International Journal of Remote Sensing and Earth Sciences Vol. 16 No. 1 June 2019: 87– 98

DETECTING AND COUNTING COCONUT TREES IN PLEIADES SATELLITE IMAGERY USING HISTOGRAM OF ORIENTED GRADIENTS AND SUPPORT VECTOR MACHINE

Yudhi Prabowo[1,*] and Kenlo Nishida Nasahara[2]
[1]Remote Sensing Technology and Data Center,
Indonesian National Institute of Aeronautics and Space
[2]Faculty of Life and Environmental Sciences,
University of Tsukuba
*E-mail: yudhi.prabowo@lapan.go.id

Received: 2 January 2019; Revised 14 August 2019; Approved: 20 August 2019

Abstract. This paper describes the detection of coconut trees using very-high-resolution optical satellite imagery. The satellite imagery used in this study was a panchromatic band of Pleiades imagery with a spatial resolution of 0.5 metres. The authors proposed the use of a histogram of oriented gradients (HOG) algorithm as the feature extractor and a support vector machine (SVM) as the classifier for this detection. The main objective of this study is to find out the parameter combination for the HOG algorithm that could provide the best performance for coconut-tree detection. The study shows that the best parameter combination for the HOG algorithm is a configuration of 3 x 3 blocks, 9 orientation bins, and L2-norm block normalization. These parameters provide overall accuracy, precision and recall of approximately 80%, 73% and 87%, respectively.

Keywords*: coconut trees, Pleiades imagery, tree detection, histogram of oriented gradient, support vector machine*

1 INTRODUCTION

A major weakness of coconut-sector development in Indonesia is the nature of land ownership. Most of the plantation areas belong to smallholders who use very limited cultivation technology and deliver low productivity (Dasgupta, 2014). Information related to the spatial distribution and number of coconut trees is therefore required to support and monitor the development of coconut-tree cultivation, in relation to factors such as the age or health of trees.

The simplest way to detect coconut trees is by manually marking every single tree on satellite imagery or on field surveys, using GPS to collect the positions of coconut trees and then overlaying their point coordinates on the image. However, if there are a lot of coconut trees in areas which may be large and contain more than 1000 trees, manual detection and field surveying can be a time-consuming and expensive process. Hence, remote sensing techniques are needed.

The new generation of imaging satellites, such as the GeoEye, WorldView and Pleiades satellite series, supports sub-metre spatial resolution. The imagery created by these satellites allows detailed objects to be visually identifiable. This increase in spatial resolution has changed the focus of many remote sensing studies, which have started to analyze not only classes of objects but also each object individually (Blaschke, 2010). In remote sensing, computer vision algorithms are often adopted for extracting geobiophysics information from the earth's surface, such as trees (Li et al., 2017), buildings (Liasis & Stavrou,

http://dx.doi.org/10.30536/j.ijreses.2019.v16.a3089

2016), and roads (Shahi, Shafri, Taherzadeh, Mansor, & Muniandi, 2015). Previous research into coconut-tree detection has been conducted by Bernales et al. (2016) using a template-matching method integrated with SVM, but it was found to be effective only for detecting sparsely growing coconut trees. In this study, we seek a robust feature descriptor that can distinguish coconut trees from cluttered backgrounds and under varied illumination in either sparse or densely cultivated areas.

The very small size of coconut trees makes detecting them using satellite imagery a challenging task. Moreover, these trees commonly grow naturally scattered and mixed with other trees, and with irregular distances between them. This differs from oil palm trees which are typically planted regularly and in exclusive plantation areas clear of other trees (Shafri, Hamdan, & Saripan, 2011). Other factors, such as variable appearance and different levels of illumination from the sunlight, may also cause misidentification. The extraction of reliable and detailed information for a single coconut tree from remote sensing imagery therefore requires sufficient spectral and geometrical resolution.

In this paper, the researchers propose the use of a histogram of oriented gradients (HOG) algorithm, commonly used in the field of computer vision, to extract features of objects from images. The HOG feature can represent information about shape and local object appearance via the distribution of local intensity gradients or edge directions. The performance in detection of coconut trees using various HOG parameters is also evaluated. The objective of this study is to identify the best HOG algorithm parameters for detecting and counting coconut trees in satellite imagery with sub-metre spatial resolution.

2 MATERIALS AND METHODOLOGY

2.1 Data

The study area is located on the seashore of Kebumen Regency in the southern part of Central Java Province, at geographical coordinates 7°45'21.36"S, 109°28'29.4"E.

