








# Design of a Blind Guidance System Based on RealSense and the Improved YOLOv5 Algorithm

Zhao Zhang<sup>1</sup> , Xiaobin Shen<sup>1</sup> , Jing Ge<sup>1</sup> , Yingying Zha<sup>1</sup> , Lisai Liu<sup>1</sup> ,  
and Sheng Liu<sup>1,2</sup> 

<sup>1</sup> College of Computer Science and Technology, Huaibei Normal University, Huaibei 235000, China

Liurise@139.com

<sup>2</sup> Anui Engineering Research Center for Intelligent Computing and Application on Cognitive Behavior, Huaibei 235000, China

**Abstract.** To assist blind people in travelling and solve the problems of high hardware costs, insufficient portability of blind assistance devices, and vulnerability to environmental impacts, a modified YOLOv5 algorithm (YOLOv5-CM) based on ground plane segmentation and Euclidean clustering algorithms is proposed and applied to a blind guidance system based on RealSense D435. This proposed algorithm adds coordinate attention and uses MobileNetv3 as the backbone network for YOLOv5s to extract the main features. Compared with those of the YOLOv5 model, the mAP was improved by 0.7%, the model size was decreased by 79.1%, and the number of parameters was reduced by 80.3%. In this study, the YOLOv5-CM algorithm was applied to design a blind guidance system. When the system detects a traffic light, the RealSense D435 is used to obtain depth images to detect and measure the distance from the pedestrian traffic light. The system provides warnings based on the measured distance. When the system does not detect a traffic light, it utilizes the ground plane segmentation and Euclidean clustering algorithms to obtain the location and distance of obstacles. This information is conveyed to the blind through voice guidance, which aids navigation.

**Keywords:** Machine Vision · Blind Guidance System · RealSense · YOLOv5 · MobileNetv3 · Coordinate

## 1 Introduction

According to statistics from the World Health Organization, over 2.2 billion people worldwide were visually impaired in a 2022 survey [1]. Visually impaired people face great difficulties and dangers in daily travel due to the unavailability of proper visual information support. Therefore, the development of assistive technology facilitating safe travel has attracted much research attention. Correspondingly, we have designed a blind guide system to provide better support while travelling.

Among the numerous challenges faced by blind people, some are unemployment, reading, writing, daily activities, and travel [2–4]. In particular, blind people face several travel-related difficulties. Accordingly, guide dogs, guide poles, and GPS-enabled devices are widely used to assist in travel. Although guide dogs possess the advantages of speed, flexibility, and good affinity, there are some problems, such as scarcity and long training times [5]. Likewise, guide poles offer the advantages of convenience and low cost; however, they cannot provide remote warnings and guidance. GPS-enabled devices have benefits such as high positioning accuracy and real-time performance; however, they cannot cope with indoor environments. Perception systems based on artificial intelligence (AI) are considered the best means to assist blind people in travelling. Recently, various AI-based blind-assistance systems have been introduced and applied. For instance, Tapu et al. [6] used a multi-scale Lucas Kanade feature tracking algorithm combined with a smartphone camera to detect obstacles. However, the system lacks three-dimensional (3D) information and cannot determine the distance between obstacles. Rodriguez et al. [7] put forward a system depended on stereo vision, which first conducted plane segmentation to detect pixels on the ground and then used polar grid symbols to detect obstacles in depth images. Although these systems have reliable obstacle detection capabilities, they cannot understand scenes of traffic roads and sidewalks or possess advanced perceptual capabilities depended on deep learning, such as semantic image segmentation and object detection.

This paper introduces a guide system based on RealSense and the YOLOv5-CM object detection algorithm. The system can not only judge the location and distance of obstacles according to 3D information but also obtain the status and distance information of traffic lights on sidewalks and transmit the results to a blind person through voice prompts to improve their perception and cognitive ability of the surrounding environment and reduce their risk of encountering dangers during travel.

## 2 Methodology

In the system operation process, when a traffic light is detected, the system performs recognition and distance measurements. When a visually impaired person walks at a pedestrian crossing, the system announces the traffic light status every 5 s. When a visually impaired person is less than 2 m from a traffic light, the system prompts them to cross the pedestrian crossing. If traffic lights are not detected, the system detects the positions and distances of obstacles. The camera field of view is divided into five directions: front, front-left, front-right, left, and right. The system first performs ground plane segmentation, followed by obstacle clustering and distance measurements. Every 3 s, the system announces the direction and distance of the nearest obstacle within 2 m. The visually impaired person receives relevant detection information through headphones. Fig. 1 illustrates a schematic of the system design.

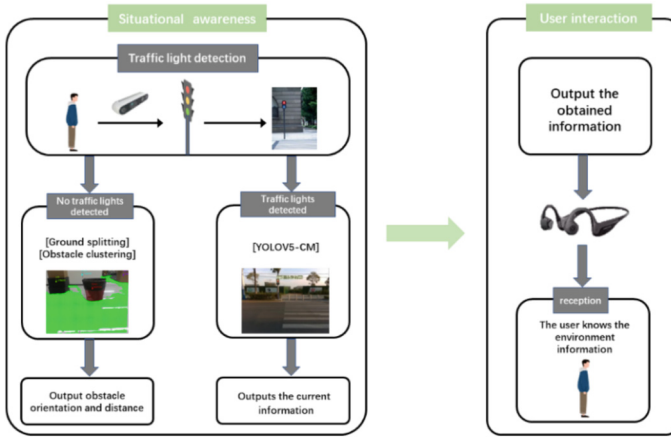


Fig. 1. The guide system diagram.

