Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)
Vol. 7, No. 2, August 2021, pp. 193-206
ISSN: 2338-3070, DOI: 10.26555/jiteki.v7i2.20684

The Detection System of Helipad for Unmanned Aerial Vehicle Landing Using YOLO Algorithm

Bhakti Yudho Suprapto, A. Wahyudin, Hera Hikmarika, Suci Dwijayanti
[1]Department of Electrical Engineering, Universitas Sriwijaya, Ogan Ilir, South Sumatera 30662, Indonesia

ARTICLE INFO

Article history:

Received May 14, 2021
Revised May 26, 2021
Accepted June 07, 2021

Keywords:

Object Detection;
Helipad;
Image Processing;
Unmanned Aerial Vehicle (UAV);
YOLO Algorithm

ABSTRACT

The challenge with using the Unmanned Aerial Vehicle (UAV) is when the UAV makes a landing. This problem can be overcome by developing a landing vision through helipad detection. This helipad detection can make it easier for UAVs to land accurately and precisely by detecting the helipad using a camera. Furthermore, image processing technology is used on the image produced by the camera. You Only Look Once (YOLO) is an image processing algorithm developed to detect objects in real-time. It results from the development of one of the Convolutional Neural Network (CNN) algorithm methods. Therefore, in this study, the YOLO method was used to detect a helipad in real-time. The models used in the YOLO algorithm were Mean-Shift and Tiny YOLO VOC. The Tiny YOLO VOC model performed better than the Mean-Shift method in detecting helipads. The test results obtained a confidence value of 91.1%, and the system processing speed reached 35 frames per second (fps) in bright conditions and 37 fps in dark conditions at an altitude of up to 20 meters.

Bhakti Yudho Suprapto,
Department Electrical Engineering, Universitas Sriwijaya, Jl. Raya Palembang-Prabumulih Km. 32 Inderalaya, Ogan Ilir, South of Sumatera, Indonesia
Email: bhakti@ft.unsri.ac.id

1. INTRODUCTION

Nowadays, the technology of Unmanned Aerial Vehicle (UAV) is developing very rapidly. UAV is used in various fields such as search and rescue, monitoring, firefighting, surveillance, agriculture, and aerial photography [1][2][3]. UAV is divided into fixed wings, rotor wings (or known as multirotor [1]), and a combination of both, which is often called Vertical Take-Off and Landing UAV (VTOL-UAV) [4]. UAVs can either be controlled manually using a remote control or autonomously. To carry out its mission, the UAV needs a landing platform. If the command is manual using the remote control, the proper landing process is not a problem. However, if it is controlled autonomously, the landing platform (the helipad) requires an appropriate recognition process so that the landing can be carried out accurately. This recognition process requires good image processing, but recognizing helipad images in real-time is difficult.

Image processing for object detection is useful for determining the presence of objects as digital image input to be processed with better quality [5]. Many studies have discussed the detection of helipad objects, including using algorithms such as Speeded Up Robust Features (SURF) [6], Scale Invariant Feature Transform (SIFT) [7], Normalized Wavelet Descriptor (NWD) [8], and Edge Distribution Function (EDF) [9]. However, these studies still had weaknesses, such as suboptimal accuracy value, relatively long processing speed, ground surface area, and camera movement in real-time shooting or video. To solve this problem, this study used another algorithm, namely the You Only Look Once (YOLO) algorithm.

YOLO is a development of the Convolutional Neural Network (CNN) algorithm, which is used to detect real-time objects. Yolo has been widely used for object detection because Yolo is known to be fast in the process. Yolo's algorithm is used to detect humans, cars, animals, and others[10][11]. The CNN algorithm has also been used to detect multiple objects [11]. In this study, object detection was considered successful. Object detection carried out on UAVs using the YOLO algorithm is still rarely implemented. Because this algorithm has multiple advantages, including a good level of precision, it performs quicker due to its use of a bounding box during training time and when identifying objects [10]. In this study, the YOLO algorithm is used to detect helipad objects to assist UAVs in making accurate landings. Helipad detection for the landing process has been extensively researched using methods such as deep learning [12][13], CNN [14], RCNN [15]. The other researchers develop other strategies, i.e., The Airborne Vision using the SSD (Single Shot MultiBox Detector) model [16], the Pose estimator, and principal component analysis (PCA) [17], Infrared stereo system with a pan-tilt unit (PTU) [18]. However, at this time, precise helipad detection is needed, and the detection process is fast.

2. RESEARCH METHOD

2.1. You Only Look Once (YOLO) Algorithm

YOLO is a development algorithm that uses neural networks and is used to detect objects in real-time. YOLO is based on simple network architecture, namely convolutional network, where this network implements the entire image during training and testing. Convolutional network optimizes detection by predicting multiple probability classes and bounding boxes for these objects. The working principle of this algorithm is to change the size of the input image, then perform a single convolutional network process. The detection result of this algorithm is limited based on the confidence model. This detection system typically uses a repurpose classifier or localizer in carrying out its mission to detect objects. Systems such as the Deformable Parts Model (DPM) converts it to an object detection approach using a sliding window. The classification process that occurs when carrying out the object detection approach experiences a very slow speed because it is executed repeatedly.

In the use of the YOLO algorithm system, the approach taken is very different from the previous approach. YOLO processing is performed by seeing the image directly and predicting which objects are located effectively while being able to determine the suitability of the objects in the image in which the predictions made with the synthetic neural network are different from those of the Region Convolutional Neural Network (R-CNN) system, which requires thousands of predictions for an image, which makes YOLO several times faster than R-CNN [19].

