INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2021, VOL. 67, NO. 4, PP. 727-733
Manuscript received February 9, 2021; revised November, 2021 DOI:10.24425/ijet.2021.137869

Identification of Garbage in the River Based on The YOLO Algorithm

Bhakti Yudho Suprapto, Kelvin, Muhammad Arief Kurniawan, Muhammad Kevin Ardela, Hera Hikmarika, Zainal Husin, and Suci Dwijayanti

***Abstract*—This paper discusses the identification of garbage using the YOLO algorithm. In the rivers, it is usually difficult to distinguish between garbage and plants, especially when it is done in real-time and at the time of too much light. Therefore, there is a need of an appropriate method. The HSV and SIFT methods were used as preliminary tests. The tests were quite successful even in close condition, however, there were still many problems faced in using this method since it is only based on pixel and shape readings. Meanwhile, YOLO algorithm was able to identify garbage and water hyacinth even though they were closed to each other.**

***Keywords*—control, identification, HSV and sift method, USV, yolo algorithm**

I. Introduction

THE existence of rivers is crucial for humans and other living things either for irrigation or consumption. Nowadays, the widespread use of plastics has resulted in a lot of plastic garbage polluting rivers of various sizes[1]. In urban areas, garbages and sewages are produced and disposed of by residents directly into the waters, thus, worsening the condition of the rivers as a water resource. Manual cleaning is ineffective because it cannot cover a large work area and equipment that are expensive and require a lot of labor. Therefore, there is a need for a tool that utilizes technology to solve this problem, namely, the Unmanned Surface Vehicle (USV) to collect the garbages[2][3]. However, USV still uses the paths as navigation[2] and still moves manually [3].

In this paper, USV will be controlled automatically by moving down the river that will be cleaned. However, considering that the river is quite wide, there will be some garbage that is missed from the USV. Therefore, it requires an arm to pick up the garbage. The arm will move automatically based on object recognition using the camera as its input source. Recognition of the object becomes an important element in the classification and identification systems of the existing garbage and then this process will generate an instruction for the arm to take the garbages. Therefore, the type of garbages needs to be identified. There have been several methods of identification and classification used in previous studies, such as the SURF-BoW method and Multi-Class SVM[4], Content-Based Image Retrieval (CBIR)[5], K-NN Algorithm[6], and K-means Clustering Algorithm[7]. However, in these studies, there were several obstacles, namely the low accuracy and limited types of garbage that could be identified. To improve the performance, Alex Krizhevsky, Geoff Hinton, and Ilya Sutskever developed Convolutional Neural Networks (CNN)[8]. In its development, CNN Method develops again into several algorithms including Region-based CNN (R-CNN)[9], Fast R-CNN[10], Faster R-CNN[11], and You Only Look Once (YOLO)[12], so that, it has many options in its application. Based on the studies that have been conducted, the CNN method has been used for classifying garbage such as the use of smartphone applications for CNN-based garbage identification [13] and CNN-based garbage classification in a free environment [14]. Based on this basis, the CNN method was chosen to solve this problem.

This paper used the CNN method with the YOLO algorithm in identifying garbage. So, it distinguished between garbage and plants, in this case, water hyacinth (in Indonesian known as enceng gondok). The research was conducted in a river where the most abundant garbage is water hyacinth (*Eichhornia Crassipes*) and bottles made of plastic or metal. This YOLO algorithm works by arranging the frame of the object detected as a single regression from the pixel image to the coordinates of the bounding box [12]. One of the strengths of using the YOLO algorithm is faster object detection. YOLO can detect all images during training and testing so that it implicitly encodes contextual information about the class and good images, low patch errors in detecting images, and eventually, YOLO can generalize the image of objects in the box [12][15]. So with such strength, this method will be used to identify garbage.

In the discussion to solve the problem, this paper was divided into several chapters, as follows; the first chapter contains the problems and objectives of this paper in its introduction, the second chapter discusses the YOLO Algorithm and Hue, Saturation, Value (HSV) method as preliminary object detection, the third chapter discusses the design of USV, the fourth chapter explains the research method, the fifth chapter contains the results of experiment and analysis, and the last chapter contains conclusions.