## 2.1 The YOLOv5-CM Improved Algorithm

**Introduction to the YOLOv5 Algorithm.** Target detection algorithms for deep learning can be categorized into two-stage algorithms, such as faster region-based convolutional neural network (R-CNN) [8], and one-stage algorithms, such as solid-state drive (SSD) [9] and you only look once (YOLO) [10–13]. These algorithms are widely used in computer vision applications. These two types of algorithms exhibit unique characteristics. One-stage algorithms have advantages in speed, while two-stage algorithms have advantages in accuracy. Considering both model precision and timeliness, this study used the improved version 5 of YOLOv5s as a pretrained model to implement pedestrian traffic light detection. The structure comprises the input, backbone, neck, and head, as shown in Fig. 2. The input is used to preprocess image; the backbone extracts features from the preprocessed image; the neck module blends or combines the feature maps to form more complex features; and the head module generates bounding boxes, predicts the features of the image, and predicts the classes.

**Embedded Coordinate Attention Mechanism.** During the convolution process in the YOLOv5 algorithm, the iterations of the algorithm can accumulate a large amount of redundant background information, which can easily cause the loss of features related to traffic lights, resulting in low precision. Therefore, we added a coordinate attention mechanism (CA) [14]. To ease the previous attention mechanisms (such as squeeze and excitation networks (SE Nets) [15] and convolutional block attention module (CBAM) [16]) and address the two-dimensional global pooling location information loss problem, the CA mechanism module improves the expression ability of feature learning in mobile networks. As shown in Fig. 3, there are two steps in this process. The first step is the embedding of coordinate information. For the input feature graph  $X$  with dimensions of  $C \times H \times W$ , global average pooling is performed using the pooling kernels of size  $(H, 1)$  and  $(1, W)$  in the horizontal and vertical coordinate directions, respectively. The second step is to generate the coordinate information feature map. Both generated feature graphs are first concatenated, and the dimensionality is subsequently reduced using a

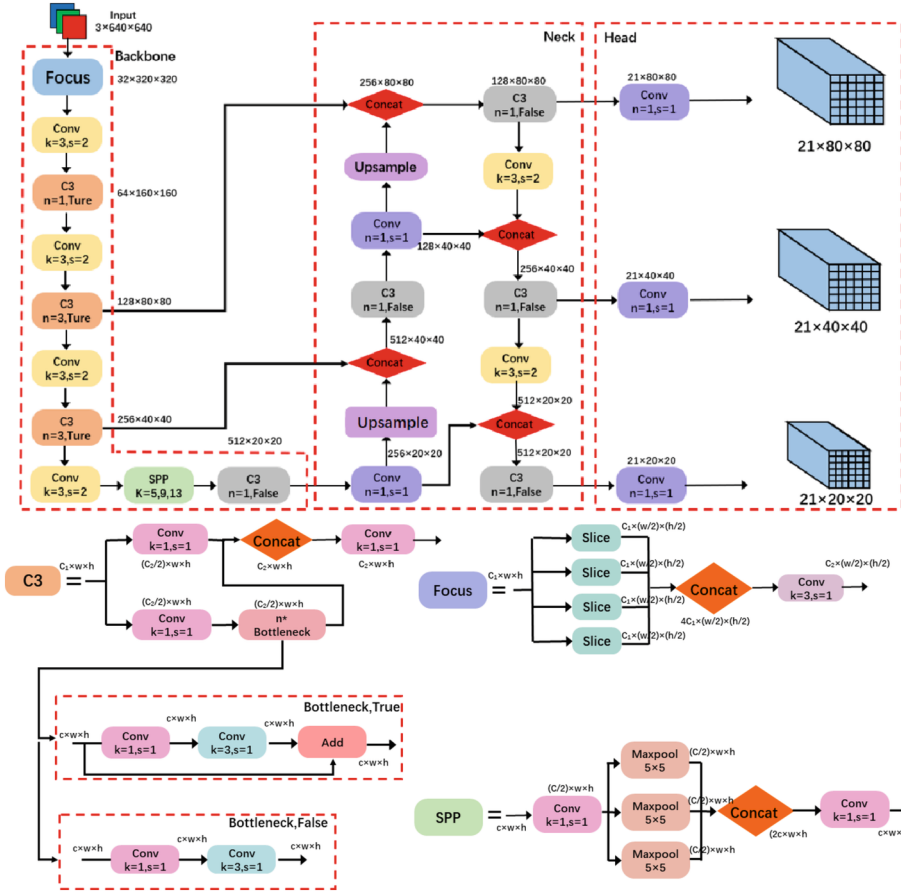


Fig. 2. YOLOv5 network structure diagram.

$1 \times 1$  convolutional kernel. After processing with a BN and an activation function, they were divided into two independent tensors. The resulting tensors undergo two convolutional transformations to produce tensors with the same channel number and are then expanded and output as attention weight allocation values. To identify and locate pedestrian traffic lights more accurately, this paper embedded the CA attention module in the C3 module after the 13th layer. Experiments demonstrate that the improvement can effectively enhance the feature-extraction capability of the network.

**Improvement of Backbone Network.** If the trained YOLOv5 is directly applied to real-world scenarios, issues such as memory and latency must be considered. Lightweight networks have the advantages of low computational costs, few parameters, and short inference time. The MobileNet series of lightweight networks has received considerable attention. As shown in Fig. 4, where the overall structures of the “large” and “small” versions are the same, except for the number of basic units “bneck” and their internal parameters. In this paper, the “small” version of MobileNetv3 is used.

