**Multi-object Road Waste Detection and Classification based on Binocular Vision**

**Abstract**

A road multi-object detection algorithm is one of the core algorithms for intelligent road cleaning robots relying on machine vision. Most existing object detection algorithms analyze all image regions and finally calculate the category and location of each object. These algorithms are suitable for various applications and achieve accurate results when detecting objects of small size or large numbers. However, detecting objects on the road surface where the background changes little and the number of objects is small is challenging, imposing many invalid calculations during the detection process. Therefore, this paper proposes a multi-object detection method using a binocular camera and a convolutional neural network (CNN) that effectively reduces invalid calculations during the detection process and improves detection efficiency. In the developed method, the binocular vision image acquired by the binocular camera is stereo matched and equalized, while linear regression and coordinate transformation eliminate the angle of the camera pair concerning the road surface. Then, we calculate the coordinates of the regions of interest (ROI) in the left vision image and extract the features within the ROI from the corresponding CNN’s feature map. Next, ROI pooling resizes the extracted feature maps of different sizes to the same size, which are then input to the fully connected layers to output the results. The proposed binocular network and the Faster R-CNN (VGG16) are trained and tested on a dataset involving 1000 road waste images. The experimental results demonstrate that the developed binocular network improves the detection accuracy and speed by 21.3% and 62.9%, respectively, compared with Faster R-CNN (VGG16), providing a reliable algorithm basis for a machine vision-based intelligent road cleaning robot.

**Keywords：** binocular vision positioning; neural network; deep learning; waste identification and classification

**0 Introduction**

With the continuous improvement of people's quality of life, their requirements for a comfortable living are continuously increasing, with road cleanliness being an essential factor affecting people's living comfort. On the other hand, waste classification is one of the ways to protect the environment and has recently become a part of national policy. Road cleaning is time-consuming and laborious, mainly done manually by driving the road sweeper. Although the amount of labor is significantly reduced and the cleaning speed has been improved, manual operation is still required, and the waste after cleaning must be manually classified. Thus, we design an intelligent road cleaning robot that automatically detects, cleans, classifies, and stores road waste.

In order to improve the working efficiency of the road cleaning robot, the robot vision system must identify and localize the objects quickly and accurately. Typically, the road cleaning robot operates on the road, which has a common object background, and the number of objects that appear simultaneously in each image is small. However, currently, there is no specialized object detection algorithm for such a scene. Although most existing detection algorithms can accurately detect objects in various scenes, the invalid calculations will increase significantly in scenes with more objects and changeable backgrounds. Thus, reducing these invalid calculations improves road waste detection efficiency. Therefore, this paper utilizes binocular camera positioning to reduce invalid calculations and improve detection efficiency.

The remainder of this paper is organized as follows: Section 1 discusses the related research on object detection, and Section 2 introduces the architecture of the proposed network and discusses in detail the developed algorithm to extract the regions of interest. Section 3 presents the network’s loss function, and Section 4 describes the comparison and real scene experiments and the network’s performance. Section 5 discusses the results, and finally, Section 6 concludes this work and suggests future research directions.

**1 Related Work**

This section presents current popular object detection methods and their applications, studies the binocular camera performance, and finally, focuses on combining binocular camera and neural networks.

In the early years, although machine vision could identify objects accurately in a single environment, it presented low adaptability in a complex environment. With the introduction of neural networks and deep learning, machine vision has made a breakthrough. Recently, neural networks have developed rapidly, with Region-Convolutional Neural Network (R-CNN) beginning a milestone extending single target identification to multi-target detection. In R-CNN, 2000 regions of interest (ROI) are extracted from the input image and input into a convolutional neural network (CNN) to compute features, and finally, classify the ROIs by employing a support vector machine (SVM) [1] and regress the bounding boxes. R-CNN was then extended to Fast-RCNN, which extracted the ROIs from the feature map. Compared with R-CNN, which extracted the ROIs from the input image, Fast-RCNN reduced the calculations and improved detection efficiency. In the Faster-RCNN scheme, the Region Proposal Network (RPN) was proposed to generate the ROIs, with RPN improving the ROI’s quality and reducing their number. Compared with Fast-RCNN, Faster-RCNN attains an improved performance [2-6]. An alternative method is the You Only Look Once (YOLO) algorithm, which predicts bounding boxes and class probabilities directly from full images in a single evaluation. YOLO can be optimized end-to-end directly on the detection performance [7].