II. Materials and Methods

A. You Only Look Once (YOLO) Algorithm

YOLO is an algorithm for object detection that is currently popular and commonly developed by researchers. This algorithm uses the entire image of the object to be identified by a single neural network. This principle is different from other object detection algorithms. Such algorithms usually apply the

This research was funded by DIPA of Public Service Agency of Universitas Sriwijaya 2019 (Grant No 149.036/UN9/SB3.LP2M.PT/2019).

Authors are with Department of Electrical Engineering, Faculty of Engineering, Universitas Sriwijaya, Indonesia (e-mail: bhakti@ft.unsri.ac.id, orangawam16@gmail.com, arifkurniawan07200@gmail.com; kevinard710@-gmail.com, hera_aqila@yahoo.com, zaenalhusin@gmail.com; suci.dwijayanti-@ft.unsri.ac.id).

model to the image at several locations as well as scaling on the image. This process will give value to the image as material for object detection. In the YOLO algorithm, this single neural network will divide the image of objects into regions in the form of boxes and then predict bounding boxes and probabilities, for each bounding region box to be weighed probabilities to produce a classification as objects or not.

The YOLO algorithm has a simple architecture, which is a convolutional neural network as shown in Fig. 1.

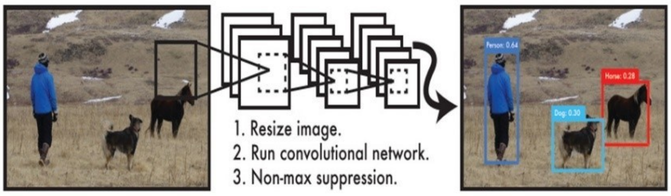


Fig. 1. The Procedures of YOLO Algorithm in Detecting Object [12]

As shown in Fig. 1, this algorithm will predict several *bounding boxes* and class probabilities for the boxes simultaneously. YOLO trains images in full direction and directly optimizes performance [12]. The YOLO algorithm has several steps: (1) resizing the input images to 448 × 448, (2) running a single convolutional network on the images, and (3) limiting detection generated by the confidence of the model.

B. The Procedures of YOLO Algorithm

The first step taken is by taking a sample *classifier* such as VGGNet or *Inception* and converting it into an object detector with a small *sliding window* in the image. At each step, the classifier gets a prediction of what object is in the current window. Using a sliding window gives hundreds or thousands of predictions for the image, but only the most definite stored classifier will be saved. This approach is successful but it will obviously be very slow because it will run the classifier repeatedly. A more efficient approach is to first estimate which part of the image contains interesting information called region proposal and then carry out classifiers only in this region. These classifiers must take fewer steps than sliding windows but still run repeatedly. The YOLO algorithm takes a very different approach. YOLO only sees the image once (because it's called: You Only Look Once) but in a smart way.

The YOLO algorithm divides the image into 13 grids with 13 cells. Then each cell is responsible for predicting 5 bounding boxes. The bounding boxes describe the boxes that enclose the object. The YOLO algorithm also shows a confidence score that gives information on how certain the predicted boxes cover several objects. This score does not say anything about what objects are in the boxes, only if the shape of the boxes are good. The predicted bounding boxes might look like this (the higher the confidence score, the bolder the box). For each bounding box, the cell also predicts class. It functions as a classifier: it gives a probability distribution to all possible classes. The confidence scores for bounding boxes and class predictions are combined into one final score which tells us the probability that these bounding boxes contain a certain type of object. For example, the large, bold yellow box on the left is 85% confident that it contains a "dog" object. Because there are 13 × 13 = 169 grid cells and each cell predicts 5 bounding boxes, we ended up with a total of 845 bounding boxes. It turned out that most of these boxes had very low confidence scores, so we only keep boxes with a final score of 30% or higher. From a total of 845 bounding boxes, we only kept these three boxes because they provided the best results. However, note that although there were 845 separate predictions, everything was made at the same time - the neural network only ran once. Fig. 2 shows the process that occurred on the YOLO network.

