Journal of Network Intelligence
Taiwan Ubiquitous Information

Volume **8**, Number **2**, May **2023**

Obstacle Detection Method Based on Non-iterative K-means Algorithm

Hanchun Hu[1], Rong Su[1], Zhendong Zhang[1*], Yiheng Wang[1], Zongjun Yin[1*], Khalid Yahya[2]

[1] School of Mechanical Engineering, Anhui Institute of Information Technology, Wuhu 241100, China
[2] Department of Electrical and Electronics Engineering, Nisantasi University, Istanbul 34398, Turkey
Hanchun Hu: 1045073114@qq.com, Rong Su: yinzongjunyzj@163.com
Yiheng Wang: 1315813724@qq.com, Khalid Yahya: Khalid.omy@gmail.com

* Corresponding author: Zhendong Zhang: 444040730@qq.com , Zongjun Yin: 1976225962@qq.com
Received December 2, 2022, revised February 5, 2023, accepted March 24, 2023.

Abstract. *Obstacle detection is an important research direction in the field of mobile robot avoidance. Traditional binocular recognition mainly uses binocular cameras to obtain the disparity map of each object in the field of view to measure the distance of the object. In this way, the depth information about the obstacle can be well obtained. However, the disparity map does not distinguish obstacles and cannot identify object attributes, which may prevent mobile robots from taking obstacle avoidance behaviors. Therefore, this article proposes a mobile robot obstacle detection method based on the non-iterative K-means algorithm to identify the attributes of objects in the binocular image and complete the obstacle classification detection. This method uses version 3 of You Only Look Once (YOLOv3) algorithm framework to perform fast target detection in the Microsoft Visual Studio 2005 (VS2015) environment, and uses the non-iterative K-means algorithm to classify the detection target into obstacles and non-obstacles, to achieve rapid detection of obstacles. The results show that the average speed of YOLOv3 is about 30ms, almost 1/3 or even 1/4 times that of other types of detection algorithms (involving SSD (Single Shot MultiBox Detector) for 61ms and R-FCN (Region-based fully convolutional network) for 85ms). The non-iterative K-means algorithm can be used to realize the rapid detection of obstacles, while the RCNN (Region-based Convolution Neural Networks), Fast-RCNN, and SSD have wrong identification.*
Keywords: Mobile robot; Detect obstacle; Binocular vision; YOLOv3; Non-iterative K-means algorithm

1. **Introduction.** In recent years, research and application of mobile robots have become more extensive, extending from the military and industrial fields to agriculture, household, service, and security industries. One of the difficulties in mobile robotics applications is detecting obstacles such as fixed pots and tables, random pedestrians or animals, etc. The existing detection methods, such as ultrasonic sensor, millimeter-wave (MMW), and laser radar, has been widely used to detect obstacles. For instance, Bers et al. [1] reported different 3D-imaging radar sensors for military aircraft, which can be used to achieve the real-time processing of the measured range image data and obstacle classification and visualization. An and Wang [2] presented a new radar-based obstacle avoidance method for mobile robots on the basis of the combination of PFM and VFH+ methods. They stated that this method permits the detection of unknown obstacles in real-time when the mobile robot steers toward the target without collisions. Mao et al. [3] proposed a novel method to detect obstacles based on monocular camera and laser radar in the

field of environmental perception for intelligent ground robots, and used the marker-based watershed algorithm to segment obstacles contours. Duan et al. [4] applied the multi-layer laser radar to detect roads and obstacles in which the roads are divided into the drivable areas and undrivable area. Wang et al. [5] presented a systematic scheme for fusing MMW radar and a monocular vision sensor for on-road obstacle detection. They used edge detection to assist in determining the boundary of obstacles based on the adaptive threshold algorithm for image shadows. In order to perceive and avoid obstacles for visually impaired people, Long et al. [6] proposed the unified framework of multiple target detection, recognition, and fusion in which the sensor fusion system comprises a low-power MMW radar and an RGB-Depth (RGB-D) sensor. Zhang et al. [7] also used the MMW radar to detect the position and velocity of the obstacle and also combined the deep learning algorithm of the image to obtain the shape and the class of the obstacle.