The study examined a panchromatic band of Pleiades imagery with a spatial resolution of 0.5 metres acquired in February 2014. The data has been radiometrically corrected using TOA reflectance value and orthorectified using the rational polynomial function (RPF) method. For orthorectification, this study used RPC parameters available in the Pleiades imagery product and DEM SRTM 30m.

Coconut trees appear as very small features in Pleiades imagery. The average size of a coconut-tree crown is approximately 16 x 16 pixels. In addition, most coconut trees are randomly scattered and mixed with other types of trees, and this may cause misidentification. Some of these other trees are the same height as the coconut trees, with some being even taller and thus covering parts of the coconut-tree crown.

For validation, the study used drone-based aerial photographs with higher spatial resolution acquired in April 2017. The flight was planned to take photographs at an altitude of 100 metres with endlap and sidelap of 75% and 60%, respectively. This set of aerial photographs was geometrically rectified and mosaiced using Agisoft Photoscan software. The researchers also conducted an image-to-image co-registration process from mosaic aerial image to Pleiades satellite image to adjust the pixels, shifting to achieve better overlapping. The resulting mosaic image has spatial resolution of 0.03 metres, clearly showing single coconut trees. However, since the study area

experienced significant changes over the three years between the production of the satellite imagery and the aerial photography, judicious analysis was applied. Many coconut trees had been felled and replaced with buildings, roads, swimming pools, and so on.

2.2 Histogram of oriented gradients

The general flowchart for this study is shown in Figure 2-1. In the initial step, a group of 16 x 16 pixel sample images were cropped from the Pleiades imagery for use in SVM training. This image dataset was separated into two categories: positive images and negative images. Positive images are samples which contain coconut trees whereas negative images represent non-coconut trees do not.

A number of positive images were cropped manually to the same size as the sliding detection window, whereas negative images were cropped from the regions of the image representing open water, paddy fields, bare land, man-made objects, other trees, etc. Then, each image from both categories was extracted to the HOG feature vectors used as the input parameters in the training SVM. A linear SVM predicts each detection window as either coconut tree or non-coconut tree based on the HOG feature vector and the SVM model which resulted from the training process.