Object detection algorithms based on neural networks have been applied in various fields [8-15]. For instance, in [16], the authors proposed a deep learning-based object detection approach that detects objects from the acquired disk image of the suspect machine to make the forensic investigation process fast, efficient, and robust. Lee *et al.* [17] from South Dakota State University fused 2D images and 3D point clouds to filter out false alarms originating from the Mask RCNN detection results and improved object detection accuracy in road scenes. He *et al.* [18] from the Chinese Academy of Sciences developed a pedestrian flow statistics algorithm based on a front-down monocular camera. Specifically, a CNN detects the pedestrian targets in the image, and the uniform linear motion models are established with a Kalman filter. A data association algorithm tracks the target, and pedestrian flow is counted through a pedestrian counting method.

 Considering the performance of a binocular camera, Sukarnur *et al.* [19] from the University Technology MARA summarized the factors affecting the accuracy of binocular camera distance detection. Among all factors, the authors claimed that the camera’s detection range and calibration accuracy have the most significant impact on the distance detection accuracy, while color is not the main factor affecting accuracy. Regarding combining a binocular camera with neural networks, scholars from Jiangsu University proposed a pedestrian obstacle detection method, which input the one-side image captured by a binocular camera into an improved YOLOv3 to detect the position of pedestrian within the image and calculated the pedestrians’ three-dimensional coordinates exploiting the images of both sides (left and right image of the binocular camera).

To our knowledge, the closest papers to our research are the following. Scholars from Donghua University proposed an identification and positioning method to detect plastic bottles and cans. They use YOLOv4 to identify objects, extract bounding boxes, and then use stereo matching to achieve three-dimensional positioning. Researchers from Harbin Engineering University proposed a binocular vision garbage classification system, which adopted the improved mobile net-single shot multi-box detector (MSSD) as the front-end primary network and embedded the Resnet50 and the convolutional block attention module (CBAM) for garbage detection. Finally, the stereo matching algorithm was used to calculate the garbage depth information. Nevertheless, in both works, the authors first use a neural network to detect the target and a binocular camera to locate it. Thus, in both methods, the separate detection and location processes reduce the entire algorithm's detection efficiency. Therefore, this work focuses on integrating the binocular camera location algorithm into the neural network to improve the accuracy and detection speed of the entire algorithm.

**2 Network Architecture**

The proposed network architecture is illustrated in Figure 1. The size of the binocular image captured by the binocular camera is 1280×480 pixels, and the algorithm first crops the image to the left and right image, with each image having a size of 640×480 pixels. Then, the images are transformed to gray images and are input into a stereo matching algorithm to calculate the corresponding depth image. The difference of pixels’ gray value in the depth image is enhanced via equalization, while linear regression and coordinate transformation eliminate the angle of the camera pair concerning the road surface. Then the left vision image's ROIs are calculated using sliding window scanning.

According to the CNN’s input image requirements, the left vision image cropped from the binocular vision image is resized and padded and then input into the CNN to obtain the feature map. The ROI features are extracted from the feature map by coordinate mapping, while ROI pooling resizes the features within the ROIs to have the same size. Then the resized features are input to the fully connected layers to obtain the output results.

* 1. **Calculation of The Regions of Interest**

**2.1.1 Stereo Matching and Equalization**

This paper utilizes the OpenCV library's stereo calibration algorithm and block matching (BM) algorithm to create the depth image, which directly presents the distance between the object and the camera. The gray value of each pixel ranges from 0 to 255, and the higher the value, the closer the distance between the object and the camera. The detected objects are the wastes on the road with their height being less than 30mm, concentrating the pixels’ gray values in the depth image in a certain range. We equalize the depth image to enhance the difference of the pixels’ gray value in the depth image and distinguish the different objects better. Figure 2 (a) illustrates the origin of the image captured by the binocular camera, subfigure (b) depicts the depth image achieved by stereo matching, (c) shows the image after equalization, and (d) and (e) present the gray histogram of the effective area in the depth image before and after the equalization. Figures 2 (d) and (e) highlight the number of pixels per gray value after equalization. Moreover, Figures 2 (b) and (c) reveal that the histogram equalization process enhanced the difference in the pixels’ gray values at different depths within the image. From Figure 9, we also observe that due to the angle of the binocular camera concerning the road surface, the gray value of the pixels is increasing gradually in an uncertain direction. Thus, to calculate the ROIs more accurately and increase the detection accuracy and the network’s detection speed, the angle of the binocular camera concerning the road's surface has to be compensated.