However, these above-mentioned radar detection methods can only obtain obstacle distance without the judgment of objects' behavior and types. In response to the above problems, the use of images to identify obstacles has been suggested [8]. At present, obstacle detection based on image recognition is mainly divided into monocular visual detection, binocular vision detection and multi-visual detection. The use of monocular visual detection relies too much on differential features. Therefore, if the difference between obstacles and background is small, then it will drop off. At the same time, traditional algorithms need to be paired multiple times to identify which will reduce accuracy and efficiency [9]. In contrast, binocular vision detection has the following advantages [10]: 1) the camera is a passive measurement tool that will not be interfered with the surrounding environment; 2) it can detect obstacles at a wider viewing angle without scanning; 3) it can get information on one surface instantaneously to further obtains information of the whole area; 4) the camera is cheaper than other sensors (e.g. MMW radar), and the image contains more information. We also have to say that traditional binocular-vision recognition mainly uses binocular-vision cameras to obtain a disparity map of each object to measure their distance of them. Since disparity-map distance measurement does not distinguish between obstacles, it is only a suitable identification for obstacles concentrated in the forward route in the open field. So, this method does not involve the fusion of object property recognition and visual vision. In order to solve this problem, an obstacle detection method for a mobile robot which is based on the non-iterative K-means algorithm is proposed. Here, the non-iterative K-means algorithm has been proven to be feasible in the field of obstacle detection applications. For example, Wu et al. [11] focused on the detection and tracking method of unmanned sailing obstacles on a count of MMW radar. They used the K-means clustering method to segment the millimeter wave radar data and clustered multiple points of the same target. Duan et al. [12] combined the improved light detection and ranging technology (DBSCAN) with K-Means, and proposed an obstacle detection method based on four-line laser radar for the vehicle. They reported that this method achieved obstacle detection in the structural roads while the vehicle is traveling at a speed of 5 20km/h.

Among the existing binocular-vision detection techniques, the techniques developing from R-CNN) [13], Fast Region-based Convolution Neural Networks (Fast-RCNN) [14], Faster Region-based Convolution Neural Networks (Faster-RCNN) [15], SSD [16] to YOLO, the accuracy and speed of target detection have reached a new level. R-CNN is the first algorithm that successfully applies deep learning to target detection. R-CNN realizes target detection technology based on convolution neural network (CNN), linear regression, support vector machine (SVM) and other algorithms [17]. Fast R-CNN algorithm is an improvement of R-CNN algorithm by author Girshick [18]. Although R-CNN has achieved good results, its shortcomings are also obvious. Fast R-CNN also uses VGG-16 network

structure. Compared with R-CNN, the training time is 9 times faster, the test time is 213 times faster, and the accuracy rate is increased from 62% to 66%. After the accumulation of R-CNN and Fast R-CNN, Faster R-CNN has integrated feature extraction, potential extraction, bounding box region and classification into a network, which has greatly improved its comprehensive performance, especially in terms of detection speed [19]. Based on the forward convolution network, SSD generates some fixed-size boundary boxes and the fraction of the target category in the representative box, and finally obtains the final detection result through NMS processing. SSD network can achieve 59FPS speed and mAP74.% results in VOC2007 test set [20]. According to relevant research [21], YOLO network is the most prominent in detecting real-time performance. Some of the earliest study in YOLO network dates back to Redmon et al. [22] who presented YOLO for the first time in 2016. They reported that YOLO network takes the frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities, which outperforms other detection methods regardless of the generalization from natural images to other domains like artwork. In 2017, Redmon et al. [23] introduced the YOLO9000 object detection system that can detect over 9000 object categories in real-time. They stated that YOLO9000 trained by the COCO detection dataset and the ImageNet classification dataset have obtained much progress towards lessening the dataset size gap between detection and classification. Later, Redmon et al. [24] also showed the feature-optimized YOLOv3 algorithm in 2018. They demonstrated that compared to the accurate performance of SSD and RetinaNet, YOLOv3 is three times faster. Recently, Wang et al. [25] Compared the advantages and disadvantages of SPP-NET, Fast R-CNN, YOLO, and SSD. They also designed a GuideNet with transfer reinforcement learning method.

In this study, we show that the object properties in the binocular image can be identified by the target detection method, and the obstacle classification detection can be completed by the non-iterative K-means algorithm. In addition, because the light in the indoor environment has less influence on the camera's photosensitive element, the paper mainly talks about the moving robots' binocular-vision recognition of obstacles in indoor environment.

2. **2. Obstacles detection algorithm and design of hardware.**

2.1. **The network structure of YOLOv3 and Darknet-5.**

On the network architecture of deep learning image processing, YOLOv3 adopts the network structure of Darknet-53. As shown in Figure 1, the network structure contains 53 reel layers, similar to the structure of the Residual Network, which means the shortcut connections are set up between some layers. A detailed introduction to the 53 reel layers can be found in the references.