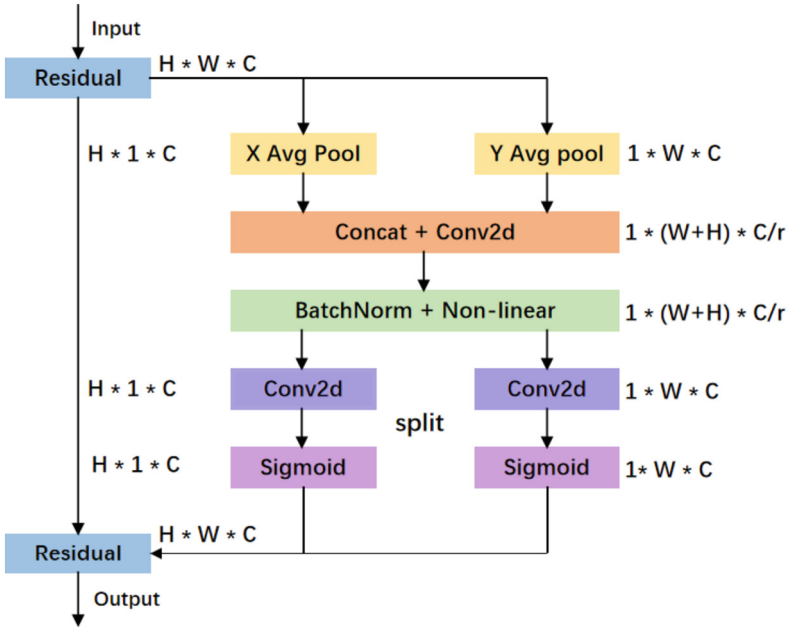


Fig. 3. CA structure.

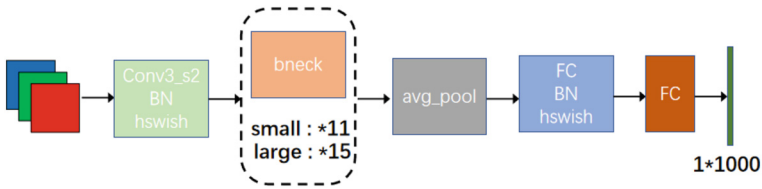


Fig. 4. MobileNetV3 Network structure diagram.

In the MobileNet series of networks, MobileNetV3 [17] is the latest version and has the following four features. First, depthwise separable convolutions are introduced. The depthwise separable convolutions comprise two parts: depthwise and pointwise convolutions for filtering and merging, respectively. The depthwise convolution is performed in the 2D plane, with each kernel corresponding to one channel, which reduces the computation of the network. The pointwise convolution is combined with the output of the depthwise convolution to generate new feature maps. Fig. 5 shows the decomposition of a standard convolution into depthwise and pointwise convolutions. Using depthwise separable convolutions, the computation of the network can be significantly reduced while maintaining good accuracy [18].

MobileNetV3, similar to MobileNetV2 [19], introduces linear bottlenecks and inverted residual structures. The inverted residual structure was significantly different from the original residual structure. The original residual structure first uses a  $1 \times 1$  convolution to reduce the dimensions, a  $3 \times 3$  convolution to achieve feature extraction,

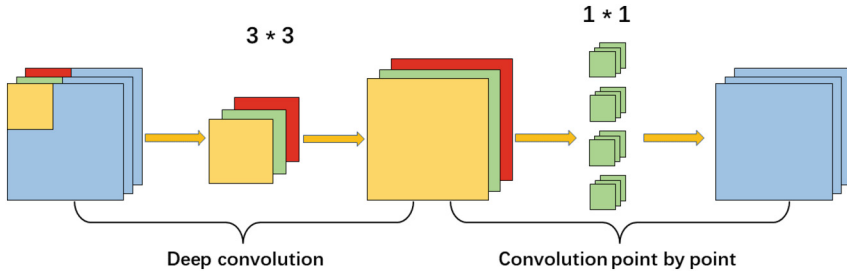


Fig. 5. Standard convolution decomposition process.

and a  $1 \times 1$  convolution to increase the dimensions. The inverted residual structure first uses a  $1 \times 1$  convolution to increase the dimensions, a  $3 \times 3$  depthwise separable convolution to extract features, and a  $1 \times 1$  convolution to decrease the dimensions. To prevent the excessive loss of low-dimensional information, linear mapping was used instead of the ReLU6 layer for the final layer of each block to reduce feature loss and achieve better detection results, as shown in Fig. 6.

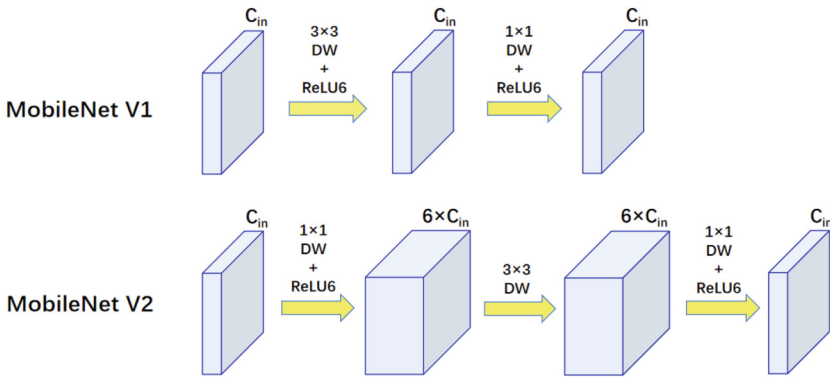


Fig. 6. Reverse residual module structure.