**2.1.2 Angle Elimination**

The size of the effective area in the depth image is . Considering the top left corner of the area as the origin, the plane with a gray value of 0 is the XOY plane, and we build a three-dimensional coordinate system illustrated in Figure 3, where the Y-axis points right, the X-axis points down, and the Z-axis points outward. Each pixel in the effective area of the image is a point. The pixels’ position relative to the origin is the points’ x and y coordinates, and the gray value is the z coordinate.

 Starting from the origin, we scan the z-coordinate value of each pixel row by row in the effective area and find the pixel with the minimum gray value closest to the origin, defined as point . Given the angles of the binocular cameras to the road surface,  is usually on the left or right side of the effective area. Therefore, starting from , our method scans the z value of the points having the same x value as  with a step size of  in the effective area. Then, we take the z value of  as the base value, calculate the z value difference between the point in the current step and the previous step, and record the point in the current step if the difference is less than 10. If the amount of the recorded points reaches 30, scanning and recording the pixels is completed. Otherwise, if the current row scanning completes and the pixels counted are less than 30, all the recorded points are removed, and a new point  is found, which is in the same column with , and the z value of  is the new base value. The scanning and recording process restarts at a new row.

When 30 points are successfully recorded, linear regression is used to fit the 30 points in line. Since the recorded points have the same x value , i.e., in-plane values of , the coordinates of the 30 points are  , and the target line is . The proposed method adopts the least squares algorithm and the square of the distance between the point and the line to make the partial derivative of  and  equal to 0.





By solving the above equations, we achieve:





In the plane , the target line  and line  intersect at a point, while line passes through the intersection and is parallel to the X axis, expressed as:



From  we can obtain the angle , which is the rotation angle of the road surface about line concerning plane XOY. All points in the effective area are rotated around  for . Let a point’s coordinates before the rotation be  and after the rotation be . Thus:



By saving the  value of each point after rotation and the  value before rotation, the new points  are obtained, which create a new image that contains a new road surface.

In the new image, starting from the origin, we scan the z-coordinate value of each pixel row by row in the effective area and find the pixel with the minimum gray value closest to the origin. Let this point be . Starting from , the proposed method scans the z value of the points that have the same y value as  with step size of  in the effective area. By considering the z value of  as the base value, we calculate the z value difference between the point in the current step and the previous step and record the point in the current step if the difference is less than 10. If the number of recorded points reaches 30, the process of scanning and recording the pixels terminates. Otherwise, all the recorded points are removed, and a new point  is found, which is in the same row with . The z value of  is considered the new base value, and the process of scanning and recording the points restarts in a new column.

When 30 points are successfully recorded, linear regression is used to fit the 30 points in line. Because the recorded points have the same y value , they belong in the plane , the coordinates of the 30 points are , and the target line is . According to the least squares algorithm, we minimize the square of the distance between the point and the line and make the partial derivative of  and  equal to 0.





By solving the equations simultaneously, we achieve:





On the plane, the target line  and line  provide an intersection at the point. Line passes through the intersection and is parallel to the Y axis. The expression is:



From , we obtain angle , which is the rotation angle of the new road surface about line with respect to plane XOY. All points in the effective area are rotated around  for . The coordinates of a point before the rotation are  and after the rotation .



The  value of each point after rotation and the x and y values before rotation are saved, and the final points  are obtained. These points comprise an image that eliminates the angle of the binocular camera with respect to the road's surface. The final result is achieved by increasing the image contrast after the angle elimination, as illustrated in Figure 4, which displays the objects.