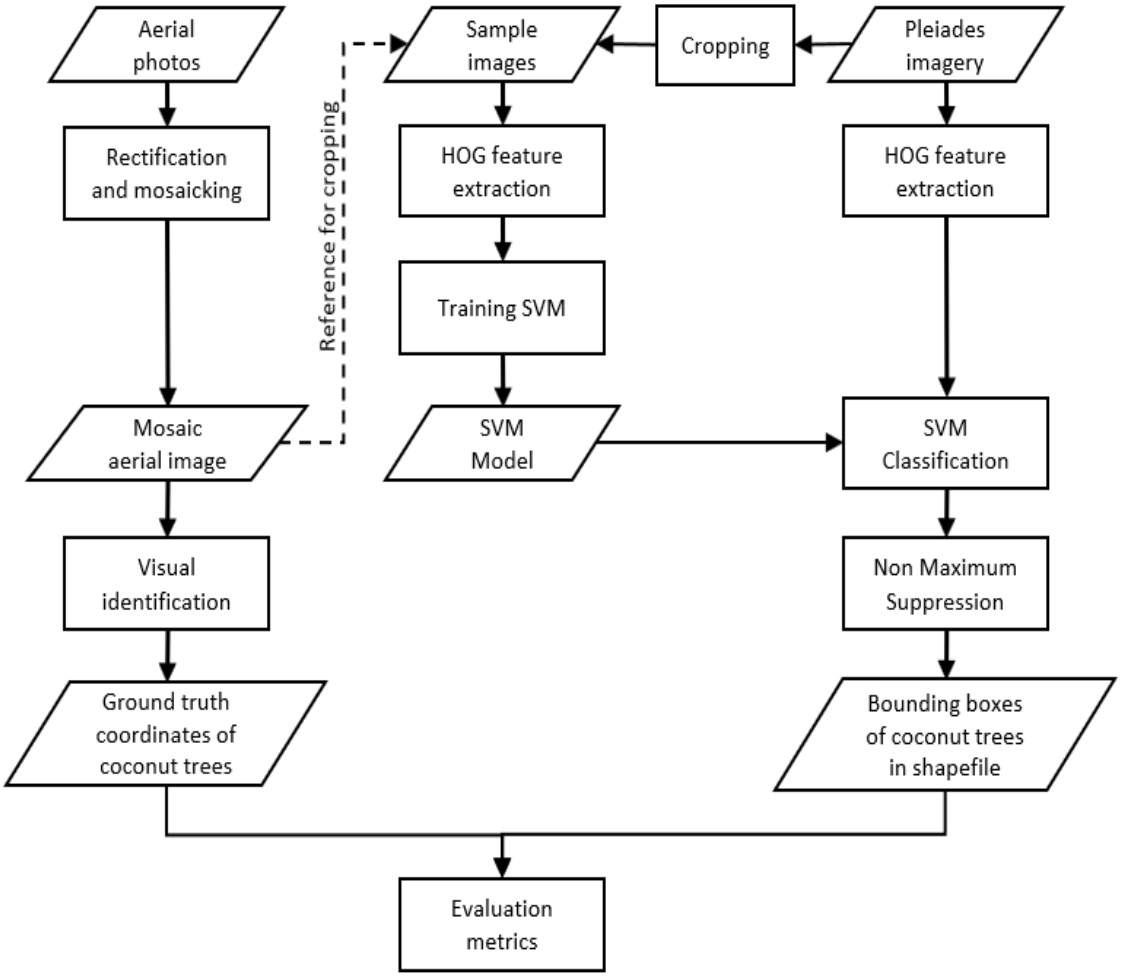

Figure 2-1: Flowchart for coconut tree detection.

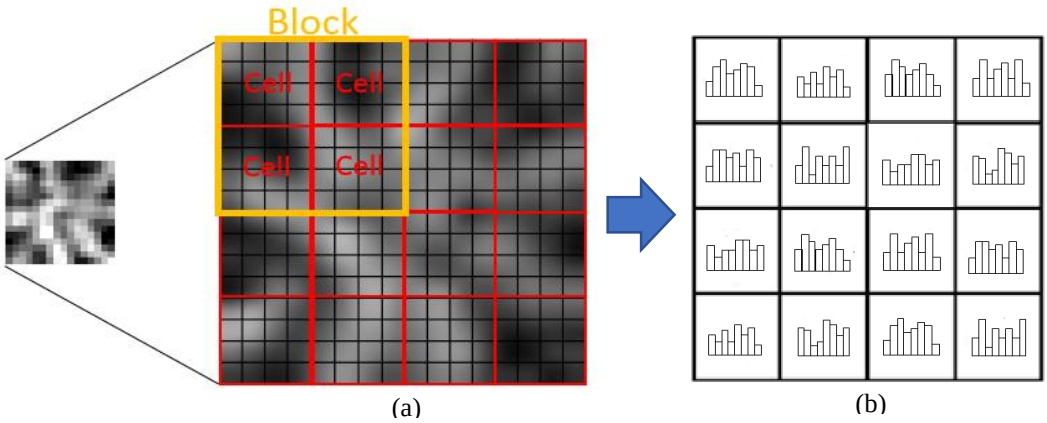

Figure 2-2**:** Illustration of HOG descriptor: (a) division of window image into cells (b) histograms for each cell in the image.

The HOG algorithm is one of the most common features of extraction algorithms in computer vision and image processing. It was first introduced by Dalal and Triggs (2005) for human detection. This method has been widely used to solve a variety of problems in object detection, including pedestrian recognition and tracking (Wójcikowski, 2016), hand-gesture recognition (Žemgulys et al. 2018), face recognition (Dadi and Mohan Pillutla, 2016), eye detection (Savakis, Sharma, & Kumar, 2014), and body-part recognition for tracking (Corvee & Bremond, 2010). This is because the HOG algorithm has proven effective for classification purposes and provides high detection accuracy via its simple calculations. The HOG descriptor has also been used in tasks other than human detection, such as car detection (Yan, Yu, Yu, & Fan, 2016), animal detection (Rangda & Hanchate, 2014), and fruit detection (Shruthi, 2012). HOG features can represent information about shape and appearance of local objects by the distribution of local intensity gradients or edge directions. It therefore provides good resistance to changes of illumination and shadowing (Baghdadi and Latif, 2015).

This method is based on evaluating well-normalized local histograms of image gradient orientation in a dense grid (Dalal and Triggs, 2005). In practice, the HOG detector uses a sliding detection window which moves throughout the image from the top left to the bottom right. At each window step, a HOG feature vector is calculated. The present study used a fixed-sized detection window in accordance with the average size of coconut-tree crowns of 16 × 16 pixelsThis window image is then divided into small spatial regions, called cells, of 4 × 4 pixels. Each 2 × 2 cell is grouped into one block, as shown in Figure 2-2. For each cell, all the gradient directions are calculated and accumulated into a local histogram of N-orientations bins. Blocks are typically set to overlap each other, so that the same cell may be used in several blocks.