2.2. **Comparison of mainstream algorithm models.**

YOLOv3 has been improved from previous versions which creates deeper network hierarchy and multi-scale detection, and improves the detection results of mean Average Precision (mAP) and small objects. YOLOv3 performs well when using COCO mAP-50 (COCO denotes Common Objects in Context) as an evaluation indicator. According to the reference [24], the average speed of YOLOv3 shown in Figure 2 is about 30ms, almost 1/3 or even 1/4 times that of other types of detection algorithms, since other algorithm models may use more time such as SSD321 for 61ms and R-FCN for 85ms. Therefore, YOLOv3 detection time is less, and the running process is faster. In addition, the Darknet-53 model is superior to the ResNet-101, ResNet-152, and Darknet-19 in terms of the accuracy of image classification, the combined effect and processing efficiency. YOLOv3 uses the Darknet-53 structure to

	Type	Filters	Size		Output	
	Convolutional	32	3	3	256	256
	Convolutional	64	3	3 / 2	128	128
1	Convolutional	32	1	1		
	Convolutional	64	3	3		
	Residual				128	128
	Convolutional	128	3	3 / 2	64	64
2	Convolutional	64	1	1		
	Convolutional	128	3	3		
	Residual				64	64
	Convolutional	256	3	3 / 2	32	32
8	Convolutional	128	1	1		
	Convolutional	256	3	3		
	Residual				32	32
	Convolutional	512	3	3 / 2	16	16
8	Convolutional	256	1	1		
	Convolutional	512	3	3		
	Residual				16	16
	Convolutional	1024	3	3 / 2	8	8
4	Convolutional	512	1	1		
	Convolutional	1024	3	3		
	Residual				8	8
	Avgpool		Global			
	Connected		1000			
	Softmax					

Figure 1. The network structure of Darknet53 (see reference [24]).

ensure that the frame rate (FPS) is more than 60, and the high frame rate allows for more smoothing images. But the FPS of other models are quite different, for example, the FPS of Darknet-19 is 17, that of ResNet-101 is 53 and that of ResNet-152 is 37 24. Therefore, the Darknet-53 improves the comprehensive performance of YOLOv3 in visual obstacles detection on the basis of meeting real-time performance, including improving floating-point operation (BFLOP/s) and FPS. Therefore, this paper uses the dark-net version of the yolov3 neural network framework and trained dataset file yolov3.weights to run the target detection program in the VS2015 environment.

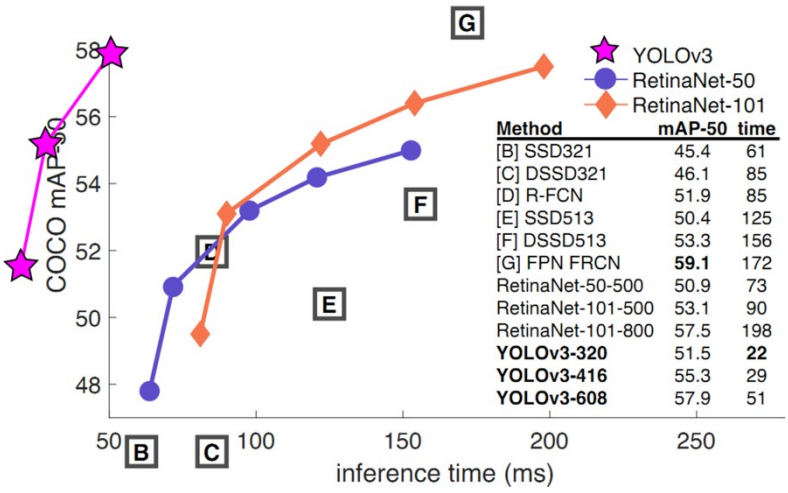

Figure 2. Comparison between YOLOv3 and other models (see reference [24]).

2.3. **Obstacles Classification Algorithm.** After the target detection in the captured image is completed, there is a problem: it is not possible to determine which detection

target is an obstacle. To this end, the present paper is devoted to helping the robot classify the obstacle target by using the non-iterative K-means algorithm. The K-means clustering algorithm is to constantly update the core of cluster C by using iterative method according to the minimized square error E of cluster C, and then judges the sample similarity within the cluster by E value - the smaller E value the higher the similarity of the sample. When the m value iteration becomes stable, the sample cluster task is completed. For the problem of classifying detection targets in this paper, there is no need to iterate to find the core of each classification cluster, since the center point Pcentor obtained by the binocular camera can be regarded as a constant cluster core. In the moving direction of the mobile robot, an object near the Pcentor is an obstacle in front of it. For the target detected which is detected in the image, the problem of determining which detection target constitutes an obstacle is simplified into that of classifying the center point of the detected target's bounding box. Therefore, if the results show that the detection of the center point of the target and the center point of the image have the similar cluster, it can be considered that the detection target is an obstacle in the direction of the moving robot, but not vice versa. In this paper, the detection target classification task will be completed by judging the compact degree between the detection target boundary box center point vn and the image center point Pcentor. Therefore, this paper proposes that the method used to classify each center point is the non-iterative K-means algorithm. The ways of non-iterative K-means classification of data $V = [v_1, v_2, \cdots, v_n]$ sets to be as follows:
1) Initialization, determine the number a of clusters to be classified, and assume the cluster $C_j \neq \emptyset,\ 1 \leq j \leq a$;
2) Select the initial cluster core of each cluster as $b_1, b_2, \cdots, b_a$;
3) Calculate the distance between the sample $v_i\,(1 \leq i \leq n)$ and the initial cluster core of each cluster $b_j\,(1 \leq j \leq a)$; that is, $d_{ij} = \|v_i - b_j\|_2$;
4) According to the minimum distance d_{ij} for v_i, the cluster label is reconsidered as $\lambda_i = \arg\min_{j\in\{1,2,\cdots a\}} d_{ij}$. Then, the sample v_i are classified into corresponding cluster $C_{\lambda i} = C_{\lambda i} \cup \{v_i\}$.