YOLO has a simple architecture consisting of 24 convolutional layers, 4 max-pooling layers, and 2 fully connected layers. Convolutional layers extract features from images by simplifying and speeding up processing while the fully connected layers predict probability outputs and object coordinates. This architecture is inspired by GoogLeNet, but as opposed to GoogLeNet, YOLO only uses 1×1 reduction layers that are followed by the 3×3 convolutional layers. YOLO architecture can be seen in Fig. 1.

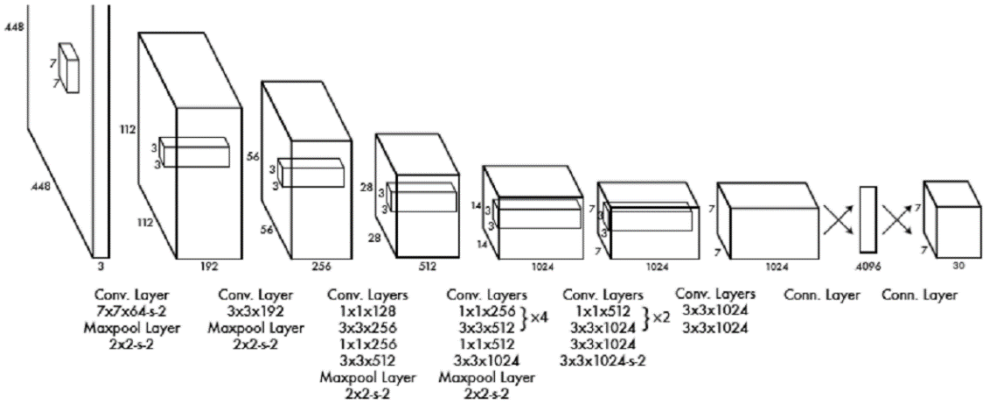


Fig. 1. The architecture of *You Only Look Once* (YOLO) [19]

In Fig. 1, it can be seen that there is a change in convolutional layers 1×1, which serves to reduce the feature space of the previous layers. The YOLO process will intellectually predict objects, namely by providing image input (resized to 224×224), passing through the one-way convolutional network, and exiting the end of the process as a $7 \times 7 \times 30$ tensor that provides bounding boxes information for the network cells. The final step is to calculate the final confidence number for the bounding box and remove values that are lower than 30%. YOLO will provide a frame on the object to be detected where the object detection is a single regression problem. The image pixels are immediately inserted into a separate spatial bounding box, and the probability of the associated class is entered. YOLO classification uses localization where there is an additional assignment of object locations in the form of bounding boxes (bx, by, bh, bw). Each image to be detected will be inserted into the bounding box. The bounding box is a special bounding in the form of a box that has coordinate points and measures the width and height. Grid cells are used to find bounding boxes by dividing the image into many boxes. Each box is used as the center point and is given 3 forms of BBoxes. BBoxes are combined with confidence; this combination is used to predict whether there are objects or not. If an object is predicted, then the BBoxes are given a value of 1 and vice versa. BBoxes with a value of 1 will produce a bounding box. Each bounding box has 5 predictions: x, y, w, h, and confidence, with (x, y) as the center coordinates of the 5 grid cells and (w, h) width and height as the limit of the overall size of the image (object). Confidence is a form of confidence value that convinces the prediction of distant and obscure objects. To detect objects at a grid cell point, an anchor is used. Anchor is an area that will be used as a detection area to assess whether the area has an object or not. The way to assign value to the anchor is to calculate the Intersection over Union (IoU). The Intersection over Union equation can be seen as

$$\text{IOU} = \frac{A \cap B}{A \cup B} \tag{1}$$

Where IoU is calculated by dividing the area where the detection box and ground truth overlap with each of the tops with the total area covered by the detection box and ground truth, as can be seen in Fig. 2.

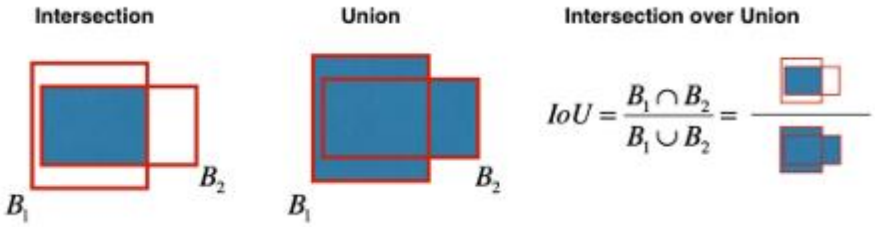


Fig. 2. Illustration of IoU Calculation

Meanwhile, the accuracy or location of the success is usually stated by the accuracy value. Accuracy is a process to find the level of closeness of the system that is read to the true value. Accuracy can be either truth or accuracy. The accuracy test can be calculated by

$$Accuracy = \frac{\sum the\ amount\ of\ correct\ data}{\sum the\ amount\ of\ input\ data} \times 100\% \tag{2}$$

2.2. Mean-Shift Method

The Mean-Shift method is one of the methods that is often used in the process of detecting objects that are carried out continuously to adapt or make adjustments to the color probability distribution, which always changes every time the frames per second (fps) change in the video. This method utilizes the HSV (Hue, Saturation, Value) color filtering system with the weight of the image originating from the target object, which can represent the possibility that the area is the target object. HSV consists of 3 elements: Hue, which represents the color value; Saturation, which represents the color intensity level; and Value, which represents the brightness level of the color.

The Mean-Shift method can be programmed using computer vision in the form of OpenCV and the Python programming language. Testing of the Mean-Shift method in detecting helipad objects was carried out with 10 test data, including 6 tests in bright conditions and 4 tests in dark conditions.