In general, object detection using the HOG algorithm was performed via the three main steps: gradient computation, orientation binning, and block normalization. The detail explanation of each step is as follows:

2.2.1 Gradient computation

The first step in HOG feature extraction is the computation of image gradients. The image gradient is information about a directional change in the intensity or colour of each pixel in the image. The gradient information has two important properties: magnitude and orientation. For each pixel within the image $I(x, y)$, the gradients in the x-direction (Gx) and y-direction (Gy) are

calculated by convolving the image array with a 1D Sobel kernel [-1 0 1]. There are many more complex kernels, such as Prewitt, Canny or diagonal kernels, but these kernels generally result in poorer performance (Dalal and Triggs, 2005).

$$Gx = [-1\ 0\ 1] * I(x,y) \quad (2\text{-}1)$$
$$Gy = [-1\ 0\ 1]^T * I(x,y) \quad (2\text{-}2)$$

The gradients of x and y (*Gx, Gy*) are then combined to calculate the magnitude (*G)* and the orientation (θ), using the formulae below:

$$G = \sqrt{Gx^2 + Gy^2} \quad (2\text{-}3)$$
$$\theta = tan^{-1}\left(\frac{Gy}{Gx}\right) \quad (2\text{-}4)$$

2.2.2 Orientation binning

In this step, the histogram of the oriented gradient is built. Each pixel in the image holds a weighted vote for the orientation bins, based on the gradient magnitude from the previous step. For every cell, the magnitudes of the gradients are accumulated in a 1D histogram with a specified number of bins. Each pixel votes for one or two of the histogram bins, according to its orientation. The orientation bins are spaced over the range 0° to 360° (signed gradient) or 0° to 180° (unsigned gradient) (Kobayashi, Hidaka, & Kurita, 2008). Since the orientation of a pixel is considered in linear interpolation, the weighted votes are calculated by multiplying magnitude and weight for orientation. Then, the weighted votes are distributed into only two bins (Kim & Cho 2014). On the other hand, orientations and positions of all pixels in the block are considered in trilinear interpolation. Trilinear interpolation applied in this step is not only voting in relation to one cell histogram of orientation but also to the three neighbouring cell histograms in the same block, as shown in Figure 2-3. Block normalization

Blocks are typically overlapped in the detection window, so that each block may contain multiple cell histograms from other blocks. All components of cell histograms in each block are normalized to enhance detection performance. There are several steps in performing block normalization, as follows:

$$L2-norm: v \rightarrow v = \frac{v}{\sqrt{||v||_k^2 + \varepsilon^2}} \quad (2\text{-}5)$$

$L2-hys$: $L2-norm$ followed by clipping and renormalizing (2-6)

$$L1-norm: v \rightarrow v = \frac{v}{||v||_k + \varepsilon} \quad (2\text{-}7)$$
$$L1-sqrt: v \rightarrow v = \sqrt{\frac{v}{||v||_k + \varepsilon}} \quad (2\text{-}8)$$

in which (v) represents the non-normalized vector that contains the entire histogram elements in a single block, ($||v||_k$) is its k-norm for k = 1,2, and (ε) is a small constant value to avoid division by zero.

The final HOG feature vector is obtained by concatenating all components of the normalized cell from all blocks in the detection window into one large vector. This vector size should be the total of $N \times (C \times C) \times (B \times B)$, in which N is the number of orientation bins, $(C \times C)$ is the dimension of the cell, and $(B \times B)$ is the dimension of the block in a detection window.

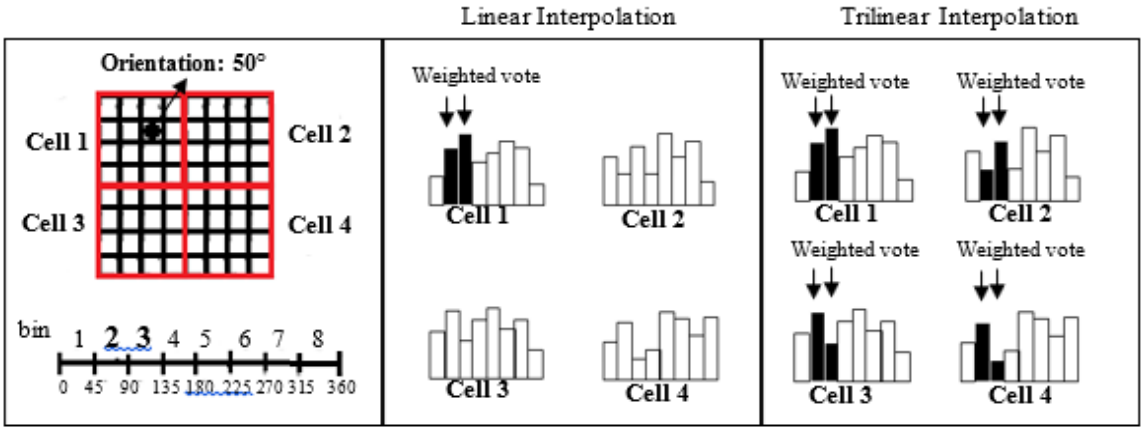

Figure 2-3: Illustration of linear and trilinear interpolation in one block.