**2.1.3 Proposal of the Regions of Interest**

Window scanning is used to define the ROIs within the image. The window size is set 5×5, with a step size of five. During each step, the maximum gray value in the window in the current step is compared with the maximum value of the previous step. If the difference exceeds 30, the coordinates of the window’s top left corner at the current step are recorded, and the window continues scanning until the difference exceeds 30 again. Then the coordinates of the bottom right corner of the last window are recorded. The top left and bottom right coordinates form a new window. Overall, the scanning process starts from the origin. The window slides from left to right row by row and once row scanning is finished, the window slides from top to bottom column by column. Thus, the entire image is scanned twice. Once the scanning process completes, the system will record many new windows covering the objects within the image, as visualized in Figure 5. These new windows are then merged to provide the final results. However, the contents in these windows may belong to different objects, so it is necessary to determine whether to merge them.

The window merging process is based on the th window recorded during window scanning. Precisely, the positional relationship of the th and tth windows is estimated, where  ranges from 1 to the number of the remaining windows. According to the positional relationship between the two windows, the algorithm determines whether to merge them or not (further details will are presented below). If the windows are merged, the coordinates of the th window will be updated to the coordinates of the merged window, while the coordinates of th windows will remain unchanged. After the merging process based on the th window completes, the base window changes from the th to the th window, and the merging operation repeats until all windows are evaluated.

During the window merging operation, the position of the th window relative to the th window is divided into the following cases:

1. Two windows separate from each other. In this case, the contents of the two windows will be judged as different objects, so they will not be merged, and the coordinates of the th window will remain unchanged.

2. One window is inside the other. This suggests that the contents of the two windows belong to the same object. Thus, the coordinates of the th window will be updated to the coordinates of the larger window.

3. The two windows intersect. In this case, the contents of the two windows may belong to different objects, so it is necessary to determine whether to merge them.

Since only the coordinates of the th window are updated during window merging, the window size is always greater than or equal to the th window. In our algorithm, the direction of the th window relative to the th window is divided into four situations: left, right, top, and bottom. If the relative direction is left, it is subdivided into three specific positions: top left, left, and bottom left. Accordingly, if the relative position is right, it is subdivided into three specific positions: top right, right, and bottom right. In any case, the algorithm first estimates the direction of the th window relative to the th window and then estimates the specific position. If the windows have different specific positions, the algorithm determines whether to merge the two windows. If merged, the th window becomes the outermost window at its position in the th window, and its coordinates will be recorded additionally to determine whether to merge the windows at the same position.

Take the th window on the left side of the th window as an example to describe the merging process. First, the two-dimensional rectangular coordinate system is built in the image, where the top left corner of the effective area is the origin of the coordinate system. The positive direction of the X axis points to the right, and the positive direction of the Y axis points to the bottom (Figure 3). The coordinates of the top left corner and the bottom right corner of the th window are  and . Accordingly, the top left and bottom right corner coordinates of the th window are  and . If the th window is on the left side of the th window. In this situation, if  and , the th window is at the bottom left of the th window. Then we calculate the distance between the bottom left window, which is recorded additionally, and the th window. If the distance is less than the preset value, the two windows are merged and the coordinates of the th window are updated to the coordinates of the merged window. Moreover, the coordinates of the bottom left window, which is recorded additionally, are updateed to the coordinates of the th window. Otherwise, if the distance exceeds the set value, the two windows will not be merged.

If  and , the th window is at the left of the th window, the algorithm calculates the distance between the left window, which is recorded additionally, and the th window. If the distance is less than the set value, the two windows are merged, and the coordinates of the th window are updated to the coordinates of the merged window. Moreover, the coordinates of the left window, which is recorded additionally, are updated to the coordinates of the th window. Otherwise, if the distance exceeds the set value, the two windows will not be merged.

If  and , the th window is at the top left of the th window. Then the algorithm calculates the distance between the top left window, which is recorded additionally, and the th window. If the distance is less than the set value, the two windows are merged, and the coordinates of the th window are updated to the coordinates of the merged window. Additionally, the coordinates of the top left window, which is recorded additionally, are updated to the coordinates of the th window. Otherwise, if the distance exceeds the set value, the two windows will not be merged.

Regarding the other cases, if the th window is on the right side of the th window, it is subdivided into three specific positions. Otherwise, if the th window is on the top or bottom of the th window, it is no longer subdivided. The merging process of the other situations is similar to the situation where the  window is on the left side of the th window.

After the merging process, the ROIs are determined by deleting the duplicate windows whose size and position are the same and the windows whose size is less than the set value. The final output is illustrated in Figure 6.