3. RESULTS AND DISCUSSION

The results and discussion will discuss the results of an implementation of the performance of the YOLO algorithm method as a helipad detection system, which includes the stages of data processing, data training, and system testing, as well as analysis of the results of the system.

3.1. Training Data Collection

This study used primary data in the form of helipad image data, as shown in Fig. 3. The helipad measures 1.5 meters on each side. This helipad image data was taken using a drone from the air and used a camera with a flying height ranging from 1–20 meters above ground level. The exposure value on the camera was set to adjust the exposure while taking images. When it was in the bright light conditions during the daytime, the exposure value used on the camera was -2 so that the image was not obscured and the colors on the helipad remained clear. At night, the helipad was illuminated by a spotlight, and the exposure value used on the camera was 0. The image was taken at the Sriwijaya University Football Field, Palembang.

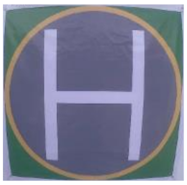
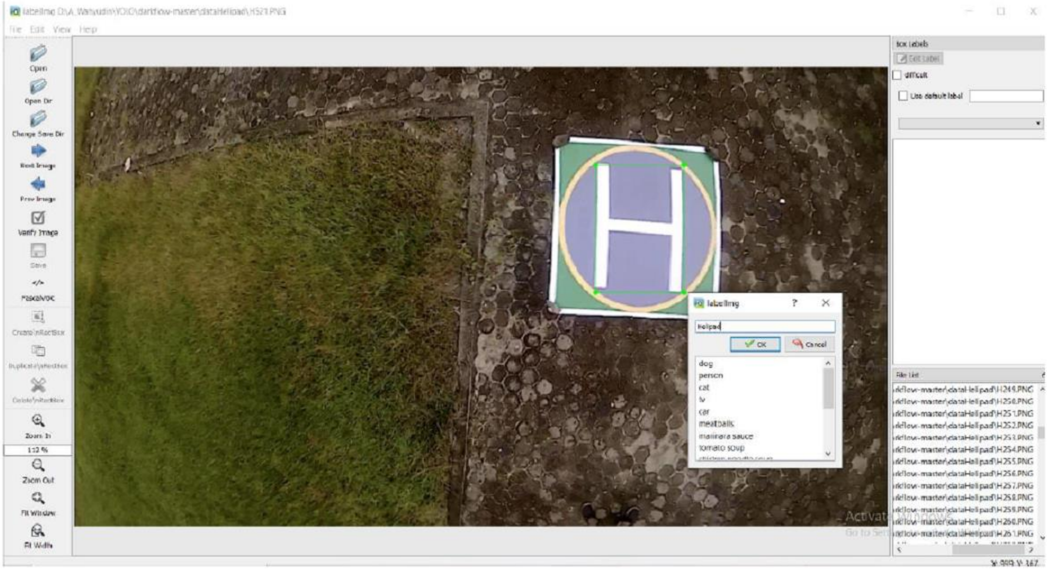


Fig. 3. Helipad Image

A total of 1000 training data images were collected consisting of 250 images with the bright light conditions at an altitude of 1–5 meters, 250 images with the bright light conditions at an altitude of 5–10 meters, 250 images with the bright light conditions at an altitude of 10–20 meters, and 250 images with dark conditions at an altitude of 1–20 meters. The labeling process was carried out with 1 object class, namely the helipad with 1000 data images. To obtain annotations from each training data that had been collected, labeling was carried out on the helipad object using the bounding box method.

3.2. Training Data Processing

The image annotation process was carried out using a labeling application in the Python programming language. The collected training data were processed to obtain annotations from each image. The labeling process using a bounding box can be seen in Fig. 4.



Helipad

dog
person
cat
tv
car
meatballs
marinara sauce
tomato soup

rkflow-master\dataHelipad\H249.PNG
rkflow-master\dataHelipad\H250.PNG
rkflow-master\dataHelipad\H251.PNG
rkflow-master\dataHelipad\H252.PNG
rkflow-master\dataHelipad\H253.PNG
rkflow-master\dataHelipad\H254.PNG
rkflow-master\dataHelipad\H255.PNG
rkflow-master\dataHelipad\H256.PNG
rkflow-master\dataHelipad\H257.PNG
rkflow-master\dataHelipad\H258.PNG
rkflow-master\dataHelipad\H259.PNG
rkflow-master\dataHelipad\H260.PNG
rkflow-master\dataHelipad\H261.PNG

Fig. 4. Training Data Labeling Process

The training data was labeled and named. The labeling results are the coordinates of the annotation process of each bounding box carried out from the data. After conducting the processes of labeling and annotating all images, the training dataset was then carried out. The training process was executed using the Tiny YOLO VOC model. This model was used for training because the GPU memory usage of this model was simpler and less expensive than the other YOLO models. The number of epochs was determined as a ratio to the number of epochs specified, which were 50 and 100 epochs. The accuracy value of success was detected, and the results from using the confidence value as a reference were tested during training.