The HOG feature vector of each detection window is then classified as either coconut tree or non-coconut tree using the trained SVM. The location of the detection window classified as the coconut tree candidate will be localized within a bounding box, but the desired output of an object detection system is not entirely clear, because whatever method the researchers use almost certainly detects multiple bounding boxes surrounding the same object in the image. To correct this, post-processing was performed to reduce the number of redundant bounding boxes in the same detected object.

2.3 Non-maximum suppression

Non-maximum suppression is a popular post-processing method for eliminating redundant object detection windows (Neubeck & Van Gool, 2006), requiring the bounding boxes coordinates and the threshold value for minimum tolerated overlap. Using the coordinates of top left and bottom right of each bounding box $(x_1, y_1, x_2, y_2,)$, the area of overlap can be calculated:

$$\text{overlap} = \frac{\text{area}(\text{box}_1 \cap \text{box}_2)}{\text{area}(\text{box}_1 \cup \text{box}_2)} \quad (2\text{-}9)$$

This is an iterative process. It takes each bounding box and starts calculating the overlapping area of two bounding boxes. If the overlap area of two bounding boxes is more than a tolerated threshold, it will be deleted. Then the process continues with the next bounding box and repeats the same process until no more bounding boxes are left to compare.

2.4 Evaluation metrics

To evaluate the performance of coconut-tree detection quantitatively, precision, recall, and overall accuracy were calculated by comparing the result with ground truth.:

$$\text{precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (2\text{-}10)$$

$$\text{recall} = \frac{\text{TP}}{\text{TP+FN}} \quad (2\text{-}11)$$

$$\text{overall accuracy} = \frac{\text{precision+recall}}{2} \quad (2\text{-}12)$$

In this context, true positive (TP) is the number of coconut trees correctly detected by the proposed method, false positive (FP) is the number of non-coconut trees detected as coconut trees, and false negative (FN) is the number of coconut trees not detected. Precision can be interpreted as the probability that a detected coconut tree is valid, and recall is the probability that a coconut tree in ground truth is correctly detected.

For validating the result, the ground truths of coconut trees were collected manually by visual interpretation from two images (mosaic aerial photos and Pleiades satellite imagery). When performing detection, the coordinates of

the coconut-tree crowns may deviate from the ground truth. To cope with this, a threshold of 10 pixels was set as a maximum deviation. The Euclidean distance from the centre coordinate of the bounding box to the coordinates of ground truth was calculated. If the distance from the centre of the bounding box to the closest ground truth is less than 10 pixels, it will be classified as true positive (TP); otherwise, it will be classified as false positive (FP).

3 RESULTS AND DISCUSSION

3.1 Results

The results of detection were visualized with a bounding box as the representation of a coconut tree. Figure 3-1 shows the bounding boxes overlaid on the panchromatic band of Pleiades imagery on the left (Figure 3-1a) and the mosaic aerial image on the right (Figure 3-1b). The bounding boxes from the detection process are still in the pixel coordinate system.

Therefore, the coordinate system of the bounding box must be converted from the pixel coordinate system to the projected coordinate system of the input image in order to be overlaid.

However, the images contain significant differences. Some bounding boxes are located on ground without coconut trees.