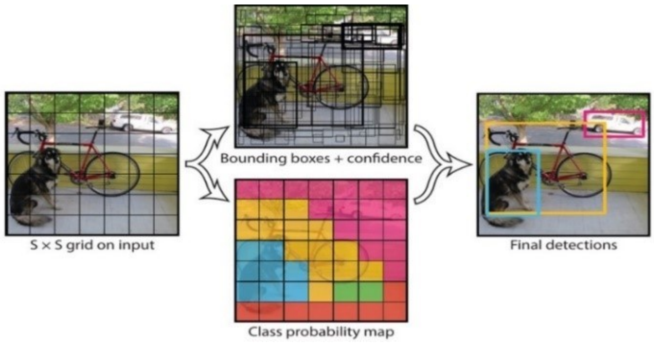


Fig. 2. YOLO Model [12]

Our system detected the model as a regression problem. This divided the image into an S × S grid and for each grid cell predicted a bounding box B, confidence for that box, and class C probability. This prediction was encoded as the tensor S × S × (B ∗ 5 + C).

C. Hue, Saturation, Value (HSV) Method

The systems that use the Hue, Saturation, Value (HSV) method use color space. *Hue* states the actual colors, such as red, blue, yellow, and so on. *Saturation* is a value to replace the purity or strength of color. While *Value* means that the brightness of the color of which value ranges from 0-100%. So, if there is a value of 0, the color will be black. So the greater the value, the brighter the new variations, as shown in Fig. 3 below:

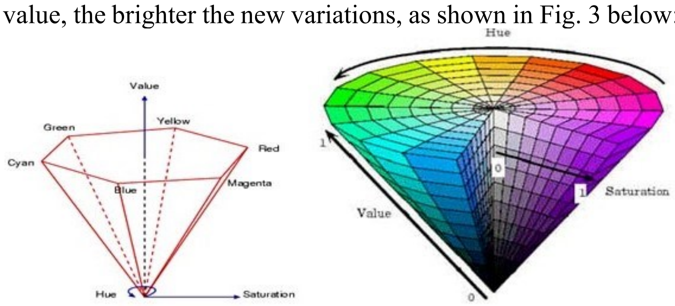


Fig. 3. Triangle of HSV Color

The following equation is from Travis' Theory that will convert the RGB value into the value of HSV Function:

$$r = \frac{R}{R+G+B} \quad , \; g = \frac{G}{R+G+B} \quad , \; b = \frac{B}{R+G+B} \qquad (1)$$

$$H = \begin{cases} \frac{60*(g-v)}{s*v}, \; \substack{0, V=0 \\ V=r} \\ 60 * \left[2 + \frac{b-r}{s*v}\right], V = g \\ 60 * \left[4 + \frac{r-g}{s*v}\right], V = b \end{cases}$$

$$H = H + 360 \; if \; H < 0 \qquad (2)$$

$$S = \begin{cases} 1 - \frac{min(r,g,b)}{V}, \; \substack{0, V = 0 \\ V > 0} \end{cases}$$

$$V = max\ (r, g, b) \quad (3)$$

D. SIFT Method

Scale Invariant Feature Transform (SIFT) is one of the most popular image registration methods and is widely used. SIFT describes an image by finding points that stand out from the image. From each of the key points, we can get gradient orientation and gradient magnitude which are further processed to be a feature of the key points. Furthermore, a histogram will be constructed from the gradient magnitude based on the associated gradient orientation, the bin on the histogram will have a range of 0°, 45°, 90°, 135°, …, 315° In addition to the gradient magnitude (GM) and gradient orientation, there is also a gradient occurrence (GO) that can be considered for use in image registration.

E. Unmanned Surface Vehicle

Unmanned Surface Vehicle (USV) is an intelligent small equipment platform that has an automatic navigation function, detects targets, recognizes targets, awareness of a context, and so on [16] and is able to move on the surface of water either rivers, lakes or oceans. At present, USV is equipped with sophisticated control, sensor, communication and weapons systems that are focused on military applications [17]. USV is the same as other unmanned vehicles which grow rapidly and has been widely studied by people because of its many advantages. According to [18] USV is widely used for monitoring seas and rivers, and anti-terror activities. In addition, USV is also used for research in oceanography and meteorology [19] and other fields. There have been many researchers conducted on and uses of USV by researchers at this time, however, the use of USV for handling garbage is a relatively new application. Some previous researches were only in the form of prototype design or have not worked autonomously. To collect garbage, the USV uses a conveyor placed at the front of the USV [3]. The utilization of conveyors has limitations in the range of garbage collection that can only take garbage in front of the conveyor. Unmanned Surface Vehicle

