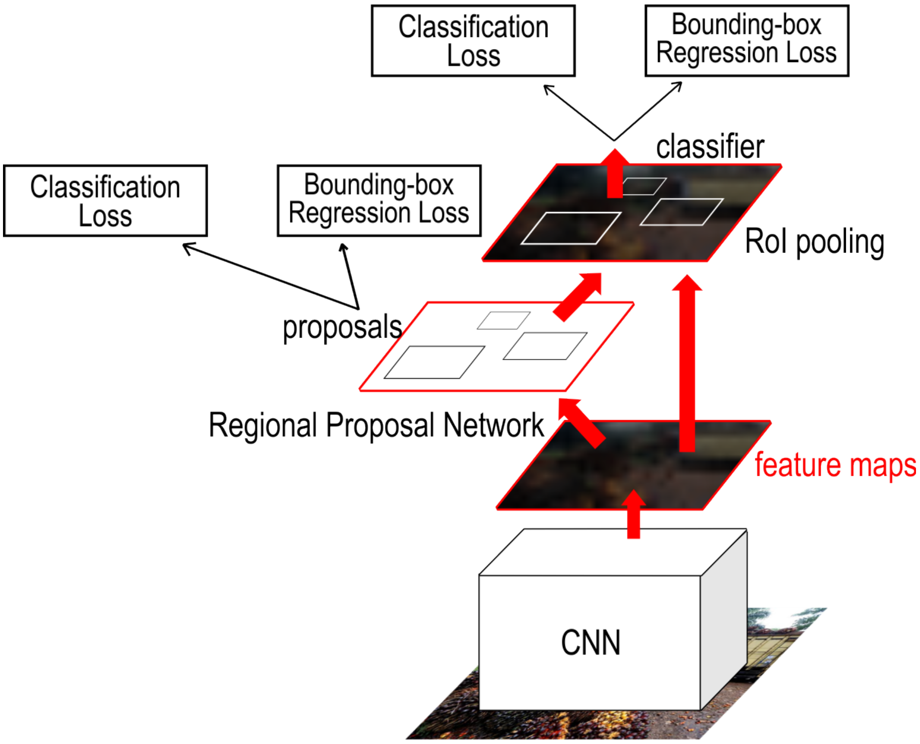


Figure 5. Architecture Faster R-CNN [28].

2.4 Network Architecture Model

This study used 4 network architectural models namely Inception Resnet V2, Inception V2, Resnet50 and Resnet101 that used as training data.

2.4.1 Inception

Inception is a development from googleNet and became the 1st place for image classification at ILSVRC in 2014 [29]. The Inception model provides an easier process in the convolutional layer section and empirically is able to learn more representations on fewer parameters. The inception model will independently see the correlation between cross-channel and spatial [10]. This architecture was originally introduced by C. Szegedy et al. [29] as Inception V1, then refined to Inception V2 [30], Inception V3 [31], and the latest was Inception-Resnet [32] which has become the best-performing family model in the ImageNet dataset [33]. The architecture of Inception V2 is designed to reduce the complexity of CNN by developing a wide-ranging architecture rather than depth. Inception has 3 modules shown in Figure 6.