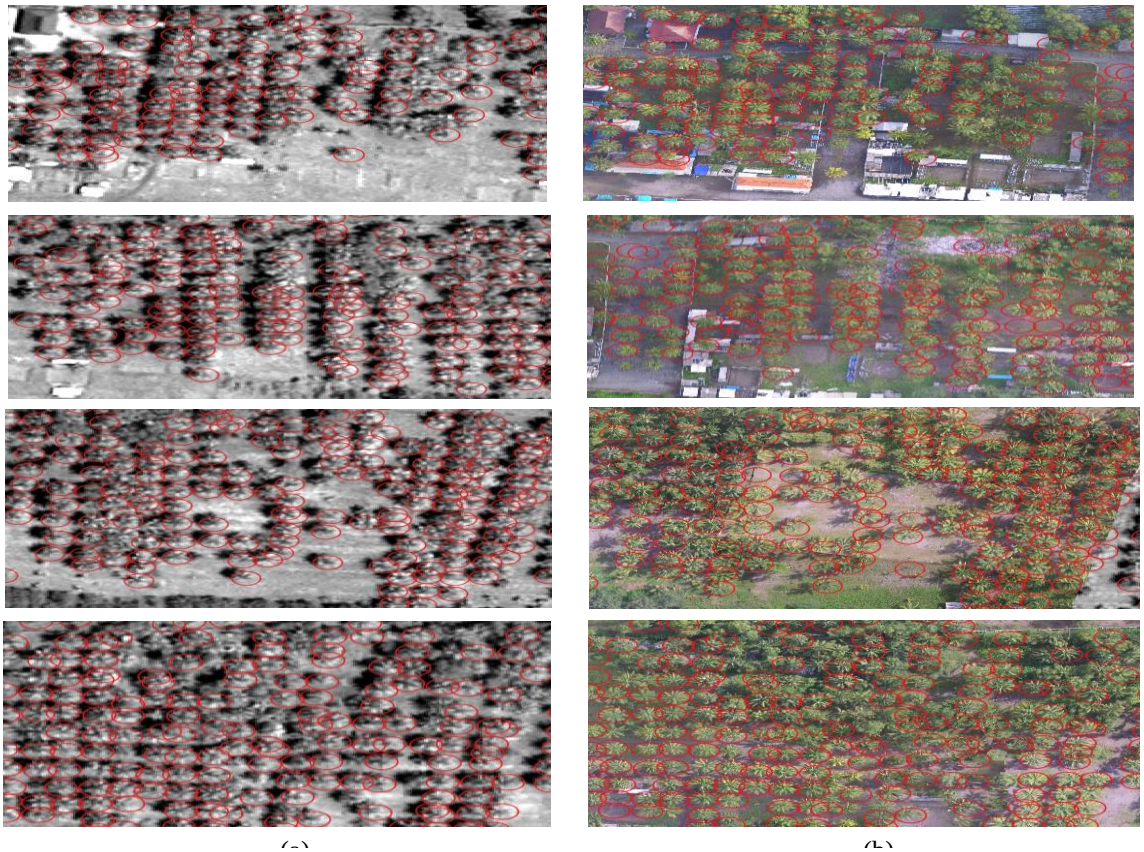

Figure 3-1**:** Detection results overlaid on (a) Pleiades images and (b) mosaic aerial images.

Table 3-1: The evaluation metrics of coconut-tree detection with block configuration of 2 x 2 (without overlapping). The proportion of training samples for this performance is 550 positive images to 19,500 negative images.

Norm	n-bins	Detected object	TP	FP	FN	Precision	Recall	Accuracy
L2-norm	6	2068	1463	605	568	0.707	0.72	0.7135
	7	2090	1473	617	558	0.705	0.725	0.715
	8	2048	1471	577	560	0.718	0.724	0.721
	9	2104	1503	601	528	**0.714**	0.74	**0.727**
	10	2166	1525	641	506	0.704	0.751	**0.7275**
L1-sqrt	6	1691	1257	434	774	0.743	0.619	0.681
	7	1752	1298	454	733	0.741	0.639	0.69
	8	1770	1332	438	699	0.752	0.656	0.704
	9	1758	1308	450	723	0.744	0.644	0.694
	10	1793	1330	463	701	0.742	0.655	0.6985

Table 3-2: The evaluation metrics of coconut-tree detection with block configuration of 3 x 3 (with 50 % overlapping). The proportion of training samples for this performance is 550 positive images to 19,500 negative images

norm	n-bins	Detected object	TP	FP	FN	Precision	Recall	Accuracy
L2-norm	6	2089	1605	484	426	0.768	0.79	0.779
	7	1973	1561	412	470	0.791	0.769	0.78
	8	1995	1572	423	459	0.788	0.774	0.781
	9	2069	1620	449	411	**0.789**	0.798	**0.7935**
	10	2080	1611	469	420	0.774	0.793	**0.7835**
L1-sqrt	6	1792	1388	404	643	0.774	0.683	0.7285
	7	1791	1391	400	640	0.777	0.685	0.731
	8	1808	1414	394	617	0.782	0.696	0.739
	9	1844	1432	412	599	0.777	0.705	0.741
	10	1845	1437	408	594	0.779	0.707	0.743

But if the image is zoomed to that location, a tree stump can be seen, proving that there was previously a coconut tree. This reflects the gap between acquisition time of the sets of images of more than three years. In this time, many coconut trees had been felled and replaced with buildings, roads and small swimming pools. In such cases, bounding boxes kept their classification as correct detections or true positives (TP).