**2.2 Coordinate mapping**

The image captured by the binocular camera is split into the left and right vision images. The depth image generated by stereo matching is based on the left vision image, so the position of the object in the depth image is the same as that in the left vision image. Therefore, the left vision image is input to the CNN to acquire the feature map.

 In the last section, the ROIs’ size and the position in the left vision image were calculated. Thus, to extract the features within the ROIs, we use coordinate mapping to calculate the ROIs’ size and position in the feature map. According to CNN’s input size requirements, we first resize the left vision image without changing its aspect ratio. The size requirement for the image is , while the size of the left vision image is . Hence, the size of the resized image is .



Then the resized image is embedded into the top left area of an empty image of size .

The size of ROI in the left vision image is  , and the coordinates of the ROI’s top left corner are . The ROI size in the feature map is , while the coordinates of the top left corner of the ROI are . Finally, the feature map size output by the CNN is , adopting the following coordinate mapping:



By rounding down  and , and rounding up  and , then the ROI’s size and the position in the feature map are determined.

**2.3 ROI Pooling**

ROI pooling is used to resize the feature map's ROIs to be the same size.

**3 Loss Function**

The network's output comprises two parts: a discrete probability distribution , where k is the ID of the class, and the bounding-box regression offsets for each class . Training the classification and bounding-box regression relies on a multi-task loss L.



The first part of the loss is the log loss for the true class u:







where and are the predicted and the true bounding-box regressions for class u and  is a hyper-parameter that balances the two losses [20].

**4 Experiment**

The following experiment comprises four stages: backbone selection, dataset preparation, training and testing, and real scene experiment.

The backbone is an essential part of the binocular network to acquire the feature map. We tested four models, i.e., Darknet53, Darknet19, Resnet101, and VGG16, using the ImageNet dataset, and finally selected Darknet53 as the backbone of the binocular network. The performance of the proposed binocular network is evaluated on 1000 images of road wastes we collected to generate the dataset. During the trials, we trained and tested the binocular network, Faster R-CNN, and YOLOv4 using the generated dataset, and evaluated them based on the average precision (AP), mean average precision (mAP), and average detection time as performance metrics. Additionally, to evaluate the binocular network’s stability and present the results of the key detection processes, we test the binocular network in various real scenes.

**4.1 Backbone Selection**

The backbone selection process evaluated four models, i.e., Darknet53, Darknet19, Resnet101, and VGG16, on the ImageNet dataset based on single-crop validation y, , ( graphics card) Table 1 reports the corresponding results, highlighting that Darknet53’s Top-5 accuracy is similar to Resnet101 but faster to execute. Darknet19 is the most processing efficient, but its accuracy is lower than that of Darknet53 and Resnet101. Regarding VGG16, its overall performance is inferior to the competitor models. Therefore, we choose Darknet53 as the backbone.

**4.2 Dataset Preparation**

Given that the input image size for the binocular network differs from the other networks, this paper prepares two datasets with images of different sizes.

About 1000 road waste images were captured using the HBV-1780 binocular camera. Each image was 1280×480, so we cropped it to its left and right vision images, each having a size of 640×480. Then we embeded the left vision image into an empty image of size 640×640 and set the pixel value of the invalid area as zero. The objects were then labeled in the 640×640 left vision image, and a .txt file was output containing the true class and each object's bounding box. Ultimately, the 1280×480 binocular vision images and the .txt files comprised the binocular dataset, 80% of which was used for training and the rest for testing. Figure 7 illustrates the process of generating the dataset. The employed labeling is , where *I* is the class ID,  are the coordinates of the center of the bounding box, and  is the size of the bounding box. Table 2 presents the Class ID.

Regarding Faster RCNN（VGG16）and YOLOv4, these were also trained and tested on our dataset. Specifically, the processed 640×640 left vision images and the .txt files were used as the dataset, where 80% was used for training and the rest for testing.