The lightweight attention mechanism of the SENet network is introduced, which is a typical channel attention mechanism, as shown in Fig. 7. Here, Ftr represents the traditional convolution operation; X and U are the input and output of Ftr, respectively; and  $(H' \times W' \times C')$  and  $(H \times W \times C)$  are the dimensions of the original feature map and convolutional feature layer obtained after Ftr convolution, respectively.  $H', W', C'$  and  $H, W, C$  represent the height, width, and channel number, respectively, of the original feature map and the feature layer obtained after Ftr convolution, respectively. The network structure following Ftr is the part added by SENet. SENet can be divided into two parts: squeeze (Fsq(.) in the figure), which primarily uses global average pooling along the height and width directions to get a  $(1 \times 1 \times C)$  feature vector. The excitation (Fex(., W) in the figure) primarily performs two different fully connected operations on the feature vector obtained by the squeeze part, one for dimension reduction and the other

for dimension expansion, to obtain the importance of the different channels.  $F_{scale}(\cdot, \cdot)$  represents a weighting operation that applies the learned importance of different channels to the corresponding channels of the previous feature map to obtain an  $(H \times W \times C)$  output feature map. This structure enhances the directionality of the extracted features by changing the scale of the attention mechanism.

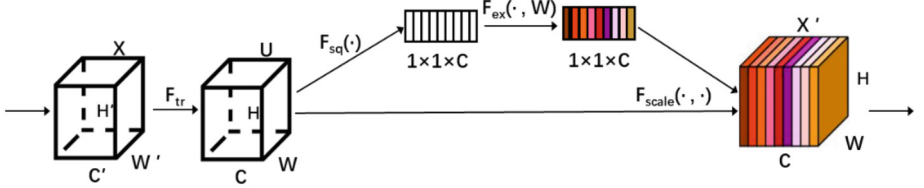


Fig. 7. SENet attention mechanism structure.

Because of the high computational cost, complex derivation, and difficult quantization process of the swish activation function, an h-swish activation function was proposed [20]. The h-swish uses the ReLU6 approximation of the sigmoid function  $\sigma(X)$  to approximate swish, thereby significantly reducing mobile computational resource consumption. The formulas for the swish and h-swish functions are given by Eqs. (1) and (2), respectively.

$$\text{swish} = x \times \sigma(x) \quad (1)$$

$$\text{h-swish} = x \times \frac{\text{RELU6}(x+3)}{6} \quad (2)$$

To further apply the model to real-world scenarios, the backbone network of YOLOv5s was replaced with a MobileNetv3 backbone network to extract features. Experiments demonstrated that this improvement could significantly reduce hardware requirements and the number of parameters.

**Pedestrian Crosswalk Traffic Light Detection and Range Experiment.** As shown in Fig. 8, after the depth camera captures the video stream, the traffic lights are recognized, and the current traffic light status can be effectively distinguished. In addition, using RealSense's infrared range, the distance from the location to the traffic light can be obtained. Through numerous experiments, it is proven that the trained model had high accuracy and stability for real-time detection.

## 2.2 Ground Plane Segmentation

**Methods for Acquiring Point Clouds.** There are many techniques for acquiring point clouds, such as light detection and ranging (LiDAR) [21], structured light [22], stereo cameras [23], and red-green-blue-depth (RGB-D) cameras [24]. LiDAR is expensive, requires contact-based acquisition, and has a slow data processing speed. Structured light is sensitive to lighting and is suitable only for indoor environments. Stereo cameras



**Fig. 8.** Ranging of traffic lights in different scenes.

typically capture relatively coarse depth information. Compared with other point cloud acquisition methods, RGB-D cameras have the advantages of fast data processing, adaptation to lighting changes, and acquisition of more accurate depth information. These advantages make RGB-D cameras more suitable as point-cloud acquisition devices in many application scenarios.

**Real-Time Ground Plane Segmentation.** The point cloud obtained by the sensor contained two categories: obstacles and ground. To obtain accurate obstacle information and avoid introducing additional errors, the ground point cloud needs to be extracted in advance.

The coordinates of each point in the original image were obtained before obtaining the data of the point cloud. From the RGB and depth images obtained by Real D435, the  $x$  and  $y$  coordinates in the pixel coordinate system and the  $z$  coordinate in the camera coordinate system are obtained. The depth image obtained from the RealSense D435 was converted into a point cloud by transforming the coordinate system. Equation (3) is used to transform the pixel coordinate system  $P'$  into the camera coordinate system  $P$ .

$$Z_c P' = Z_c \begin{bmatrix} X_p \\ Y_p \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = KP \quad (3)$$

In the equation,  $(X_c, Y_c, Z_c)$  and  $(X_p, Y_p)$  represent the coordinates in the camera and the pixel coordinate systems, respectively, and  $K$  represents the camera intrinsic matrix.