This process took computation time of around two hours using pure Python language. This running time was too slow and not efficient for the object detection task. The researchers then optimized the Python code by using Cython to improve the speed of computation and managed to reach around two minutes for detection, 60 times faster than the performance using pure Python.

3.2 Discussion

The study evaluated the performance of detection with various HOG parameter settings and different proportions of training data. In this study, the performance of two block models of HOG was compared. The first model is window image with 2 x 2 blocks (without overlapping) and the second is window image with 3 x 3 blocks (with 50% block overlapping).

Tables 3-1, 3-2, 3-3 and 3-4 show the effects of various HOG parameter

settings, in terms of precision, recall and accuracy of detection.

For the results in Table 3-1 and Table 3-2, the SVM model used training sample images with a proportion of 550 positive images to 19,500 negative images. From these results it can be seen that, in general, the HOG descriptor with block model of 3 x 3 significantly outperforms the 2 x 2 block model. The results show that using 9 orientation bins gives the highest precision and overall accuracy, followed by 10 orientation bins. Increasing the number of orientation bins slightly improves the accuracy, by approximately 0.01. The best performance based on precision, recall, and overall accuracy was achieved by the configuration of L2-norm normalization, 9-orientation bins and 3 x 3 blocks with overlapping.

The researchers also sought to evaluate performance in detection of coconut trees by increasing the number of positive images for training the SVM. The proportion of training sample images used for the results shown in Tables 3-3 and 3-4 was 850 positive images to 19,500 negative images.

Table 3-3**:** The evaluation metrics of coconut trees detection with block configuration of 2 x 2 (without overlapping). The proportion of training samples for this performance is 850 positive images to 19,500 negative images.

Norm	n-bins	Detected object	TP	FP	FN	Precision	Recall	Accuracy
L2-norm	6	2563	1634	929	397	0.637	0.804	0.7205
	7	2644	1678	966	353	0.635	0.826	0.7305
	8	2616	1670	946	361	0.638	0.822	0.73
	9	2629	1687	942	344	**0.642**	0.831	**0.7365**
	10	2629	1677	952	354	0.638	0.826	**0.732**
L1-sqrt	6	2114	1404	710	627	0.664	0.691	0.6775
	7	2178	1433	745	598	0.658	0.706	0.682
	8	2191	1461	730	570	0.669	0.719	0.694
	9	2306	1521	785	510	0.66	0.749	0.7045
	10	2309	1525	784	506	0.66	0.751	0.7055

Table 3-4**:** The evaluation metrics of coconut trees detection with the block configuration of 3x3 block (with 50 % overlapping). The proportion of training sample for this performance is 850 positive images and 19500 negative images

Norm	n-bins	Detected object	TP	FP	FN	Precision	Recall	Accuracy
L2-norm	6	2382	1730	652	301	0.726	0.852	0.789
	7	2426	1751	675	280	0.722	0.862	0.792
	8	2408	1762	646	269	0.732	0.867	0.7995
	9	2437	1777	660	254	**0.73**	**0.875**	**0.8025**
	10	2546	1820	726	211	0.715	0.896	**0.8055**
L1-sqrt	6	2193	1553	640	478	0.708	0.765	0.7365
	7	2184	1552	632	479	0.71	0.764	0.737
	8	2190	1577	613	454	0.72	0.776	0.748
	9	2236	1603	633	428	0.717	0.79	0.7535
	10	2280	1620	660	411	0.71	0.798	0.754

Generally, the accuracy of detection is slightly increased following the increase in the number of positive images, by approximately 0.01. Overall, the recall value in this experiment was increased significantly compared to Tables 3-1 and 3-2. The highest recall value reached 0.896 for 10-orientation bins but the precision of 9-bins outperformed 10-bins. From the two different proportions of training samples, it can be summarized that by increasing the number of positive images the recall value will increase but the precision decreases. This is due to the number of detected objects which are affected by increasing the number of positive images not being comparable with the number of true positives. The highest recall value was improved by 10%, from 0.793 to 0.896. It is also noted that using L2-norm as block normalization resulted in better accuracy than L1-sqrt normalization.