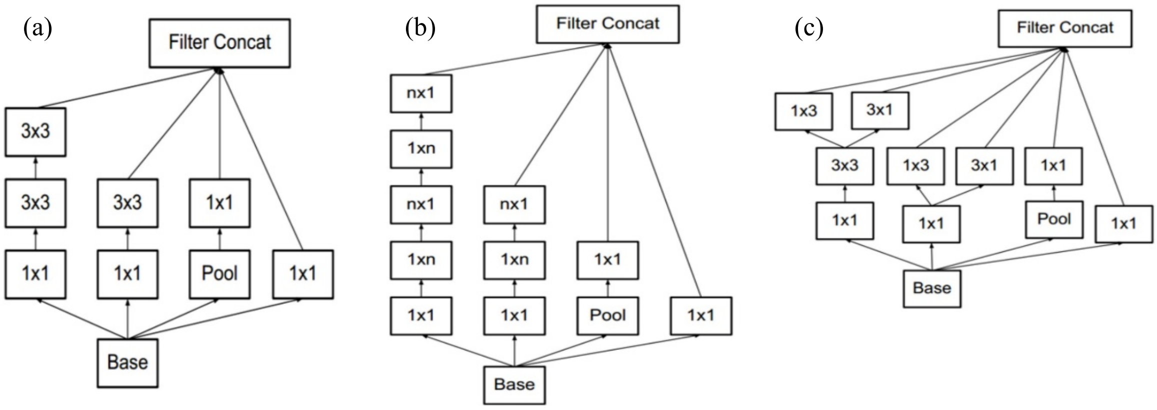


Figure 6. Module inception [31].

The first module in Figure 6(a) replaces the 5 × 5 convolutions to 3 × 3. Furthermore, convolution factoring is shown in Figure 6(b). Finally, the module is changed to be wider to reduce the complexity of convolution networks that is shown in Figure 6(c) [31]. The use of all three modules in the architect network inception is shown in Table 1.

Table 1. Architecture inception [31].

Type	Patch Size/Stride or Remarks	Input Size
Convolution	3×3/2	299×299×3
Convolution	3×3/1	149×149×32
Convolution Padded	3×3/1	147×147×64
Pooling	3×3/2	147×147×64
Convolution	3×3/1	73×73×64
Convolution	3×3/2	71×71×80
Convolution	3×3/1	35×35×192
3× Inception	Figure 6(a)	35×35×288
5× Inception	Figure 6(b)	17×17×768
2× Inception	Figure 6(c)	8×8×1280
Pooling	8×8	8×8×2048
Linear	logits	1×1×2048
Softmax	classifier	1×1×1000

2.4.2 Resnet

Resnet is a network model developed by Microsoft and has won 1st place in the 2015 ILSVRC competition [34]. The ResNet Model uses deep residual learning framework. By using this framework, each network layer has a reference to the previous network layer; this makes the optimization process easier than the network-layer network layer that has no connection. The easier optimization process leads to the more layers formed by the neural network (34 layers) so that it has higher accuracy than the neural network that does not use residual networks [34].

3. Result

After all datasets have been successfully trained with each using 5000 epochs, the detection and automatic calculation of oil palm has been successfully carried out and has been tested using several images that have been prepared for testing with 4 neural network convolution network architectures.

3.1 Evaluation Criteria

To evaluate the test results, parameters will be created by grouping the detection results as follows:

3.1.1 True Positive (TP)

Detection box with a positive class and produces a true value. This is called the detection box that successfully detects bunches.

3.1.2 False Positive (FP)

Detection box with a positive class and produces an incorrect value. It is called a detection box that detects objects instead of bunches.

3.1.3 False Negative (FN)

Detection box with a negative class and produces an incorrect value. It is called the undetectable bunches.

Based on the parameters above, the evaluation calculation formula is determined as in Table 2 below:

Table 2. Evaluation formula

Definisi Evaluasi	Formula
Precision Rate (P) is the level of accuracy of the detection produced.	$P = \frac{TP}{TP + FP}$
Recall Rate (R) is the level of success in making detection.	$R = \frac{TP}{TP + FN}$
False Negative Rate (FNR) is a positive value proposition that is considered wrong.	$FNR = \frac{FN}{TP + FN}$
False Alarm (FA) is negative value proposition with true positive.	$FA = \frac{FP}{TP + FP}$
F1 Score is a measure used to find a balance between precision and recall.	$F1 = 2 \times \frac{P * R}{P + R}$

3.2 Analysis

The analysis was carried out by evaluating the results of system detection and calculation compared to the results of manual calculations, the criteria for the images tested can be seen in Table 3.

Table 3. Image for testing.