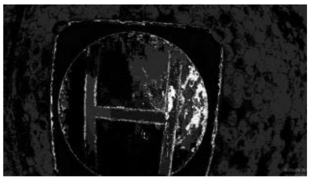
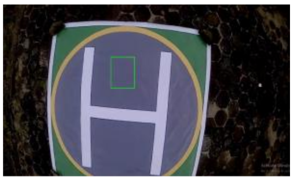
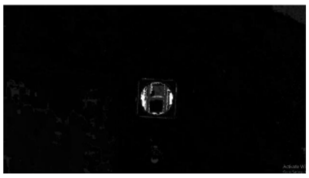
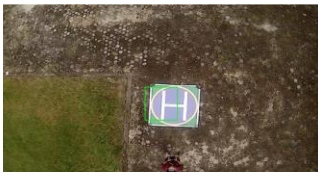
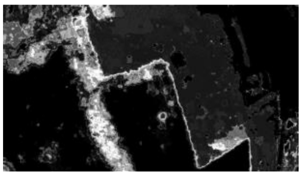
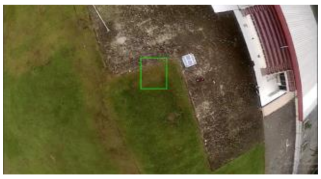
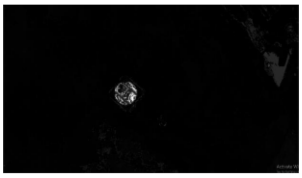
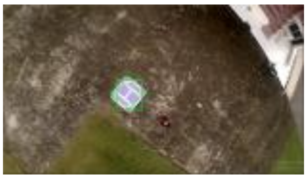
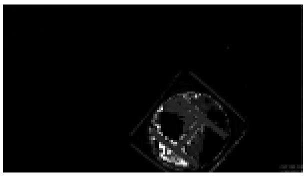
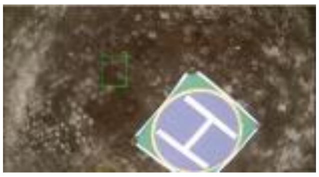
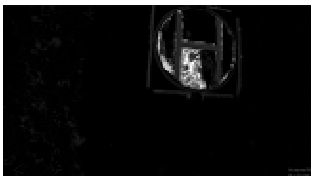
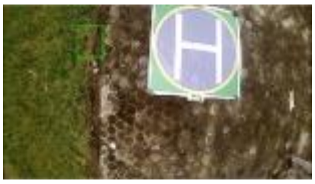
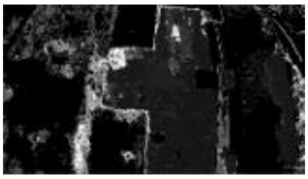
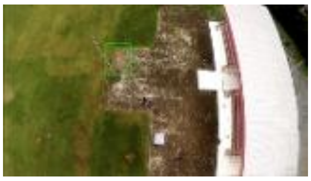
3.3. Testing

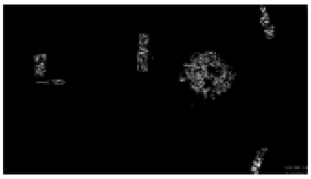
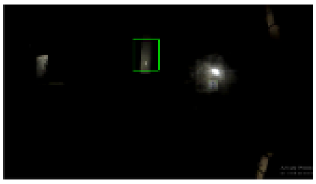
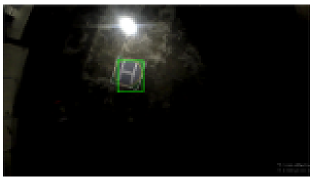
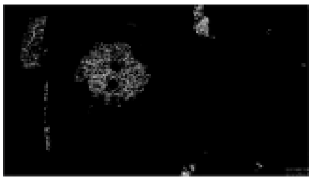
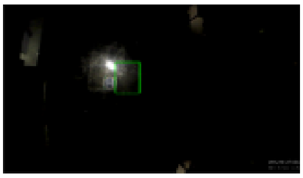
The tests were carried out by detecting the helipad 10 times, including 6 tests in bright conditions and 4 tests in dark conditions, with varying positions and altitudes at the time of detection of the helipad. For the helipad detection process, there were two methods used, namely the Mean-Shift method and the YOLO method with the Tiny YOLO VOC model using the results from training based on Tiny-YOLO-VOC-1c.cfg with a ratio of 50 and 100 epochs.

3.3.1. Testing Using Mean-Shift Method

The Mean-Shift method can be programmed using computer vision in the form of OpenCV and the python programming language. The testing using the Mean-Shift method in detecting helipad objects was carried out with 10 test data, including 4 tests in bright conditions and 6 tests in the dark condition that have been prepared. The results of the tests are shown in Table 1.

Table 1. Testing Using *Mean-Shift* Method

HSV Filter	Output	Result
		Failed
		Failed
		Failed
		Success
		Failed
		Failed
		Failed

HSV Filter	Output	Result
		Failed
		Success
		Failed

Based on the results from Table 1, from 10 tests that had been carried out, the success of the Mean-Shift method system in detecting the helipad could only succeed two times. Because the input frame detected multiple colors around the target object, which were similar to the helipad, the HSV color filtering system could not accurately detect the helipad. Therefore, the use of Mean-Shift method cannot be used for testing the helipad object detection system.

3.3.2. Testing Using Tiny YOLO VOC Model

The Tiny YOLO VOC model is a simplification of the YOLOv2 with several modifications to the YOLO network structure. This Tiny YOLO VOC network structure can be seen in Table 2.

Table 2. Structure of Tiny YOLO VOC

Layer Description	Output Size
input	(416, 416, 3)
conv 3x3p1_1 + bnorm leaky	(416, 416, 16)
maxp 2x2p0_2	(208, 208, 16)
conv 3x3p1_1 + bnorm leaky	(208, 208, 32)
maxp 2x2p0_2	(104, 104, 32)
conv 3x3p1_1 + bnorm leaky	(104, 104, 64)
maxp 2x2p0_2	(52, 52, 64)
conv 3x3p1_1 + bnorm leaky	(52, 52, 128)
maxp 2x2p0_2	(26, 26, 128)
conv 3x3p1_1 + bnorm leaky	(26, 26, 256)
maxp 2x2p0_2	(13, 13, 256)
conv 3x3p1_1 + bnorm leaky	(13, 13, 512)
maxp 2x2p0_2	(13, 13, 512)
conv 3x3p1_1 + bnorm leaky	(13, 13, 1024)
conv 3x3p1_1 + bnorm leaky	(13, 13, 1024)
conv 1x1p0_1 linear	(13, 13, 40)

The network structure of the Tiny YOLO VOC model was used for training. The training was carried out with some epochs, but among the number of epochs, the optimal results were at 50 and 100. The results of the training with an epoch of 50 and 100 were used in system testing. In training with epoch 50, the number of iterations was 6250 with an initial loss value of 103.068634 and ended at a loss of 0.198292837. The loss value and moving average loss data for this training are presented in Fig. 5.