4 CONCLUSIONS

The study concludes that HOG and linear SVM techniques could be used to detect coconut trees in Pleiades satellite imagery. It is found that the best HOG parameter settings for window size of 16 x 16 pixels is a combination of block model of 3 x 3 with 50% overlapping, 9 orientation bins, and L2-norm block normalization. These configurations are proven to have an overall accuracy of 80% with precision of 73% and recall of 87%.

ACKNOWLEDGMENTS

This study was funded by the Risetpro programme, Ministry of Research, Technology and Higher Education of Indonesia.

REFERENCES

B.Rangda M, Hanchate DB (2014) Animal Detection Using Histogram Oriented Gradient. Int J Recent Innov Trends Comput Commun 2:178–183

Baghdadi A, Latif SA (2015) Enhancement of Fast Pedestrian Detection Using HOG. Int J Ind Electron Electr Eng 3:18–22

Bernales AMJ, Samonte CO, Antolihao JAF, et al (2016) Integration of Template Matching and SVM Technique for Coconut Tree Detection. In: 37th Asian Conference of Remote Sensing. Asian Association on Remote Sensing

Blaschke T (2010) Object based image analysis for remote sensing. ISPRS J Photogramm Remote Sens 65:2–16. doi: 10.1016/j.isprsjprs.2009.06.004

Corvee E, Bremond F (2010) Body parts detection for people tracking using trees of Histogram of Oriented Gradient descriptors. Proc - IEEE Int Conf Adv Video Signal Based Surveillance, AVSS 2010 469–475. doi: 10.1109/AVSS.2010.51

Dadi HS, Mohan Pillutla GK (2016) Improved Face Recognition Rate Using HOG Features and SVM Classifier. IOSR J Electron Commun Eng 11:34–44. doi: 10.9790/2834-1104013444

Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. Proc - 2005 IEEE Comput Soc Conf Comput Vis Pattern Recognition, CVPR 2005 I:886–893. doi: 10.1109/CVPR.2005.177

Dasgupta S (2014) Report of the FAO High Level Expert Consultation on Coconut Sector Development in Asia and the Pasific. Bangkok, Thailand

Kim S, Cho K (2014) Fast calculation of histogram of oriented gradient feature by removing redundancy in overlapping block. J Inf Sci Eng 30:1719–1731

Kobayashi T, Hidaka A, Kurita T (2008) Selection of histograms of oriented gradients features for pedestrian detection. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell

Lect Notes Bioinformatics) 4985 LNCS:598–607. doi: 10.1007/978-3-540-69162-4_62

Li W, Fu H, Yu L, Cracknell A (2017) Deep Learning Based Oil Palm Tree Detection and Counting for High-Resolution Remote Sensing Images. Remote Sens 9:. doi: 10.3390/rs9010022

Liasis G, Stavrou S (2016) Satellite images analysis for shadow detection and building height estimation. ISPRS J Photogramm Remote Sens 119:437–450. doi: 10.1016/j.isprsjprs.2016.07.006

Neubeck A, Van Gool L (2006) Efficient non-maximum suppression. Proc - Int Conf Pattern Recognit 3:850–855. doi: 10.1109/ICPR.2006.479

Savakis A, Sharma R, Kumar M (2014) Efficient Eye Detection using HOG-PCA Descriptor. 9027:1–8. doi: 10.1117/12.2036824

Shafri HZM, Hamdan N, Saripan MI (2011) Semi-automatic detection and counting of oil palm trees from high spatial resolution airborne imagery. Int J Remote Sens 32:2095–2115. doi: 10.1080/01431161003662928

Shahi K, Shafri HZM, Taherzadeh E, et al (2015) A novel spectral index to automatically extract road networks from WorldView-2 satellite imagery. Egypt J Remote Sens Sp Sci 18:27–33. doi: 10.1016/j.ejrs.2014.12.003

Shruthi G (2012) Automatic Detection and Classification of Apple-A Survey. In: Procedding of International Conference on Current Trends in Engineering, Science and Technology

Wójcikowski M (2016) Histogram of oriented gradients with cell average brightness for human detection. Metrol Meas Syst 23:27–36. doi: 10.1515/mms-2016-0012

Yan G, Yu M, Yu Y, Fan L (2016) Real-time vehicle detection using histograms of oriented gradients and AdaBoost classification. Optik (Stuttg) 127:7941–7951. doi: 10.1016/j.ijleo.2016.05.092

Žemgulys J, Raudonis V, Maskeliunas R, Damaševičius R (2018) Recognition of Basketball Referee Signals from Videos Using Histogram of Oriented Gradients (HOG) and Support Vector Machine (SVM). Procedia Comput Sci 130:953–960. doi: 10.1016/j.procs.2018.04.095