It can be seen that the value of d_{ij} in step 3) can reflect the samples' compactness to the cluster mean. We believe that without iterative process the above algorithm can complete the classification of the image detection target, so as to identify it.

2.4. **Hardware.** The wiring diagram of the hardware is shown in Figure 3(a), and the specific hardware options are as follows.
1) Images are acquired using a clear binocular camera, which is in black and white format.
2) The binocular camera is mounted on a two-degree-of-freedom (2-DOF) cloud platform that works through two steering engines. The boost module should also be installed on the control board of the all-terrain mobile robot to make it work.
3) In this process, we choose the all-terrain mobile robot designed by Wuhu Ample Robotics Research Institute Inc. The combination of camera and cloud platform is set on the robot. Here, the clear binocular camera is connected to the laptop through a USB port. Then, the laptop is placed on the platform of the all-terrain mobile robot. Figures 3(b) and 3(c) show the physical pictures of the binocular camera and the mobile robot.

3. Experimental results and analysis.

3.1. **The results of the target detection.** In this experiment, the YOLOv3 target detection algorithm is used to quickly detect the target in front of the mobile robot in the image. As can be seen from the experimental results in Figure 4 and Table 1, pedestrians in front of the mobile robot are detected, and the recognition rate is over 90%. In addition, the recognition tests of the chair also show good results. The mean

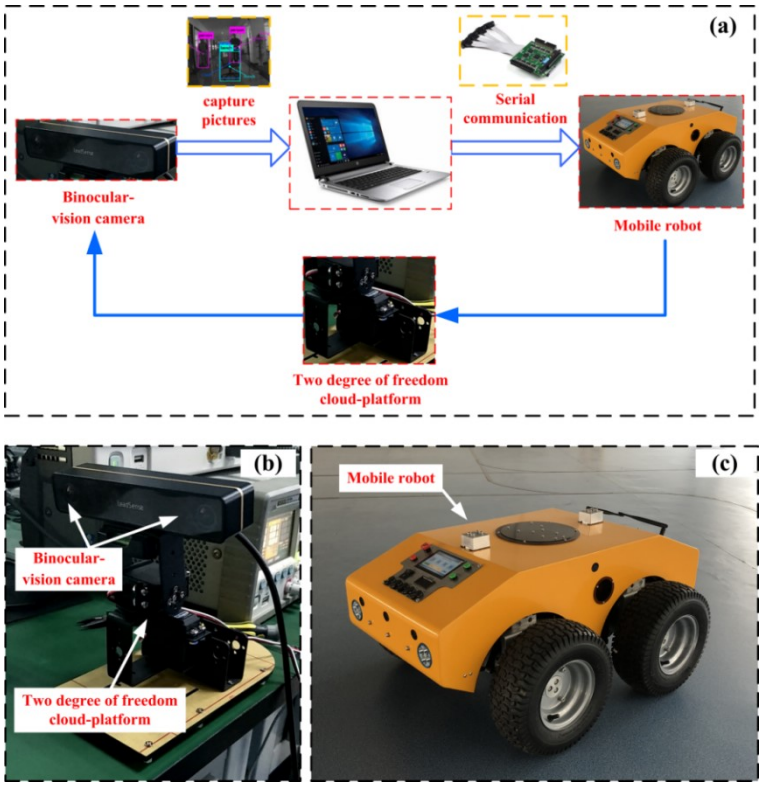

Figure 3. (a) Hardware wiring diagram; (b) image acquisition device; (c) the binocular-vision camera and all-terrain mobile robot.