III. METHOD

A. USV Design

Unmanned Surface Vehicle (USV) is an important component in this study since USV is used in testing this YOLO algorithm in real-time. The USV that has been designed is shown in Fig. 4 below:

Fig. 4. USV Design

USV moves according to the specified path in which the garbages are collected. This USV movement is autonomous. The parameter used in this system is a compass to make the USV run straight down the path. The path is shown in Fig. 5 below:

Fig. 5. Map of USV Movement

As shown in Fig. 5, the numbers at that key points are in the order of their movements. Starting from home then goes to keypoint 1 to keypoint 2 and ends in keypoint 3. HSV and SIFT Methods

B. HSV and SIFT Methods

The preliminary test on the identification of the garbages was carried out using the HSV and SIFT methods. As already mentioned, the HSV method has several important aspects that must be considered, namely lower HSV, upper HSV, contour, coordinates and object radius. Lower HSV is the smallest range of color values to be detected. While the upper HSV is the largest or darkest color range. The difference between the lower and upper colors is the color range that will be detected by the camera. Contour in an object is a collection of dots that are included in the range of colors that have been set, so that will form an object. The next aspect is the coordinates, in this case the x and y axis coordinates, taken as a reference in retrieving the object. Coordinates will be formed if there are detected objects that are included in the radius. The last aspect is the radius that functions to adjust the size of the object to be detected. If the object to be detected is greater than the radius that has been set, the object is not included in the calculated object as identified object.

C. Dataset Design for the YOLO Algorothm

The garbage and plant images were taken directly (realtime) using a camera and complemented by some images from the internet as a dataset. The process of taking pictures from Google was done by using python-based programming. This was done to increase the variation of the dataset. Then the images that have been collected and selected manually were formatted as .PNG and were named in numbers, such as 000001.PNG. The images that have been formatted were labeled to collect data on the coordinates of the object to be detected in each of the image data provided. This process was done manually using tools that have been coded based on Python and OpenCV. The script program was used to format and label the images.

There were 2 classes of an image used, namely water hyacinth and milo paper boxes. Milo paper boxes were separated into 2 conditions, namely good and deformed. There were 250 sample images prepared for each class and sub-class.

D. Training Network Design

The training used the YOLO network circuit in the modified .cfg format by replacing the number of class = 2 and the number of filters as follows:

(number of classes + 5)*5 = filter (4)

The training process continued until the *Moving Average Loss* value decreased to below 1.0. In this study, a 2GB *Graphics Processing Unit* (GPU) was used and, based on the results of the experiment, the batch size used was batch 4 to avoid the GPU overloading which could cause errors. The training ran at 7 seconds per iteration in 100 epoch.

IV. RESULT AND DISCUSSION

This section will discuss the results of testing using the HSV and SIFT methods initially using the YOLO algorithm.

A. HSV and SIFT Method Testing

Testing using HSV and SIFT methods is shown in Fig. 6 below.

Fig. 6. The Results of Image Processing using HSV Algorithm for Aqua Bottle Garbage

The test was conducted using Aqua bottle sample which was detected properly by getting a radius threshold range of 123 - 160. The threshold value will be smaller if direct sunlight is highlighted on the aqua bottle which results in a change from the HSV values set previously. In Fig. 6, it can be seen that the object has been detected properly as an aqua bottle and has been mapped in zone B and has a radius range of 154 - 156.

Fig. 7. The Results of Image Processing using HSV Algorithm for Milo Box Garbage

The following test was conducted using the milk box (Milo box) sample as can be seen in Fig. 7. The milk box was detected properly with a radius threshold range of 87-90. The milk box has the same constraints on lighting objects. Milk box will have a larger threshold value when the light directed into the milk box is not excessive. Apart from the two samples, there were also other two samples that have been analyzed, namely cigarette box and milk can, which have the same level of detection accuracy. The following table is about the testing on the water hyacinth sample by using the SIFT algorithm as the main algorithm in determining the similarity between water hyacinth and the real water hyacinth.