The transformation between the world coordinate system and camera coordinate system requires only rotation and translation. The rotation matrices were obtained by rotating around different coordinate axes at different angles. First, rotate around the  $z$ -axis by  $\alpha$  degrees as in Eq. (4), then rotate around the  $y$ -axis by  $\beta$  degrees as in Eq. (5),



and finally rotate around the x-axis by  $\varphi$  degrees as in Eq. (6):

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos a & -\sin a & 0 \\ \sin a & \cos a & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R_1 \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \quad (4)$$

Similarly, rotating around the y-axis and x-axis yields

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos b & 0 & \sin b \\ 0 & 1 & 0 \\ -\sin b & 0 & \cos b \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = R_2 \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \quad (5)$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \# & -\sin \# \\ 0 & \sin \# & \cos \# \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = R_3 \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \quad (6)$$

From this, the rotation matrix R can be obtained as shown in Eq. (7).

$$R = R_1 R_2 R_3 \quad (7)$$

Multiplying the rotation matrix by the camera coordinate system coordinates and adding the offset vector yields the coordinates in the world coordinate system, as expressed in Eq. (8)

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} = R \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} + T \quad (8)$$

As the sensor maintains a certain posture with respect to the ground after it is worn, fitting a plane [25] using the random sample consensus (RANSAC) algorithm can yield a ground point cloud. The key of the RANSAC algorithm is to select three points randomly and repeatedly to construct a plane equation, as shown in Eq. (9). If this plane contained sufficient points, it was considered the ground. Using the RANSAC algorithm for robust estimation and random search on the ground point cloud, points are determined to be ground based on Eq. (10). After K iterations, the ground can be preliminarily segmented [26], as shown in Eq. (11).

$$AX_c + BY_c + CZ_c + D = 0 \quad (9)$$

$$d = \frac{|AX_c + BY_c + CZ_c + D|}{\sqrt{A^2 + B^2 + C^2}} < T \quad (10)$$

$$K = \frac{\log(1 - P)}{\log(1 - W^n)} \quad (11)$$

A drawback of the RANSAC algorithm is that the calculation time for fitting a plane using the RANSAC algorithm is too long, resulting in a loss of real-time performance. Therefore, a method of rotating the ground point cloud was adopted. In general, the

coordinate system of the original point cloud obtained by the depth camera is  $z$  moving forward along the optical center,  $x$  to the right, and  $y$  downward. For ease of processing, the point cloud coordinates are transformed into  $z$  moving forward in the direction of the optical center,  $x$  to the left, and  $y$  upward. For real-time processing, the same rotation was loaded. In order to reduce the operation time after the conversion, the VoxelGrid voxel filter in the Point Cloud Library is used to downsample the point cloud. Generally, the ground is located at the bottom of the image point cloud and is concentrated near the smallest coordinates in the point cloud. Therefore, the ground point cloud plane can be obtained by calculating the minimum value of the coordinates and setting a threshold. This eliminates the need for extensive calculations using the RANSAC algorithm, thereby improving the computational efficiency.

In point cloud rotation, let vector  $n_0$  be the original normal vector and  $n_1$  be the target vector. The goal is to rotate  $n_0$  to direction  $n_1$ . The coordinate of  $n_0$  is  $(x_0, y_0, z_0)$ , that of  $n_1$  is  $(x_1, y_1, z_1)$ , and that of the origin is  $O(0, 0, 0)$ . These three coordinates form a plane. From the cross-product of the coordinates, it is concluded that the normal vector of the plane is perpendicular to the plane and passes through the origin of the coordinates, namely the axis of rotation. Using these three points, the equation of the plane can be calculated as shown in Eq. (12).

$$\frac{z_1y_0 - z_0y_1}{x_0y_1 - x_1y_0} \cdot x - \frac{z_1x_0 - z_0x_1}{x_0y_1 - x_1y_0} \cdot y + z = 0 \quad (12)$$

A normal vector of the plane, which is also the vector of the axis of rotation, can be calculated as shown in Eq. (13).

$$(z_1y_0 - z_0y_1)x - (z_1x_0 - z_0x_1)y + (x_0y_1 - x_1y_0)z = 0 \quad (13)$$

The rotation angle  $\theta$  between vector  $n_0$  and vector  $n_1$  can be calculated as the angle between these two vectors, as shown in Eq. (14):

$$\theta = \frac{\vec{n}_0 * \vec{n}_1}{|\vec{n}_0| |\vec{n}_1|} \quad (14)$$

**Obstacle Clustering.** After obtaining the ground point cloud, clustering was performed on the point clouds of the obstacles to obtain more information. After clustering, each point cloud cluster can be regarded as an obstacle, and the distance and azimuth from the sensor to the center point are then obtained.

The clustering of obstacles has three steps:

Step 1: Filtering and denoising. Filtering serves many purposes, such as removing noise points and outliers [27], smoothing point clouds [28], filling holes, and compressing data. Subsequent processing can be conducted more effectively only by filtering out noise and outliers during preprocessing. In this study, a statistical outlier removal (SOR) filter was used for filtering and denoising.

The implementation principle of the SOR filter is as follows: The average distance between a point and its  $k$  nearest neighboring points is calculated, and the shape of the Gaussian distribution depends on the mean distance value  $\mu$  and the standard deviation  $\sigma$ . The distance threshold is then calculated using Eq. (15).

$$d = \mu + A \times \sigma, \quad (15)$$

where  $A$  is a proportionality constant that is determined by the number of neighboring points  $k$ . Finally, points with an average distance greater than  $d$  from their  $k$ -nearest neighbors are removed.