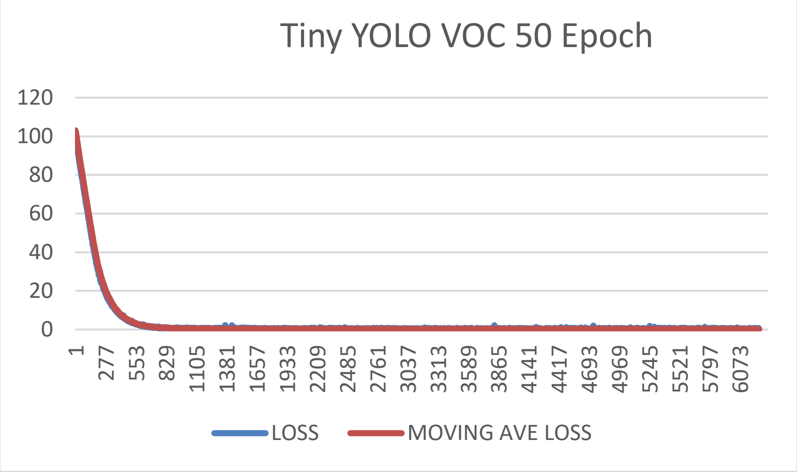
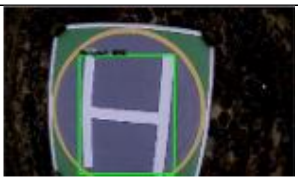
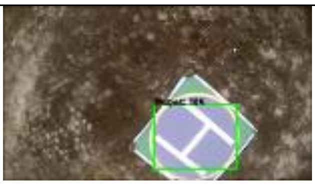
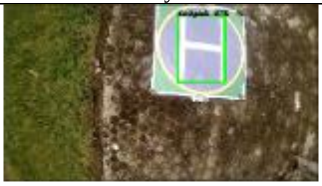
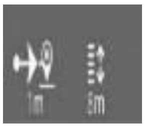
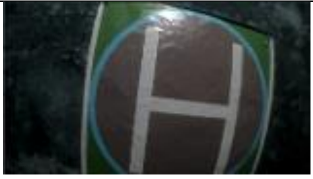
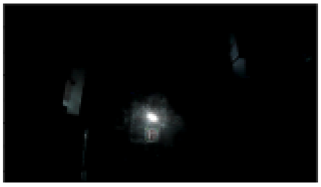
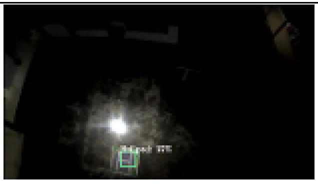
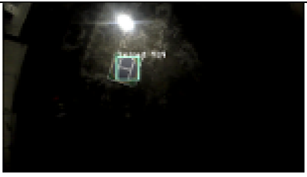


Fig. 5. Graphic of Training Results of Tiny YOLO VOC 50 Epoch

In this case, the training also showed good results because the loss value was already below 1.0, and there was even the smallest loss value reaching 0.009 in the 3580 iterations. To prove the training results, tests were carried out with 10 test images. The test results can be seen in Table 3.

Table 3. System Testing on The Training Results of Tiny YOLO VOC 50 Epoch

Number	Image of Testing Result	Confidence Value	Distance
1	 Successfully Detected	83%	 3 m
2	 Successfully Detected	100%	 9 m
3	 Successfully Detected	100%	 21 m
4	 Successfully Detected	98%	 12 m

Number	Image of Testing Result	Confidence Value	Distance
5	 Successfully Detected	72%	 7 m
6	 Successfully Detected	97%	 8 m
7	 Failed to Detect	-	 2 m
8	 Failed to Detect	-	 17 m
9	 Successfully Detected	57%	 13 m
10	 Successfully Detected	73%	 11 m

Based on the results in Table 3, the level of confidence of the system detecting the helipad was successful. However, some tests failed to detect and had little confidence. From the 10 tests that had been carried out, the average confidence value of the whole test was 74.1%. From the 10 tests, 7 tests were carried out in bright conditions, which resulted in an average high confidence value of 91.67% in 6 tests, whereas 4 tests were carried out in dark conditions, which resulted in an average confidence value of 47.75%, and 2 tests (the seventh and eighth tests) still failed to detect the helipad. After that, on the ninth and tenth tests, the system succeeded in detecting the helipad, but an error still occurred in that the bounding box was not accurate or deviated from the target. All tests that failed to detect the helipad occurred when carrying out the tests in dark conditions.

Furthermore, the dataset training was carried out using the Tiny YOLO VOC model, where there were 100 epochs in total. In this training, the results obtained 12500 iterations. This training process started with an initial loss of 103.068634 and ended with a loss of 0.645778179 in the 12500 iterations. Data loss and moving average loss can be seen in the image presented in Fig. 6.