value of detecting in the left and right images is 64.5%. In Figure 4, the identified objects labeled Person from left to right are pedestrians, pedestrians, robots, but the robot on the far right is mistakenly identified as Person. Therefore, the objects labeled as Person here are numbered as Person1, Person2, and Robot in the order from left to right. As can be

Figure 4. The detection results of the binocular-vision camera.

seen from Figure 4, the positions of the center points of each detection target's bounding box can also indicate that of detection targets. Therefore, the classification of detection targets as obstacles and non-obstacles can be achieved by classifying the central points of each detection target's boundary box. The non-iterative K-means algorithm is used to classify the detection target that can be a barrier into the obstacle category, and for those that does not hinder the movement is classified as a non-obstacle, so as to filter the detection target from the non-obstacle category. The cluster core of the obstacle category in this experiment is the center point of the captured image. Both the left and right

Table 1. Comparative target detection results of left and right images.

Target	Imagery recognition of left camera	Imagery recognition of right camera	Average imagery recognition
Person1	90%	94%	92%
Chair	52%	77%	64.5%
Bench	79%	52%	65.5%
Person2	99%	100%	99.5%
Robot	37%	63%	50%

Table 2. The related parameters of left and right images function of target detection.

Target	Characterization value of left image	Characterization value of right image
Person1	392, 278, 76, 206, 0.9	377, 280, 79, 197, 0.94
Chair	539, 351, 146, 208, 0.52	506, 360, 157, 191, 0.77
Bench	543, 366, 144, 211, 0.79	509, 373, 155, 199, 0.52
Person2	611, 241, 77, 207, 0.99	590, 235, 84, 226, 1.00
Robot	1003, 287, 43, 82, 0.37	994, 287, 51, 83, 0.63

images are 1280px´800px pixels, and the coordinate value of the center point Pcentor is (640, 400).

3.2. **The center point of the bounding box.** Figure 5 is marked with the parameters of the bounding box, each of which actually contains five elements. Its representational function $f(x, y, w, h, c)$ is:

$$f = (x, y, w, h, c)\,. \tag{1}$$

where the first four elements indicate the position and size of the bounding box, and the last value c denotes the confidence level. (x, y) is the position of the upper left corner of the bounding box in the image, and (w, h) is the width and length of the bounding box. The units of x, y, w, and h are the smallest units of pixels in the image. The parameters related to the function of the detection target in this experiment are shown in Table 2.

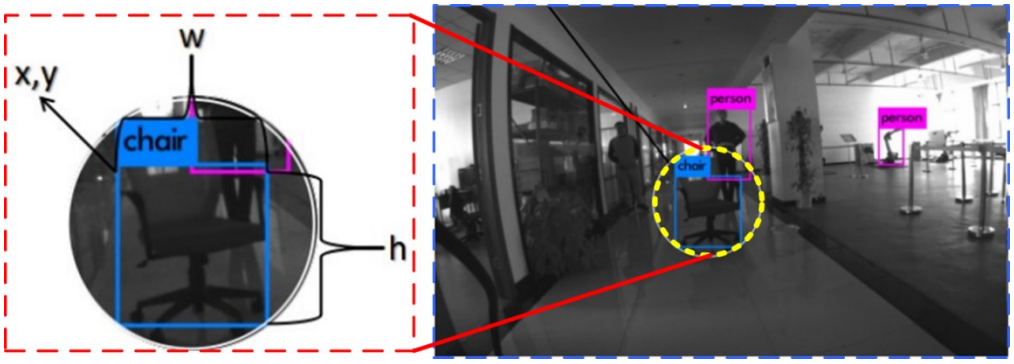

Figure 5. Schematic diagram of boundary box parameters.

The center point of the boundary box of the detection target is shown in Figure 6, and the coordinates (x_{bi}, y_{bi}) of the center point of each boundary box meet the following formula:

$$\begin{cases} x_{bi} = x_i + h_i/2 \\ y_{bi} = y_i + w_i/2 \end{cases}. \tag{2}$$

By calculating the characterization values of left and right images in Table 3, the coordinates of the center point of the boundary box of each detection target $(x_{bi} + h_{bi}/2, y_{bi} + w_{bi}/2)$ can be obtained, as shown in Figure 7 respectively. In addition, the image specification is 1280px*800px pixels, so the coordinates of the center pixel of image Pcentor are (640, 400).

Figure 6. The positions of the center points of each detection target's bounding box.