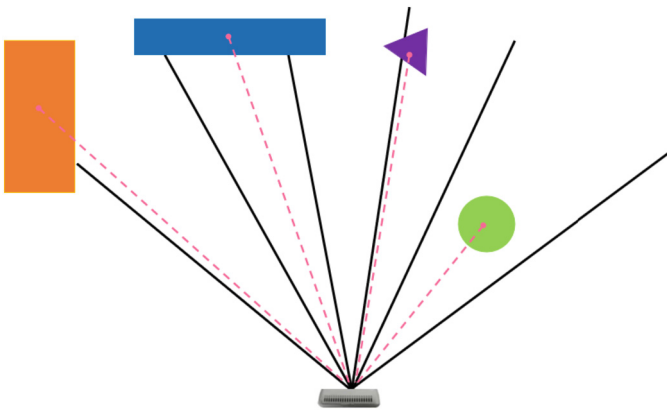
Step 2: Obtain the obstacle point clusters using Euclidean clustering [29]. Obstacles can be separated by performing Euclidean clustering on point clouds, and invalid points can be removed. This algorithm comprises the following steps:

1. Starting from an arbitrary point  $P$  (seed point), search the KD-Tree for its  $k$ -nearest neighbors. If the distance between these neighboring points is less than the set threshold, these points are considered to belong to the same class and are marked.
2. Use the neighboring points as new seed points and repeat the process in (1) until no neighboring points can be found, at which point the points are considered to belong to the same cluster.
3. Find the next unmarked point and repeat processes (1) and (2) until all points have been searched for and marked, at which point the algorithm ends. After performing Euclidean clustering, the original point cloud was divided into multiple clusters, and each cluster was regarded as an obstacle that appeared in front.

Step 3: Determination of the azimuth angle. After performing Euclidean clustering, multiple sets of obstacle point clouds were obtained, and it was necessary to determine their positions in sight. In this study, the horizontal field of view was divided into five directions: left, left front, front, right front, and right. The median method, which uses the midpoint to represent the entire set of point clouds, was employed to accelerate the computation when determining the direction to which the obstacle belongs.

As seen in Fig. 9, the orange obstacle is located in the left direction. Although the purple obstacle occupies the left, left-front, and front directions, it is mainly distributed in the left-front direction and is therefore considered to be in the left-front direction.

At this point, we can obtain the distance and azimuth from the sensor to the obstacle and then output the information to the visually impaired person through the voice system.



**Fig. 9.** Schematic diagram of the obstacle clustering principle.

**Clustering Experiment on Obstacles.** The proposed method was incorporated into an auxiliary system. As seen in Fig. 10, the system comprises a RealSense sensor, computer (NVIDIA GeForce GTX 1650, Ubuntu system), and bone-conduction headset that feeds nonsemantic stereo sound back to the ear. RealSense uses a USB 3.0 interface and computer to transmit data. The computer communicated with the headset using Bluetooth 4.0. A computer was placed in a backpack, a RealSense sensor was placed on the chest, and bone-conducting headphones were attached to each ear. The bone conduction headphones do not clog the user’s ears. Therefore, an auxiliary prototype allows blind people to hear ambient sounds simultaneously. This is very important for assisting the blind in travelling, as they need to maintain awareness of their surroundings to ensure safety.

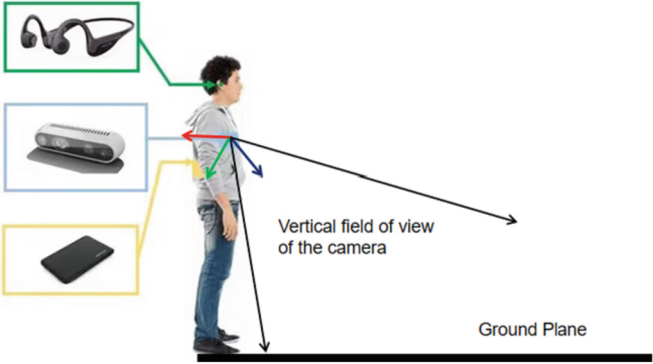


Fig. 10. Schematic diagram of blind wearing

As shown in Fig. 11, this method could effectively segment flat ground in complex environments. The obstacle point cloud is then clustered, and the center point of the obstacle is obtained. Thus, the distance and orientation between the center point of the obstacle and the blind person can be detected. After many experiments with different scenes and comparisons with the actual distance and orientation, the results were accurate, proving the reliability and robustness of the method adopted in this study, and to a certain extent, ensuring the accurate perception of the scene by the guide system.

### 3 Results and Discussion

#### 3.1 Image Data Acquisition

Presently, in various public datasets, such as the COCO dataset, Google Open Image (GOI) dataset, and Intelligent Safety Automotive Laboratory (LISA) traffic sign dataset, traffic lights are mostly in motor lanes. To better adapt to the problems studied, we selected a self-built dataset. To enhance the ability to generalize of the learned model, environmental factors such as sunny, cloudy, and rainy days, days, and nights were also considered in the image acquisition, and different scenes, environments, shooting