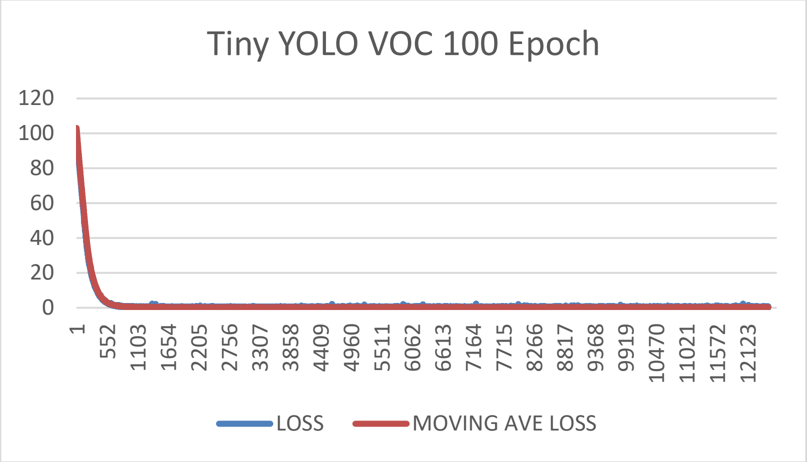
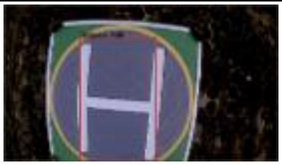
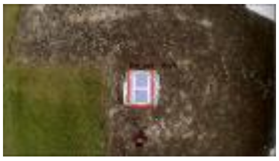
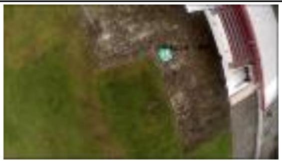
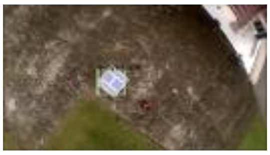
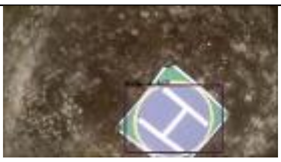
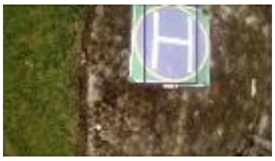
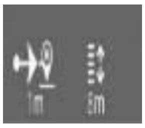
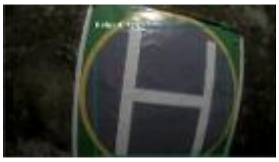
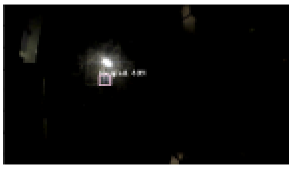
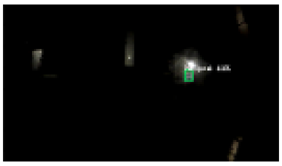
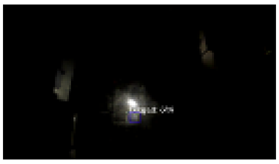


Fig. 6. Graphic of the Training Results of Tiny YOLO VOC 100 Epoch

Based on Fig. 6, the training resulted in a loss value that was already very small or close to zero. The smallest loss value reached 0.003562304 in the 10164 iterations. To see the results of this training, system testing was carried out in real-time with 10 tests. The test results are presented in Table 4.

Table 4. System Testing on the Testing Results of Tiny YOLO VOC 100 Epoch

Number	Image of Testing Result	Confidence Value	Distance
1	 Successfully Detected	82%	 3 m
2	 Successfully Detected	100%	 9 m
3	 Successfully Detected	100%	 21 m
4	 Successfully Detected	99%	 12 m
5	 Successfully Detected	64%	 7 m

Number	Image of Testing Result	Confidence Value	Distance
6	 Successfully Detected	97%	 8 m
7	 Failed to Detect	61%	 2 m
8	 Successfully Detected	40%	 15 m
9	 Successfully Detected	43%	 20 m
10	 Failed to Detect	87 %	 17 m

Based on the test results using the 100 epoch training data presented in Table 4, there is an increase in the confidence value that increases significantly in detecting helipads. However, some tests decreased. Out of 10 tests, there were 6 helipad tests in bright conditions with a high average confidence value of 90.3%. Though 4 helipad tests in dark conditions had an average confidence value of 15.67%, 2 tests successfully detected the helipad in dark conditions.

After testing the performance of the YOLO-based system to detect helipads, a comparison of the success of detecting helipads can be seen in Table 5.

Table 5. The Comparison of Detection Success

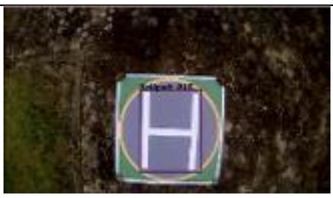
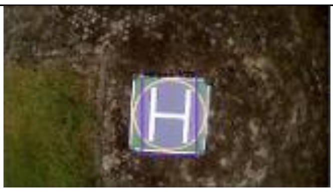
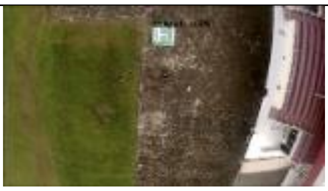
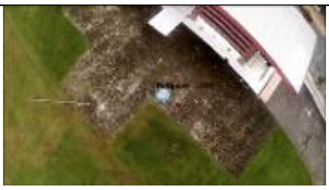
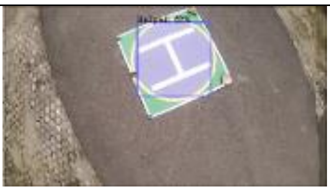
Test Order	**Detection Success**		
	Mean-Shift	**50 *Epoch***	**100 *Epoch***
1	Failed	Success	Success
2	Failed	Success	Success
3	Failed	Success	Success
4	Success	Success	Success
5	Failed	Success	Success
6	Failed	Success	Success
7	Failed	Failed	Failed
8	Failed	Failed	Success
9	Success	Success	Success
10	Failed	Success	Failed