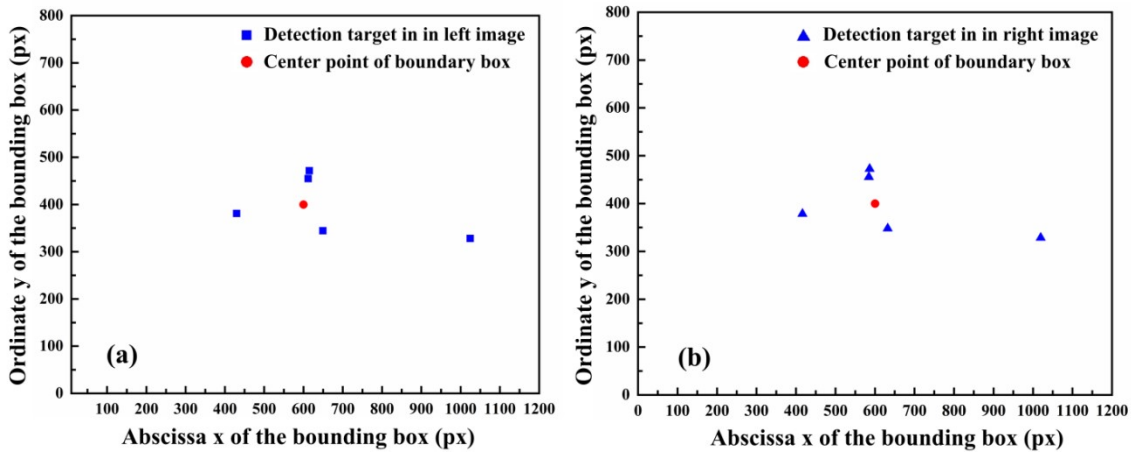

Figure 7. The positions of the center points of each detection target's bounding box.

3.3. **Classification of detection target.** The description of detection target can be divided into two categories, the first category is the center point of the target bounding box which is within the forward direction of the mobile robot, representing the main obstacles. This category is the obstacle target which should be detected, which are closer to the image center point. The second type is the object that is detected during the image recognition. But the object is not within the forward direction of the moving robot which does not belong to the obstacle. This is the interference item of the detection results which should be removed from the detection data.

In our experiment, the center points of the boundary box of the detected object in the left image are named *L-Person1, L-Chair, L-Bench, L-Person2* and *L-Robot* from left to right, while those in the right image called *R-Person1, R-Chair, R-Bench, R-Person2* and *R-Robot*. We divide the center points of the left and right images into two independent sets: V_{left} and V_{right}, as shown below

$$V_{Left} = \{L - Person1, L - Chair, L - Bench, L - Person2, L - Robot, P_{centor}\}, \quad (3)$$

$$V_{Right} = \{R - Person1, R - Chair, R - Bench, R - Person2, R - Robot, P_{centor}\}. \quad (4)$$

Subsequently, we can classify them separately by non iterative k-means.

We get the cluster number $a = 2$ by using the non-iterative K-means classification algorithm; that is, two obstacle target clusters, obstacle C_O and non-obstacle C_N clusters, are chosen. Taking the left image detection results in Figure 7(a) as an example, it is clear that L-Robot is an obvious non-barrier target. The algorithm starts to select two samples $P_{centor} = (640, 400)$ and $L - Robot = (1024.5, 328)$ as the cluster cores, and correspond to obstacle C_O and non-obstacle C_N clusters, respectively. Taking the sample $L - Person1 = (430, 381)$ into the calculation, the the distances between this sample and the core coordinates of the current two clusters can be presented respectively as below:

$$d_1 = \|L - Person1 - P_{centor}\|_2, \quad (5)$$

$$d_2 = \|L - Person1 - L - Robot\|_2. \quad (6)$$

The results are $d_1 = 210.86$ and $d_2 = 596.86$, respectively. Therefore, it will be included in the obstacle cluster C_O. Similarly, after investigating all the samples in the center point set V_{left}, the results are shown in Table 3 below. Similarly, the set V_{right} of detection target center points in the right image is calculated, and the classification results are shown in Table 4. As can be known from Tables 3 and 4, the current cluster in left image is divided into the obstacle cluster C_O containing *L-Person1, L-Chair, L-Bench, L-Person2, Pcentor*, and the non-obstacle cluster C_N involving *L-Robot*, as shown in formula 7; the current cluster in right image is divided into the obstacle cluster C_O including *R-Person1, R-Chair, R-Bench, R-Person2, Pcentor*, and the non-obstacle cluster C_N including *R-Robot*, as shown in formula 8. The formulas 7 and 8 are as follow:

$$\begin{cases} obstacle = \{L - Person1, L - Chair, L - Bench, L - Person2, P_{centor}\} \\ non - obstacle = \{L - Robot\} \end{cases}, \quad (7)$$

$$\begin{cases} obstacle = \{R - Person1, R - Chair, R - Bench, R - Person2, P_{centor}\} \\ non - obstacle = \{R - Robot\} \end{cases}. \quad (8)$$

The center point of the obstacle detection box after clustering in the left and right images is depicted in the plane coordinates, and the results of obstacle detection classification are obtained. As shown in Figure 8, both of the left and right obstacle detection classification pairs are more consistent with the actual situation.