Image	Occlusion			Total
	0%	1-75%	75-99%	
Image 1	12	11	5	28
Image 2	7	5	4	16
Image 3	1	3	1	5
Image 4	3	4	1	8
Image 5	2	6	9	17
Image 6	1	12	11	24
Image 7	3	6	2	11
Image 8	5	3	8	16
Image 9	4	1	4	9
Image 10	7	8	23	38

Figure 7 is an example of a picture that has been manually counted and will be used for testing, calculations were carried out gradually as shown in Table 3 based on 0% occlusion (bunches that are not covered by other objects), occlusion 1-75% (bunches covered by other objects in where the closed area is not greater than 75%), and occlusion is 75-99% (bunches covered by other objects where the closed area is greater than 75%).

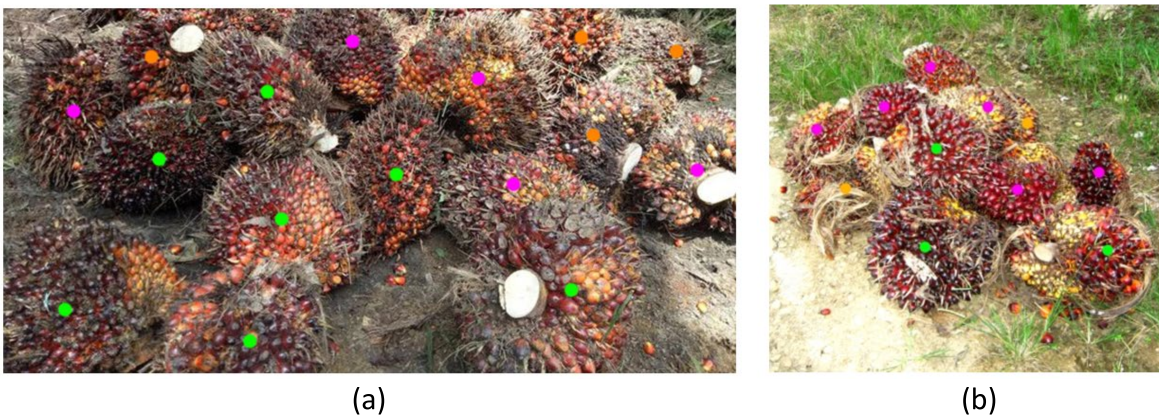


Figure 7. Example image for testing. Section (a) is image testing with name Image 2, section (b) is image testing with name Image 7. Green dots are occlusion 0%, purple dots are occlusion 1-75% and orange dots are occlusion 75-99%.

The evaluation results of detection of the Inception Resnet V2 model can be seen in Table 4, the Inception V2 model can be seen in Table 5, the Resnet 50 model can be seen in Table 6, and the Resnet 101 model can be seen in Table 7. The evaluation criteria used are in accordance with the explanation in section 3.1.

Table 4. Results of evaluation network architecture Inception Resnet V2.

Evaluation Criteria	Exp-1	Exp-2	Exp-3	Exp-4	Exp-5	Average
P	97%	96%	97%	98%	97%	97%
R	74%	69%	62%	73%	78%	71%
FNR	26%	31%	38%	27%	22%	29%
FA	3%	4%	3%	2%	3%	3%

Table 5. Results of evaluation network architecture Inception V2.

Evaluation Criteria	Exp-1	Exp-2	Exp-3	Exp-4	Exp-5	Average
P	95%	97%	97%	98%	98%	97%
R	79%	72%	69%	74%	82%	75%
FNR	21%	28%	31%	26%	18%	25%
FA	5%	3%	3%	2%	2%	3%

Table 6. Results of evaluation network architecture Resnet 50.

Evaluation Criteria	Exp-1	Exp-2	Exp-3	Exp-4	Exp-5	Average
P	95%	97%	97%	98%	98%	97%
R	79%	72%	69%	74%	82%	75%
FNR	21%	28%	31%	26%	18%	25%
FA	5%	3%	3%	2%	2%	3%

Table 7. Results of evaluation network architecture Resnet 101.