TABLE I
SIFT ALGORITHM TEST ON WATER HYACINTH

No	Experiment	*Good Point*	Status
1	1st	7/3	Success
2	2nd	4/3	Success
3	3rd	0/3	Failed
4	4th	0/3	Failed
5	5th	1/3	Failed
6	6th	6/3	Success
7	7th	1/3	Failed
8	8th	0/3	Failed
9	9th	4/3	Success
10	10th	2/3	Failed

The test on water hyacinth was conducted using SIFT Algorithm as shown in Fig. 7, in which water hyacinth was detected with the value of min *features* of 3. If the *match feature* or *good point* equal to 3, the water hyacinth will be detected as a match object. As in the test Table II, it is shown that it has has a good value in the first experiment, which is 7/3, that meets the requirements of the match object and will be generated as in Fig. 8. In experiments 2 to 10, it has less accuracy due to the effect of shooting angle. Apart from the aspect of picture taking that has an effect on the unstable pool of water, it makes the side that is considered match features to be more difficult to calculate which is generated as in the 3rd and 8th experiments that make the good point to 0/3.

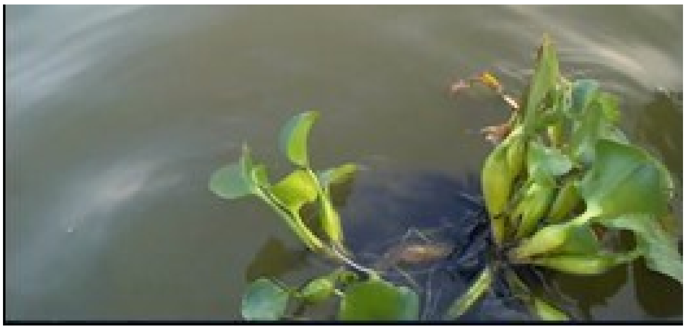
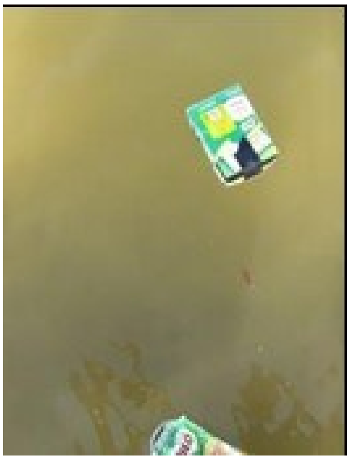


Fig. 8. The Results of Image Processing using SIFT Algorithm for Water Hyacinth

B. *YOLO Training Network*

The training was carried out using 250 data of water hyacinth from 250 images as shown in Fig. 9a and 432 data of milo box from 300 images as shown in Fig. 9b along with the data labeling of each image, *batch size* of 4, and *learning rate* of 1 x 10^{-05}. The device used was a 2GB GPU. The data used were taken directly using a phone camera which was then resized to a resolution of 480x640 and labeled.

(a)



(b)

Fig. 9. Image of Data Training Image. Water Hyacinth (a) Milo Box (b)

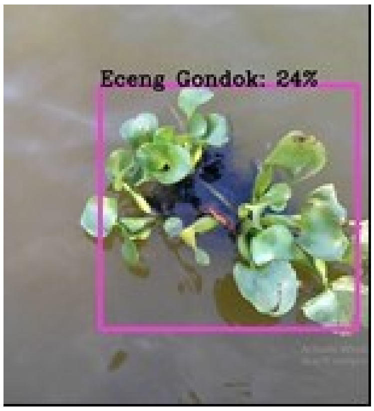
The training was carried out for 38514 iterations or 2000 epochs and generated the *Loss* and *Moving Average Loss* as shown in Fig. 10 with the blue line showing the Loss value and the orange line showing the Moving Average Loss. The data shown on the horizontal axis shows the number of steps and the vertical axis shows the resulting loss value.

The training was carried out using the overfitting method where the *training* was carried out in stages by training a small amount of data (5 data) first to get a very small *moving average loss* (MAV) value (below 1.0). Then, gradually added the training data in twice amount from the first training data until all the data has been trained. In this training, the MAV value was successfully obtained with a value of NaN. This NaN value indicates that the value of the resulting loss is very small to near zero which occurs starting at the 35430th data step/iteration.