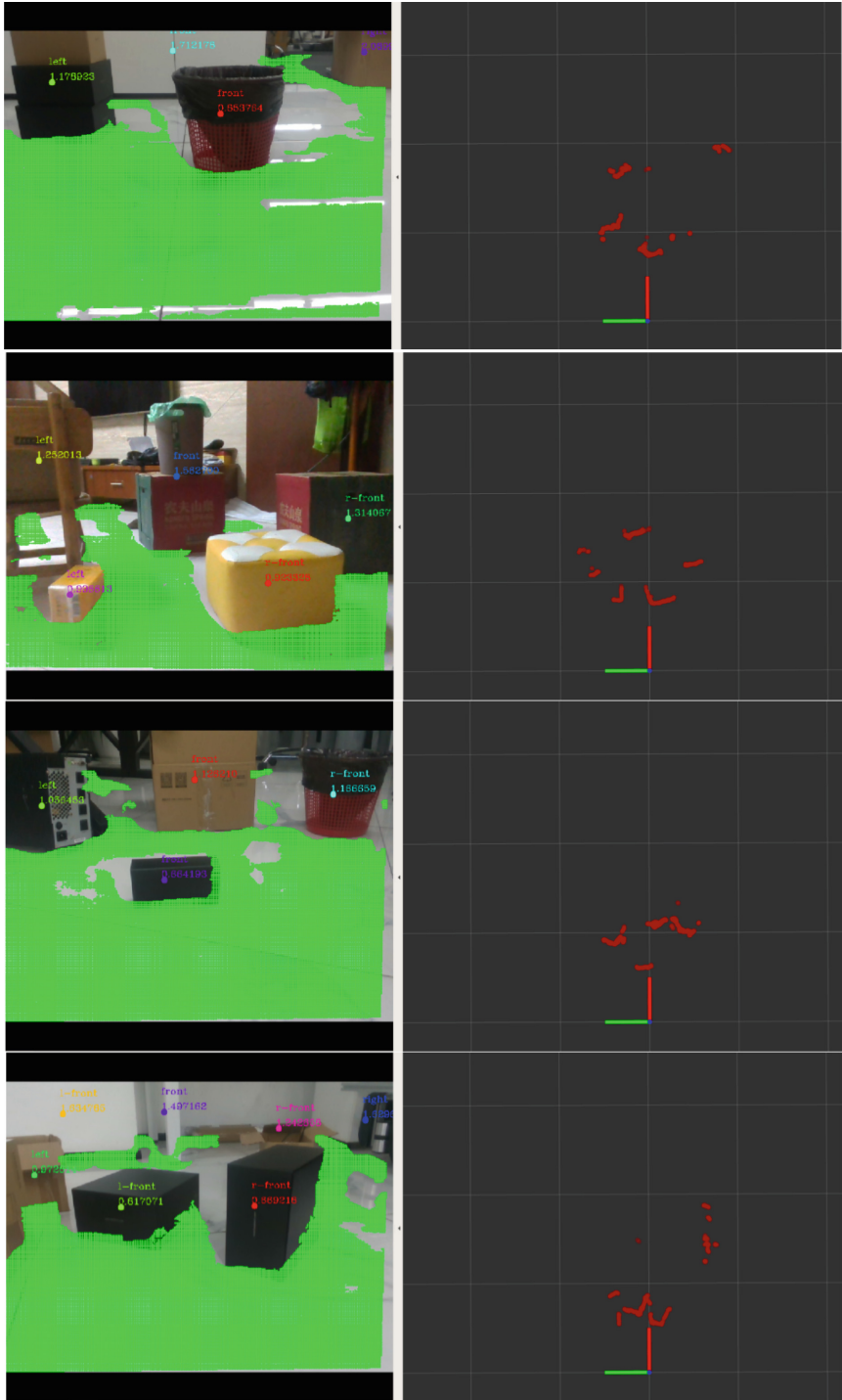


Fig. 11. Obstacle clustering.

distances, backgrounds, and angles were adopted to shoot the sidewalk traffic lights. After manual screening and labelling using Labellmg dataset software, the dataset included 3401 images. The dataset was divided into a test set and a training set in the ratio of 2:8, including two kinds of targets: “green” and “red.”

### 3.2 Model Evaluation Index

In this experiment, an RTX 3060 graphics card was used to train and test the server. Set the number of iterations to 300. The recall, precision, mAP, and model size were used to compare the performance of each model, which was calculated using the following formula.

$$P = \frac{TP}{FP + TP} \quad (16)$$

$$R = \frac{TP}{FN + TP} \quad (17)$$

$$AP = \int_0^1 P(R)dR \quad (18)$$

$$mAP = \frac{1}{m} \sum_{i=1}^m AP_i, \quad (19)$$

where TP is the number of samples correctly predicted as the true class, FN is the number of samples incorrectly predicted as the other class, and FP is the number of samples of the other class incorrectly predicted as the true class.

### 3.3 Comparative Experiment

To better demonstrate the advantages, an experiment was conducted to compare YOLOv5-CM with the faster R-CNN, SSD, YOLOv6 [30], YOLOv7 [30], and YOLOv5s. The models used the same dataset for training and verification.

**Table 1.** Experimental comparison results

	P (%)	R (%)	mAP (%)	Size of Model l. (MB)	Parameters (MB)
Faster R-CNN	57.83	88.7	86.2	110.8	28.5
SSD	98.1	67.6	90.9	93.3	26.3
YOLOv5s	98.6	93.0	95.6	14.4	7.1
YOLOv6s	97.4	93.0	94.8	36.3	17.2
YOLOv7	98.8	91.7	95.4	74.8	36.5
YOLOv5-CM	98.5	92.5	96.3	3.0	1.4

As seen in Table 1, in the YOLOv5 detection algorithm, the YOLOv5s model is a lightweight network model for Faster R-CNN and SSD. Compared with those of the YOLOv5 model, the mAP of the YOLOv5-CM was improved by 0.7%, the size of the model was decreased by 79.1%, and the number of parameters was decreased by 80.3%. The YOLOv5-CM model is better than YOLOv6s and YOLOv7 in terms of the mAP, model size, and parameters, which may also be the reason for the dataset. This shows that, compared with current mainstream detection algorithms, YOLOv5-CM maintains a higher mAP while reducing the size and number of parameters.