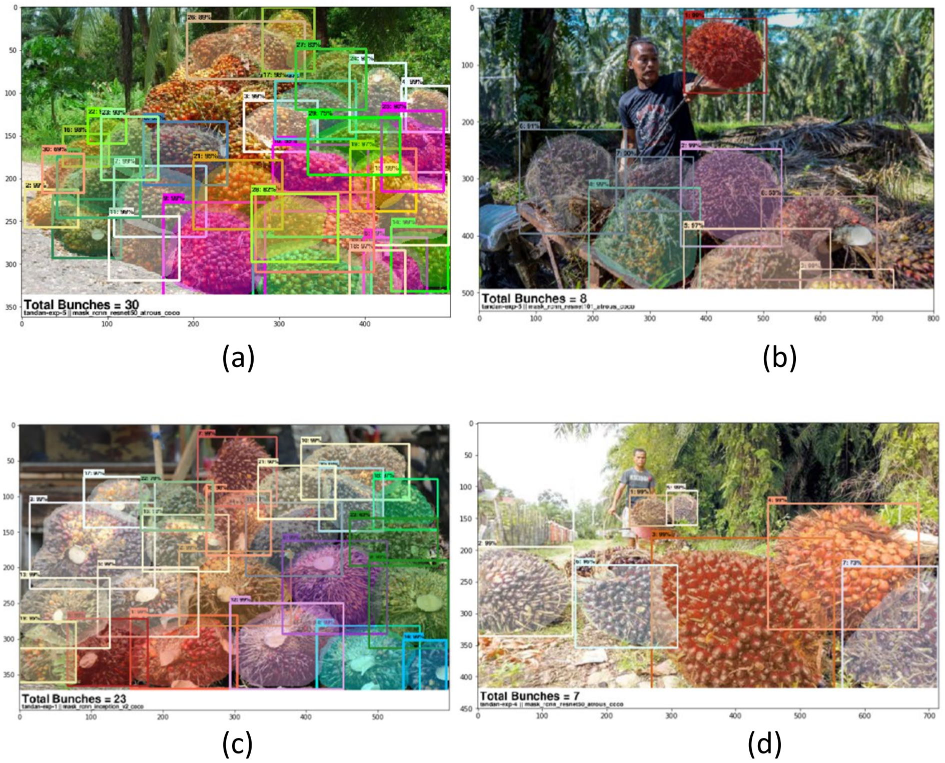
Evaluation Criteria	Exp-1	Exp-2	Exp-3	Exp-4	Exp-5	Average
P	95%	97%	97%	98%	98%	97%
R	79%	72%	69%	74%	82%	75%
FNR	21%	28%	31%	26%	18%	25%
FA	5%	3%	3%	2%	2%	3%

F1 Score results from each model can be seen in Table 8. It can be seen that the average F1 score of 5 experiments each network model has a value above 0.8, then the highest value is in the Resnet 50 network model.

Table 8. Result of testing.

Model	Exp-1	Exp-2	Exp-3	Exp-4	Exp-5	Average
Inception Resnet V2 (F1 Score)	84%	81%	76%	84%	87%	82%
Inception V2 (F1 Score)	86%	82%	80%	85%	90%	85%
Resnet 50 (F1 Score)	86%	83%	83%	90%	90%	86%
Resnet 101 (F1 Score)	80%	81%	81%	85%	85%	82%

Some detection results can be seen in Figure 8. Figure 8(a) is an example of test results in Image 6. The detection box was able to select all objects (bunches) but the detection box number 10 could not be included in the "true positive" criteria because there were 2 objects detected, so that they were included in the "false positive" criteria. Figure 8(b) is an example of a test result in testing image with name Image 9. The results of the detection carried out showed that only 8 objects (bunches) were included in the "true positive" criteria and 1 object (bunch) was not detected so that it included in the "false negative" criteria. Figure 8(c) is an example of test results in testing image with name Image 10, based on manual calculating, this image has the object with the highest occlusion of 75-99% so that it affected the results of the detection carried out where there were 9 objects detected that were included in the "false negative" criteria. Figure 8(d) is an example of a test result in testing image with name Image 4. There was 1 object (bunch) so that it was included in the "false negative" criteria where the object also has occlusion of 75-99%.



(a) (b) (c) (d)

Figure 8. Result of detection and calculation FFB.