**4.3 Training and Comparison Testing**

In this section, we train and test on our dataset the Darknet53 binocular network, Faster R-CNN (VGG16), and YOLOv4 and compare their testing results. Table 3 reports the hardware setup for the training and testing, Table 4 presents the parameters used, and Figure 8 depicts the multi-task loss of the Darknet53 binocular network. We employed 20% of the dataset to test the competitor networks, and during the test, the processing time involved from loading the images until the detection process completes. It should be noted that the detection time of the binocular network included the image processing time and AP and mAP were employed as evaluation indicators. The testing results for each network are shown in Table 5, which reveals that all networks present a low detection accuracy for the plastic straw. Moreover, the Darknet53 binocular network and Faster R-CNN（VGG16） have a higher detection accuracy for the paper box than the other categories, with our scheme attaining the highest detection accuracy. YOLOv4 has the highest detection accuracy on waste paper. Overall, the classification accuracy and detection speed of the Darknet53 binocular network is higher by 21.3% and 62.9% compared with Faster R-CNN（VGG16）, while it attains a similar performance to YOLOv4.

**4.4 Real Scene Experiment**

In order to test the performance of the binocular network in a real scene, a group of binocular vision images of different road scenes was taken on our campus. The corresponding hardware platform is illustrated in Figure 9, and the system block diagram in Figure 10. The experimental system comprised a binocular camera and a computer. The binocular camera was a synchronous HBV-1780 with a resolution of 1280×480 and a maximum dynamic of 72dB, placed 900mm above the ground. The computer had an NVIDIA GeForce GTX 950M GPU, with a compute capability of 5.0.

Figure 11 depicts the detection processes and results using the Darknet53 binocular network in four different scenarios. Figure 11 (a) - (d) are original images of size 1280×480, presenting a scene of (a) brick pavement with ant nests and ponding, (b) a dirt road with weeds, (c) brick pavement that has two colors and with potholes on it, and (d) an object on the manhole cover. Moreover, Figure 11 (e) - (h) presents the detection processes of the four original images, each including five images: after stereo matching, after equalization, after eliminating the angle, ROI, and the detection results. Figure 11 highlights that the environment has little influence on the stereo matching process in the first three scenes. Therefore, the ROIs are accurately determined, and the objects are correctly classified. In the fourth scene, the environment greatly influences the stereo matching process, so the number of ROIs increases, but the detection results are still correct.

**5 Discussion**

The results of both trials, i.e., comparison testing and real scene experiments, demonstrate that the classification accuracy and detection speed of the Darknet53 binocular network is significantly improved compared with the original Faster RCNN (VGG16). Under various scenarios on the road, the binocular network can detect objects correctly, proving the network’s robustness and demonstrating an appealing classification performance.

The developed binocular network relies on the binocular camera positioning algorithm to determine the ROIs. Compared with Faster RCNN, the number of ROIs achieved by the binocular network is significantly reduced, affording the binocular network to detect faster. By equalizing the depth image and eliminating the angle, the algorithm can accurately separate the objects from the road surface, enhancing the accuracy of ROIs and extracting targeted regional features. Moreover, the suggested method improves the quality of the data sent to the fully connected layers, reduces the network’s calculations, ensures detection accuracy, and improves the network’s detection efficiency. Additionally, selecting Darknet53 as our scheme’s backbone improves our network’s detection accuracy to a certain extent.

In the detection process, stereo matching significantly influences the final detection result of the binocular network. This is because the premise for the binocular network to make a correct prediction is to extract the ROIs correctly, and the stereo matching result is the basis for the network to determine the ROIs. It should be noted that the detection range of the binocular camera, calibration accuracy, and substantial illumination change in the real scenes are factors affecting stereo matching. However, in our project, the detected objects are on the road, so the detection distance changes a little, and because the camera is facing the ground during detection, the illumination changes are negligible. Thus, the stereo matching results are accurate enough to provide correct detection results.

The designed binocular network will be used as the core algorithm of the vision system of an intelligent road cleaning robot that detects the wastes while cleaning. The robot will automatically fine-tune the cleaning route according to the positioning information of the wastes and store the waste separately according to their category. The designed binocular network can also be used for detection when the illumination intensity and distance change slightly.

**6 conclusion**

This paper designs a multi-object detection algorithm based on binocular camera positioning. The ROIs are determined by processing the binocular vision image, and by mapping them to the feature map we extract the regional features. The latter is sent to the pooling and fully connected layers for object classification. The comparison test and the real scene experiment demonstrate that the performance of the designed binocular network has achieved the expected effect. Future works will investigate improving the robustness of the stereo matching of the binocular vision image, employ other methods to obtain more stable depth images, and improve the network’s detection performance further and expand its application scope.