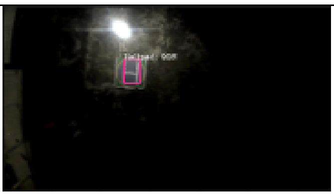
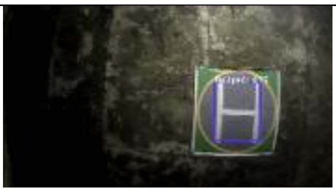
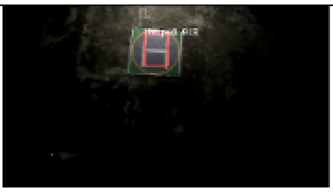
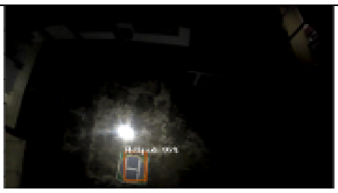
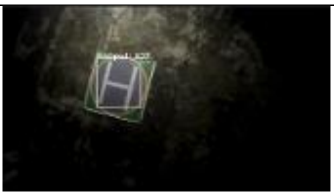
Table 5 shows the success rate in detecting the helipad. In testing using the Mean-Shift method, there were only 2 tests that successfully detected the helipad accurately. During the 50 and 100 epoch training, 2 out of 10 tests failed to detect the helipad, and 2 other tests successfully detected the helipad, but the results were not accurate or deviated from the helipad. This occurred due to the minimum confidence value limit being 20%, whereas in system testing using 100 epoch training results, some tests failed to detect the helipad due to dark conditions at night. In general, the parameters for this 100 epoch training are considered successful because the confidence value was quite high. Therefore, the following tests were carried out using the parameters on the 100 epoch training.

3.3.3. Testing Using Tiny YOLO VOC Model

The system testing was carried out in real-time to detect helipad objects in both bright and dark conditions with varying heights and positions. In this case, the test was based on the training results of a Tiny-YOLO-VOC-based model with 100 epochs that had been carried out previously. The real-time system test detected the helipad object 10 times, which consisted of 5 tests in bright conditions and 5 tests in dark conditions with varying heights and positions. The results is shown in Table 6.

Table 6. Real-Time System Testing

Test	Confidence Value	Status
 Test 1	91%	Successfully Detected
 Test 2	93%	Successfully Detected
 Test 3	100%	Successfully Detected
 Test 4	100%	Successfully Detected
 Test 5	87%	Successfully Detected

 Test 6	93%	Successfully Detected
 Test 7	67%	Successfully Detected
 Test 8	91%	Successfully Detected
 Test 9	96%	Successfully Detected
 Test 10	93%	Successfully Detected

Based on Table 6 in testing the helipad object detection system using the YOLO algorithm, the tests had been successfully carried out. The height of the UAV when testing was approximately 20 meters with a radius of the helipad distance at the camera center point of approximately 10 meters. At this altitude, the prepared UAV descended for landing, and the system performed a helipad detection. The use of the Tiny YOLO VOC model method was able to detect the helipad in both light and dark conditions with a confidence value of 91.1% with a system processing speed of up to 35 fps in bright conditions and 37 fps in dark conditions.

4. CONCLUSION

The performance of the Tiny YOLO VOC model in detecting the helipad showed good results, with system processing speeds reaching 35 fps in bright conditions and 37 fps in dark conditions. In system testing, the YOLO algorithm-based method with the Tiny YOLO VOC model was superior with a success percentage of 91.1% compared to the use of the Mean-Shift method. In the Tiny YOLO VOC model, the most effective parameter was epoch 100.

Acknowledgments

The author would like to thank the Rector of Universitas Sriwijaya for the assistance provided to complete this paper through the 2020 Competitive Grants of Universitas Sriwijaya with the contract number: 0685/UN9/SK.BUK.KP/2020.

REFERENCES

[1] B. Y. Suprapto, M. A. Heryanto, H. Suprijono, J. Muliadi, and B. Kusumoputro, "Design and development of heavy-lift hexacopter for heavy payload," in *Proceedings - 2017 International Seminar on Application for Technology of Information and Communication: Empowering Technology for a Better Human Life, iSemantic 2017*, 2017, vol. 2018-Janua. https://doi.org/10.1109/ISEMANTIC.2017.8251877

[2] S. D'Oleire-Oltmanns, I. Marzolff, K. D. Peter, and J. B. Ries, "Unmanned aerial vehicle (UAV) for monitoring soil erosion in Morocco," *Remote Sens.*, vol. 4, no. 11, pp. 3390–3416, 2012. https://doi.org/10.3390/rs4113390

[3] G. Belloni, M. Feroli, A. Ficola, S. Pagnottelli, and P. Valigi, "A mini UAV for security environmental monitoring and surveillance: Telemetry data analysis," in *Workshop Proceedings of SIMPAR 2008*, 2008, pp. 426–433. http://www.dei.unipd.it/~emg/downloads/SIMPAR08-WorkshopProceedings/MiniAndMicroUAV/Belloni.pdf

[4] B. Y. Suprapto, M. A. Heryanto, H. Suprijono, and B. Kusumoputro, "Altitude control of heavy-lift hexacopter using direct inverse control based on elman recurrent neural network," in *ACM International Conference Proceeding Series*, 2017, vol. Part F1280. https://doi.org/10.1145/3036331.3036354