Fig. 10. The Graphics of Loss and Moving Average Loss

C. *Testing on YOLO Training Network*

This chapter will discuss the results of training tests using a webcam. This test was intended to detect milo box and water hyacinth. The test results obtained is shown in Fig. 11, and Fig. 12 below:



(a)

(b)

Fig. 11. Detecting Testing. Water Hyacinth (a) Milo Box

Fig. 12. Detecting Testing on Milo Box and Water Hyacinth

The data from the water hyacinth detection experiment results are shown in Fig. 11a and are shown in Table II**.** The data from the milo box detection experiment results are shown in Fig. 11b and are shown in the Table III.

TABLE II
THE TESTING FOR WATER HYACINTH DETECTION

No.	Experiment	*Confidence* Score
1	1^{st} Experiment	24
2	2^{nd} Experiment	12
3	3^{rd} Experiment	0
4	4^{th} Experiment	30
5	5^{th} Experiment	11
6	6^{th} Experiment	16
7	7^{th} Experiment	22
8	8^{th} Experiment	11
9	9^{th} Experiment	0
10	10^{th} Experiment	44
11	11^{th} Experiment	0
12	12^{th} Experiment	10
13	13^{th} Experiment	23
14	14^{th} Experiment	14
15	15^{th} Experiment	15
16	16^{th} Experiment	0
17	17^{th} Experiment	23
18	18^{th} Experiment	0
19	19^{th} Experiment	17
20	20^{th} Experiment	13
Percentage of Detection Success		75%

TABLE III
THE TESTING FOR MILO BOX DETECTION

No.	Experiment	*Confidence* Score
1	1^{st} Experiment	53
2	2^{nd} Experiment	11
3	3^{rd} Experiment	65
4	4^{th} Experiment	92
5	5^{th} Experiment	13
6	6^{th} Experiment	83
7	7^{th} Experiment	0
8	8^{th} Experiment	29
9	9^{th} Experiment	31
10	10^{th} Experiment	36
11	11^{th} Experiment	54
12	12^{th} Experiment	77
13	13^{th} Experiment	81
14	14^{th} Experiment	62
15	15^{th} Experiment	57
16	16^{th} Experiment	0
17	17^{th} Experiment	31
18	18^{th} Experiment	83
19	19^{th} Experiment	17
20	20^{th} Experiment	91
Percentage of Detection Success		90%

TABLE IV
THE TESTING FOR WATER HYACINTH AND MILO BOX

No.	Experiment	*Confidence* Score of Milo	*Confidence* Score of Water Hyacinth
1	1^{st} Experiment	53	11
2	2^{nd} Experiment	77	33
3	3^{rd} Experiment	89	11
4	4^{th} Experiment	16	0
5	5^{th} Experiment	45	0
6	6^{th} Experiment	16	49
7	7^{th} Experiment	93	14
8	8^{th} Experiment	29	0
9	9^{th} Experiment	41	12
10	10^{th} Experiment	16	14
11	11^{th} Experiment	54	19
12	12^{th} Experiment	0	39
13	13^{th} Experiment	81	12
14	14^{th} Experiment	77	14
15	15^{th} Experiment	48	0
16	16^{th} Experiment	92	49
17	17^{th} Experiment	31	0
18	18^{th} Experiment	46	11
19	19^{th} Experiment	17	0
20	20^{th} Experiment	91	13
Percentage of Detection Success		65%	

The data from the milo box and water hyacinth detection experiment results are shown in Fig. 12 and are shown in the Table IV. Based on the results of the experiments shown in table 2, the percentage of water hyacinth detection success is 75%, while in table 3, the percentage of milo box detection success is 90%, and in table 4. the percentage of detection success is 65%. The varied confidence scores shown in the experimental data could be caused by differences in environmental conditions when conducting the experiments in which the images were used as training data, such as differences in lighting, water waves, and test objects that were different from training objects. The success rate of the detection of water hyacinth is lower than which of the detection of milo box due to the training data used for milo box (432 data) are more than those of water hyacinth (250 data). This CNN-based classification system is also able to detect garbage that has a similar color that is captured by the camera at the same frame as shown in the test results in Table 4 in which the detection success is 65%. This percentage is smaller than other experiments because it used fewer training data and was combined with other training data. The testing of YOLO network was able to detect water hyacinth and milo box separately or simultaneously in real-time with 6.5 - 7.0 Frame Per Second (FPS). This can also be due to the training has not managed to get a Moving Average Loss below 1.0. There have been several attempts made to improve the accuracy and Frame Per Second (FPS) of the detector by optimizing data sets with lower resolution photos and optimizing GPU memory capabilities, from 0.8 FPS to 6.9 FPS. In the results of this identification, the YOLO algorithm was quite successful in distinguishing between Milo Box and water hyacinth.