4. Discussion

This study has proposed a detection system and FFB automatic counting using Faster R-CNN with 4 different network architectures. The training data that has been collected and annotated later was used in training to use Faster R-CNN as many as 5 experiments. The resulting inference was each tested with 10 images. In this section, we will discuss the strengths and weaknesses of the system. The annotation process carried out on all images can produce a number of labeled FFBs. The annotation process was carried out with the polygon tool, as seen in Figure 9.

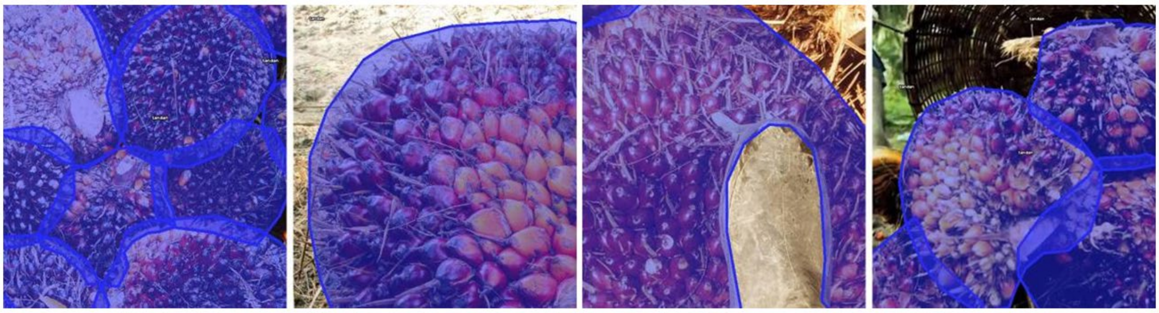


Figure 9. Detail selection line during annotation process.

The selection was carried out by slightly dragged the object to see a shift of pixel between object and background or between objects and other objects in an annotation. This was done to see that the annotations that occurred were colliding with each other. This method resulted in a fairly good detection result, but there was a wrong detection result that the detection found 2 FFBs that considered as 1 FFB as in Figure 10.

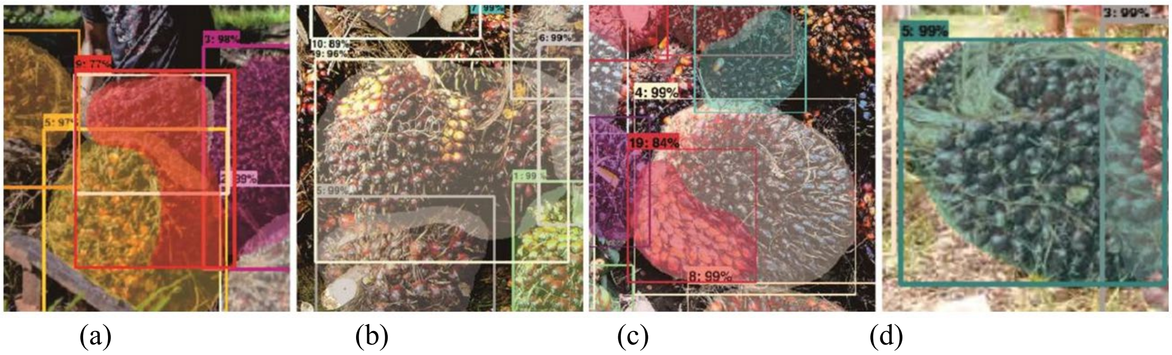


Figure 10. Example of error detection; (a) bounding box number 9 detects 2 bunches; (b) bounding box number 9 detects 2 bunches; (c) bounding box number 4 and 19 detect the same bunches; (d) bounding box number 5 detects 2 bunches.

This error occurred due to the stack of bunches that were too tight and made it unclear to find the contact between a bunch to the other bunch. In Figure 11, there were undetectable bunches and this was caused by the complexity of a less varied bunch annotations, causing less optimal recognition of bunches.