Table 3. The classification results of the left image by using non-iterative K-means algorithm.

Samples	Distance from *L-Person1*	Distance from *L-Robot*	obstacle	non-obstacle
L-Person1	210.86	596.86	✓	
L-Chair	61.72	431.6	✓	
L-Bench	75.74	433.92	✓	
L-Person2	56.31	375.36	✓	
L-Robot	391.18	0.00		✓

Table 4. The classification results of the right image by using non-iterative K-means algorithm.

Samples	Distance from *R-Person1*	Distance from *R-Robot*	obstacle	non-obstacle
R-Person1	224.53	605.07	✓	
R-Chair	78.49	453.16	✓	
R-Bench	90.10	456.32	✓	
R-Person2	52.61	387.99	✓	
R-Robot	386.18	0.00		✓

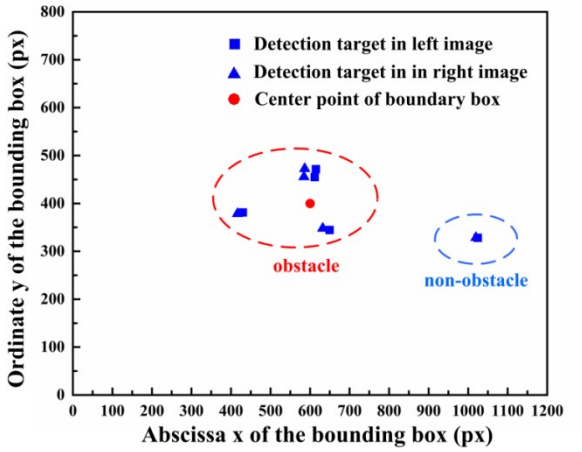

Figure 8. Obstacle detection classification results.

3.4. **Experimental comparison.** The common results of identifying obstacle by using RCNN, Fast-RCNN, and SSD algorithms are shown in Table 5 below. According to Table 5, this method can be used to distinguish robots that are not on the forward path of mobile robots into non-obstacle categories, while the RCNN, Fast-RCNN, and SSD identification methods cannot tell if the identified target is an obstacle in the robot's forward path. It can be seen that the non-iterative K-means algorithm can be used to realize the rapid detection of obstacles.

Table 5. Comparative results of four image recognition methods.

	YOLOv3		**RCNN**		**Fast-RCNN**		**SSD**	
Samples	obstacle	non-obstacle	obstacle	non-obstacle	obstacle	non-obstacle	obstacle	non-obstacle
Person1	✓		✓		✓		✓	
Chair	✓		✓		✓		✓	
Bench	✓		✓		unidentified		✓	
Person2	✓		✓		✓		✓	
Robot		✓	✓		✓		✓	

4. **Conclusion.** In this paper, a mobile robot obstacle detection method based on the non-iterative K-means algorithm is proposed. Firstly, the binocular vision camera is used to capture the images and collect the data. Then, using the dark-net version YOLOv3 neural network framework and the yolov3.weights are used as an image software detection tool to run the image target detection in VS2015 environment. The targets in front of the detected mobile robot are classified into two catalogues: obstacles and non-obstacles. The specific classification method is analyzing the similarity among the center points of each target identification box, of the acquisition image and of the obvious non-obstacle

identification box. Additionally, by using the non-iterative K-means algorithm, the detection targets can be better divided into obstacles CO and non-obstacles CN categories, so as to complete the obstacle detection.

Acknowledgements. This work was supported by the University Natural Science Research Project of Anhui Province under Grant No. KJ2021ZD0144 and No. 2022AH030160, Key project of Collaborative Innovation Fund of Jiujiang District-Anhui Polytechnic University under Grant 2021cyxta1, and Machine Design Manufacture and Automated Professional of Provincial First-class Undergraduate Specialty Construction Project under Grant 2020ylzyjsd02.

REFERENCES

[1] K. Bers, K. Schulz, W. Armbruster, "Laser radar system for obstacle avoidance," *Proceedings of Spie the International Society for Optical Engineering*, vol. 5958, pp. 377-386, 2005.

[2] D. An, H. Wang, "A new laser radar based obstacle avoidance method for intelligent mobile robots," *IEEE Fifth World Congress on Intelligent Control & Automation*, pp. 15-19, 2004.

[3] S. Mao, L. Li, J. Guo, and C. Zhao, "A Novel Obstacle Detection Method Based on Monocular Camera and Laser Radar," *8th IEEE International Symposium on Computational Intelligence and Design*, pp. 12-13, 2015.