CONCLUSION

YOLO algorithm has succeeded in identifying and differentiating between Milo boxes and water hyacinth even in close proximity. YOLO algorithm also generated the best detection results on objects by using the most training data, namely milo boxes, by 90%. The preliminary test conducted using the HSV and SIFT methods was also successful. However, this method is still limited to lighting.

REFERENCES

[1] J. Gasperi, R. Dris, T. Bonin, V. Rocher, and B. Tassin. "Assessment of floating plastic debris in surface water along the Seine River," *Environ*. Pollut, 2014;195, p. 163–166.

[2] C. Su, W. Dongxing, L. Tiansong, R. Weichong, and Z. Yachao. "An autonomous ship for cleaning the garbage floating on a lake," In: 2009 *Second International Conference on Intelligent Computation Technology and Automation*, 2009, vol. 3, p. 471–474.

[3] A. S. A. Kader, M. K. M. Saleh, M. R. Jalal, O. O. Sulaiman, and W. N. W. Shamsuri. "Design of Rubbish Collecting System for Inland Waterways," *J. Transp. Syst. Eng*, 2015; 2, no. 2, p. 1–13.

[4] Y. Liu, K.-C. Fung, W. Ding, H. Guo, T. Qu, and C. Xiao. "Novel Smart Waste Sorting System based on Image Processing Algorithms: SURF-BoW and Multi-class SVM. Comput," *Inf. Sci.* 2018; 11, no. 3, p. 35.

[5] B. M. Chinnathurai, R. Sivakumar, S. Sadagopan, and J. M. Conrad. "Design and implementation of a semi-autonomous waste segregation robot," in *SoutheastCon* 2016. p. 1–6.

[6] A. Torres-García, O. Rodea-Aragón, O. Longoria-Gandara, F. Sánchez-García, and L. E. González-Jiménez. "Intelligent waste separator," *Comput. y Sist.* 2015; 19, no. 3. p. 487–500.

[7] T. P. Deepa and S. Roka, "Estimation of garbage coverage area in water terrain," in *2017, International Conference On Smart Technologies For Smart Nation (SmartTechCon)*, 2017. p. 347–352.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*. 2012, p. 1097–1105.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, „Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2014, p. 580–587.

[10] R. Girshick. "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*. 2015, p. 1440–1448.

[11] S. Ren, K. He, R. Girshick, and J. Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*. 2015, p. 91–99.

[12] 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, p. 779–788.

[13] G. Mittal, K. B. Yagnik, M. Garg, and N. C. Krishnan. "Spotgarbage: smartphone app to detect garbage using deep learning," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 2016, p. 940–945.

[14] M. S. Rad *et al.* "A computer vision system to localize and classify wastes on the streets," in *International Conference on Computer Vision Systems*. 2017, p. 195–204.

[15] J. Redmon and A. Farhadi."YOLO9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, p. 7263–7271.

[16] D. Mu, Y. Zhao, G. Wang, Y. Fan, and Y. Bai. "USV model identification and course control," in *2016 Sixth International Conference on Information Science and Technology (ICIST).* 2016, p. 263–267.

[17] Q. Zhu. "Design of control system of USV based on double propellers," in *IEEE Reg. 10 Annu*. Int. Conf. Proceedings/TENCON. 2013.

[18] R. Yan, S. Pang, H. Sun, and Y. Pang. "Development and missions of unmanned surface vehicle," J. Mar. Sci. Appl. 2010; 9, no. 4, p. 451–457.

[19] P. H. Heins, B. L. Jones, and D. J. Taunton. "Design and validation of an unmanned surface vehicle simulation model," Appl. Math. Model.2017; 48, p. 749–774.