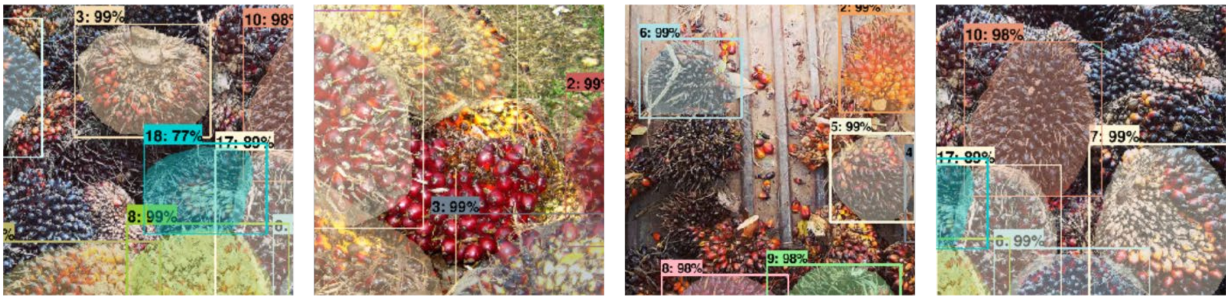


Figure 11. Example not detected.

5. Conclusions

This study presented a system of automatic detection and calculation of FFB using Faster R-CNN. Testing of automatic detection and calculation compare with manual count have resulted average F1 score above 80%. Some things that cause the detection of bunches were the high level of occlusion and the presence of several objects that cover the bunch. To improve the results of detection and calculation, it is necessary to add a dataset with better quality. Even small pixel datasets need to be collected to create a wider FFB detection and calculation area. Modifications on each network architecture can be a new discovery for detection and calculation of FFB better results.

References

[1] *Palm Oil Production by Country in 1000 MT*. 2019. Available: https://www.indexmundi.com/agriculture/?commodity=palm-oil&graph=production.

[2] Kariyasa, K., Susanti, A. A. and Waryanto, B. 2018. Center for Agricultural Data and Information System Ministry of Agriculture Republic of Indonesia. *Agricultulral Statistics*.

[3] Rahim, N. F., Othman, M., Sokkalingam, R. and Abdul Kadir, E. 2018. Forecasting Crude Palm Oil Prices Using Fuzzy Rule-Based Time Series Method. *IEEE Access*, 6: 32216–32224.

[4] Oettli, P., Behera, S. K. and Yamagata, T. 2018. Climate Based Predictability of Oil Palm Tree Yield in Malaysia. *Sci. Rep,* 8, 1: 1–13.

[5] Ciocca, G., Napoletano, P. and Schettini, R. 2018. CNN-based features for retrieval and classification of food images, Comput. *Vis. Image Underst*, 176: 70–77.

[6] Anwar, I. and Islam, N. U. 2017. Learned features are better for ethnicity classification. *Cybern. Inf. Technol*, 17, 3: 152–164.

[7] Cho, S. W., Baek, N. R., Kim, M. C., Koo, J. H., Kim, J. H. and Park, K. R. 2018. Face detection in nighttime images using visible-light camera sensors with two-step faster region-based convolutional neural network. *Sensors (Switzerland)*, 18, 9.

[8] Liu, T. and Stathaki, T. 2018. Faster R-CNn for robust pedestrian detection using semantic segmentation network. *Front. Neurorobot*, 12: 1–10.

[9] Cao et al, X. 2018. Region based CNN for foreign object debris detection on airfield pavement. *Sensors (Switzerland)*, 18, 3: 1–14.

[10] Rahnemoonfar, M. and Sheppard, C. 2017. Deep count: Fruit counting based on deep simulated learning. *Sensors (Switzerland)*, 17, 4: 1–12.

[11] Liu, G., Mao, S. and Kim, J. H. 2019. A Mature-Tomato Detection Algorithm Using Machine Learning and Color Analysis. *Sensors*, 19, 9: 2023.

[12] Muhammad, U. R., Svanera, M., Leonardi, R. and Benini, S. 2018. Hair detection, segmentation, and hairstyle classification in the wild. *Image Vis. Comput*, 71: 25–37.