[4] J.-M. Duan, K.-H. Zheng, L.-X. Shi, "Road and Obstacle Detection Based on Multi-layer Laser Radar in Driverless Car," *34th IEEE Chinese Control Conference*, pp. 28-30, 2015.

[5] T. Wang, N. Zheng, J. Xin, and Z. Ma, "Integrating Millimeter Wave Radar with a Monocular Vision Sensor for On-Road Obstacle Detection Applications," *Sensors-basel*, vol. 11, no. 9, pp. 8992-9008, 2011.

[6] N.-B. Long, K.-W. Wang, R.-Q. Cheng, W.-J. Hu, and K.-L. Yang, "Unifying obstacle detection, recognition, and fusion based on millimeter wave radar and RGB-depth sensors for the visually impaired," *The Review of Scientific Instruments*, vol. 90, no. 4, pp. 044102, 2019.

[7] X.-Y. Zhang, M. Zhou, P. Qiu, Y. Huang, and J. Li, "Radar and vision fusion for the real-time obstacle detection and identification," *Industrial Robot*, vol. 46, no. 3, pp. 391-395, 2019.

[8] S. Kumar, "Binocular Stereo Vision Based Obstacle Avoidance Algorithm for Autonomous Mobile Robots," *2009 IEEE International Advance Computing Conference*, pp. 6-7, 2009.

[9] G. Toulminet, M. Bertozzi, S. Mousset, A. Bensrhair, and A. Broggi, "Vehicle detection by means of stereo vision-based obstacles features extraction and monocular pattern analysis", *IEEE Transactions on Image Processing*, vol. 15, no. 8, pp. 2364-2375, 2006.

[10] K. He, X.-M. Ma. "Research on avoidance obstacle strategy of coal underground inspection robot based on binocular vision," *29th IEEE Chinese Control and Decision Conference (CCDC)*, pp. 28-30, 2017.

[11] K. Wu, Z. Chen, L. Zhou, and M. Li, "Research on Obstacle Detection and Tracking Method of Unmanned Sailboat Based on Millimeter Wave Radar," *2019 IEEE Chinese Automation Congress (CAC)*, pp. 22-24, 2019

[12] J. Duan, L. Shi, J. Yao, D. Liu, and Q. Tian. "Obstacle Detection Research based on Four-Line Laser Radar in Vehicle," *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Shenzhen, vol. pp. 12-14, 2013.

[13] Z. Silar, M. Dobrovolny, "The obstacle detection on the railway crossing based on optical flow and clustering," *36th IEEE International Conference on Telecommunications & Signal Processing (TSP)*, pp. 2-4, 2013.

[14] T. Tang, S. Zhou, Z. Deng, H. Zou, and L. Lei, "Vehicle Detection in Aerial Images Based on Region Convolutional Neural Networks and Hard Negative Example Mining," *Sensors-basel*, vol. 17, no. 2, pp. 336-353, 2017.

[15] Y. Xu, G. Yu, Y. Wang, X. Wu, and Y. Ma, "Car Detection from Low-Altitude UAV Imagery with the Faster R-CNN," *Journal of Advanced Transportation*, pp. 1-10, 2017.

[16] Y. Wang, C. Wang, H. Zhang, C. Zhang, and Q. Fu, "Combing Single Shot Multibox Detector with transfer learning for ship detection using Chinese Gaofen-3 images," *2017 IEEE Progress in Electromagnetics Research Symposium-Fall (PIERS-FALL)*, pp. 19-22, 2017.

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23-28, 2014.

[18] R. Girshick, "Fast R-CNN," *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440-1448, 2015.

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149, 2017.

[20] W. Liu, D. Anguelov. D. Erhan, "SSD: Single Shot MultiBox Detector," *CoRR*, vol. abs/1512.02325, pp. 21 -37, https://doi:10.1007/978-3-319-46448-0_2, 2015.

[21] C. Tang, H. Hu, P. Wei, Z. Duan, and Y. Qian, "An improved YOLOv3 algorithm to detect molting in swimming crabs against a complex background," *Aquacultural Engineering*, vol. 91, pp. 102115, 2020.

[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27-30, 2016.

[23] J. Redmon, A. Farhadi, "YOLO9000: Better, Faster, Stronger," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21-26, 2017.

[24] J. Redmon, A. Farhadi, "YOLOv3: An Incremental Improvement," University of Washington, https:.//doi.org/10.48550/arXiv.1804.02767, 2018.

[25] K. Wang, C. Chen, M.-S. Hossain, "Transfer reinforcement learning-based road object detection in next-generation IoT domain," *Computer Networks*, vol. 193, pp. 108078, 2021.