[5] H. Mulyawan, "Identifikasi dan Tracking Objek Berbasis Image Processing secara Real Time," *EEPIS Final Proj.*, 2011. https://core.ac.uk/download/pdf/12344478.pdf

[6] R. O. Prakash and C. Saravanan, "Autonomous robust helipad detection algorithm using computer vision," in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, 2016, pp. 2599–2604. https://doi.org/10.1109/ICEEOT.2016.7755163

[7] A. Cesetti, E. Frontoni, A. Mancini, P. Zingaretti, and S. Longhi, "Vision-based autonomous navigation and landing of an unmanned aerial vehicle using natural landmarks," in *2009 17th Mediterranean Conference on Control and Automation*, 2009, pp. 910–915. https://doi.org/10.1109/MED.2009.5164661

[8] G. F. Nsogo, K. Kith, B. J. Van Wyk, and M. A. Van Wyk, "Robust helipad detection algorithm (January 2007)," in *AFRICON 2007*, 2007, pp. 1–7. https://doi.org/10.1109/AFRCON.2007.4401634

[9] S. Lee, J. Jang, and K.-R. Baek, "Implementation of vision-based real time helipad detection system," in *2012 12th International Conference on Control, Automation and Systems*, 2012, pp. 191–194. https://ieeexplore.ieee.org/document/6393429

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788. https://doi.org/10.1109/CVPR.2016.91

[11] S. Hossain and D. Lee, "Deep learning-based real-time multiple-object detection and tracking from aerial imagery via a flying robot with GPU-based embedded devices," *Sensors*, vol. 19, no. 15, p. 3371, 2019. https://doi.org/10.3390/s19153371

[12] N. Q. Truong, P. H. Nguyen, S. H. Nam, and K. R. Park, "Deep learning-based super-resolution reconstruction and marker detection for drone landing," *IEEE Access*, vol. 7, pp. 61639–61655, 2019. https://doi.org/10.1109/ACCESS.2019.2915944

[13] L. Yu *et al.*, "Deep learning for vision-based micro aerial vehicle autonomous landing," *Int. J. Micro Air Veh.*, vol. 10, no. 2, pp. 171–185, 2018. https://doi.org/10.1177/1756829318757470

[14] J. Chen, X. Miao, H. Jiang, J. Chen, and X. Liu, "Identification of autonomous landing sign for unmanned aerial vehicle based on faster regions with convolutional neural network," in *2017 Chinese Automation Congress (CAC)*, 2017, pp. 2109–2114. https://doi.org/10.1109/CAC.2017.8243120

[15] J. Wang, D. McKiver, S. Pandit, A. F. Abdelzaher, J. Washington, and W. Chen, "Precision UAV Landing Control Based on Visual Detection," in *2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 2020, pp. 205–208. https://doi.org/10.1109/MIPR49039.2020.00049

[16] J. Li, X. Wang, H. Cui, and Z. Ma, "Research on Detection Technology of Autonomous Landing Based on Airborne Vision," in *IOP Conference Series: Earth and Environmental Science*, 2020, vol. 440, no. 4, p. 42093. https://doi.org/10.1088/1755-1315/440/4/042093

[17] C. Patruno, M. Nitti, E. Stella, and T. D'Orazio, "Helipad detection for accurate UAV pose estimation by means of a visual sensor," *Int. J. Adv. Robot. Syst.*, vol. 14, no. 5, p. 1729881417731083, 2017. https://doi.org/10.1177/1729881417731083

[18] W. Kong, D. Zhang, X. Wang, Z. Xian, and J. Zhang, "Autonomous landing of an UAV with a ground-based actuated infrared stereo vision system," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 2963–2970. https://doi.org/10.1109/IROS.2013.6696776

[19] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, 2015. https://doi.org/10.1109/TPAMI.2015.2437384

BIOGRAPHY OF AUTHORS

Bhakti Yudho Suprapto was born on February 11th 1975 in Palembang (South Sumatra, Indonesia). He is a graduate of the Sriwijaya University of Palembang with a major in electrical engineering. His post-graduate and doctoral program is in Electrical Engineering of Universitas Indonesia (UI). He is an academic staff of Electrical Engineering of Sriwijaya University of Palembang as a lecture. He concerns with major control and intelligent system. He is a member of IEEE. Email: bhakti@ft.unsri.ac.id

A. Wahyudin was born on October 10th 1998, in Palembang (South Sumatra, Indonesia). He was a graduate of the Sriwijaya University of Palembang in the major of Electrical Engineering last year. He is a member of the Computer and Robotic Laboratory in her major. His final project in bachelor's degree was about image processing. He concerns about learning more about robotic and programming. Email: ahmadwahyudin4uditore@gmail.com

Hera hikmarika was born on December 7th 1978 in Palembang (South Sumatra, Indonesia). She is a graduate of the Sriwijaya University of Palembang with a major in electrical engineering. Her master program is in Electrical Engineering of Universitas Gadjah Mada (UGM). She is an academic staff of Electrical Engineering of the Sriwijaya University of Palembang as a lecturer. She concerns with major control and intelligence. Email: herahikmarika@ft.unsri.ac.id

Suci Dwijayanti received the M. S. degree in electrical and computer engineering from Oklahoma State University, Stillwater, OK, the USA, in 2013. Then, in 2018, she received her Doctoral degree from the Graduate School of Natural Science and Technology, Kanazawa University, Japan. Her research interests focus on signal processing and machine learning. She was previously an engineer with ConocoPhillips Indonesia Inc. Ltd from 2007 to 2008. Since 2008, she has been with the department of Electrical Engineering at Sriwijaya University, Indonesia. She is a member of IEEE. Email: suci.dwijayanti@gmail.com