[13] Bargoti, S. and Underwood, J. P. 2017. Image Segmentation for Fruit Detection and Yield Estimation in Apple Orchards. *J. F. Robot*, 34, 6: 1039–1060.

[14] Zhang, C., Yue, P., Di, L. and Wu, Z. 2018. Automatic Identification of Center Pivot Irrigation Systems from Landsat Images Using Convolutional Neural Networks. *Agriculture*, 8, 10: 147.

[15] Maldonado, W. and Barbosa, J. C. 2016. Automatic green fruit counting in orange trees using digital images. *Comput. Electron. Agric*, 127: 572–581.

[16] Bargoti, S. and Underwood, J. 2017. Deep fruit detection in orchards. *Proc. - IEEE Int. Conf. Robot. Autom*, 3626–3633.

[17] Koirala, A., Walsh, K. B., Wang, Z. and McCarthy, C. 2019. Deep learning – Method overview and review of use for fruit detection and yield estimation. *Comput. Electron. Agric*, 162: 219–234.

[18] Kestur, R., Meduri, A. and Narasipura, O. 2019. MangoNet: A deep semantic segmentation architecture for a method to detect and count mangoes in an open orchard. *Eng. Appl. Artif. Intell*, 77: 59–69.

[19] Habaragamuwa, H., Ogawa, Y., Suzuki, T., Shiigi ,T., Ono, M. and Kondo, N. 2018. Detecting greenhouse strawberries (mature and immature), using deep convolutional neural network. *Eng. Agric. Environ. Food*, 11, 3: 127–138.

[20] Kamilaris, A. and Prenafeta-Boldú, F. X. 2018. Deep learning in agriculture: A survey. *Comput. Electron. Agric*, 147: 70–90.

[21] Brooks, Justin. 2019. *COCO Annotator*. https://github.com/jsbroks/coco-annotator/ (June 11, 2019).

[22] Lin et al, T. Y. 2014. Microsoft COCO: Common objects in context. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, 8693 LNCS, 5: 740–755.

[23] Xiang, X., Lv, N., Guo, X., Wang, S. and El Saddik, A. 2018. Engineering vehicles detection based on modified faster R-CNN for power grid surveillance. S*ensors (Switzerland)*, 18, 7.

[24] Girshick, R., Donahue, J., Member, S. and Darrell, T. 2015. Region-based Convolutional Networks for Accurate Object Detection and Segmentation, 8828: 1–16.

[25] Uijlings, J. R. R., Van, K. E. A. de Sande, T. Gevers and Smeulders, A. W. M. 2013. Selective search for object recognition. *Int. J. Comput. Vis*.

[26] Girshick, R. 2015. Fast R-CNN. *Proc. IEEE Int. Conf. Comput. Vis*, 1440–1448.

[27] Ren, S., He, K., Girshick, R. and Sun, J. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell*, 39, 6: 1137–1149.

[28] Huang et al, J. 2017. Speed/accuracy trade-offs for modern convolutional object detectors. *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR*, 2017-Janua, 3296–3305.

[29] Szegedy, C. et al. 2015. Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–12. Available: https://ieeexplore.ieee.org/document/7298594.

[30] Ioffe, S. and Szegedy, C. 2015. Batch Normalization : Accelerating Deep Network Training by Reducing Internal Covariate Shift. *Mach. Learn*.

[31] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z. 2016. Rethinking the Inception Architecture for Computer Vision. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit*, 2016-Decem, 2818–2826.

[32] Szegedy, C., Ioffe, S., Vanhoucke, V. and Alemi, A. 2016. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *Comput. Vis. Pattern Recognit*.

[33] Schillaci, G., Pennisi, A., Franco, F. and Longo, D. 2012. Detecting tomato crops in greenhouses using a vision based method. *Int. Conf. Saf. Heal. Welf. Agric. Agro-food Syst*, 1: 20–26.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-Decem, pp. 770–778, 2016.

[35] Girshick, R., Donahue, J., Darrell, T. and Malik, J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit*, 580–587.