### 3.4 Ablation Experiment

An ablation experiment was conducted to verify the optimization of the individual improvement modules. As shown in Table 2, mAP improved by 0.4% when the backbone network was replaced with MobileNetV3. After adding coordinate attention, the mAP improved by 0.7%. The ablation results show that YOLOv5-CM has good performance, achieving the goal of reducing the parameters and the size of the model on the premise of maintaining a good mAP.

**Table 2.** Ablation experimental results

	P (%)	R (%)	mAP (%)	Size of Model (MB)	Parameters (MB)
YOLOv5s	98.6	93.0	95.6	14.4	7.1
YOLOv5-M	97.4	93.3	96.0	3.0	1.4
YOLOv5-CM	98.5	92.5	96.3	3.0	1.4

## 4 Conclusion

This study proposes an improved YOLOv5 algorithm combined with RealSense, ground plane segmentation, and Euclidean clustering algorithms to assist visually impaired individuals in travel.

1. Improved the network structure of YOLOv5. MobileNetv3 was used to replace the main network of YOLOv5s, and coordinate attention was added. Compared with the traditional YOLOv5 target detection algorithm, the mAP was improved by 0.7%, the size of the model was decreased by 79.1%, and the number of parameters was decreased by 80.3%.
2. Based on RealSense and YOLOv5-CM, a comprehensive visual guide system was developed that can identify and measure the distance of sidewalk traffic lights, as well as detect the location and distance of obstacles. We believe that this study has great practical significance for assisting blind people with their travel.

Our future work will accelerate the implementation of the algorithm and deepen the study of guide systems with attitude sensors and GPS positioning systems.

**Acknowledgment.** This work was supported in part by the National University Student Innovation and Entrepreneurship in Training Program of China, the Provincial University Student Innovation and Entrepreneurship in Training Program of China, the Laboratory Opening Project Fund of Huaibei Normal University (No. 2022sykf022), and the Special Needs Project of Huaibei Normal University (No. 2021zlgc147).

## References

1. World Health Organization. <https://www.who.int/zh/news-room/fact-sheets/detail/blindness-and-visual-impairment>. Accessed 13 Oct 2022
2. Munemo, E., Tom, T.: Problems of unemployment faced by visually impaired people. *Greener J. Soc. Sci.* **3**(4) (2013). <https://doi.org/10.15580/GJSS.2013.4.020713437>
3. Daily Life Problems Faced by Blind People, In Daily Life Problems Faced by Blind People. <https://wecapable.com/problems-faced-by-blind-people/>. Accessed 25 Feb 2021
4. Challenges blind people face when living life. In Challenges blind people face when living life. <https://www.letsenvision.com/blog/challenges-blindpeople-face-when-living-life>. Accessed 25 Feb 2021
5. Guide Dogs of America, Frequently Asked Questions, Guide Dogs of America, <https://guidedogsofamerica.org/about-gda/frequently-asked-questions/>. Accessed 28 Sept 2022
6. Tapu, R., Mocanu, B., Zaharia, T.: A computer vision system that ensure the autonomous navigation of blind people. In: E-Health and Bioengineering Conference 2013, pp. 1–4 (2013). <https://doi.org/10.1109/EHB.2013.6707267>
7. Rodríguez, A., Yebes, J.J., Alcantarilla, P.F., Bergasa, L.M., Almazán, J., Cela, A.: Assisting the visually impaired: obstacle detection and warning system by acoustic feedback. *Sensors* **12**(12) (2012). <https://doi.org/10.3390/s121217476>
8. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **28** (2015)
9. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: Ssd: Single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-46448-0\\_2](https://doi.org/10.1007/978-3-319-46448-0_2)
10. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016). <https://doi.org/10.1109/CVPR.2016.91>
11. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7263–7271, (2017). <https://doi.org/10.1109/CVPR.2017.690>
12. Redmon, J., Farhadi, A.: YOLOv3: An incremental improvement, arXiv preprint [arXiv:1804.02767](https://arxiv.org/abs/1804.02767) (2018)
13. Bochkovskiy, A., Wang, C.Y., Liao, H.: YOLOv4: Optimal speed and accuracy of object detection, arXiv preprint [arXiv:2004.10934](https://arxiv.org/abs/2004.10934) (2020)
14. Hou, Q., Zhou, D., Feng, J.: Coordinate attention for efficient mobile network design. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13713–13722, (2021). <https://doi.org/10.1109/CVPR46437.2021.01350>



15. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141, (2018). <https://doi.org/10.1109/CVPR.2018.00745>
16. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: CBAM: convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV), pp. 3–19, (2018). [https://doi.org/10.1007/978-3-030-01234-2\\_1](https://doi.org/10.1007/978-3-030-01234-2_1)
17. Howard, A., et al.: Searching for MobileNetV3. In: IEEE/CVF International Conference on Computer Vision (ICCV) 2019, pp. 1314–1324 (2019). <https://doi.org/10.1109/ICCV.2019.00140>
18. Wang, W., Li, Y.T., Zou, T., et al.: A novel image classification approach via dense- mobilenet models. *Mob. Inf. Syst.* 1–8 (2020). <https://doi.org/10.1155/2020/7602384>
19. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510–4520, (2018). <https://doi.org/10.1109/CVPR.2018.00474>
20. Ramachandran, P., Zoph, B., Le, Q.V.: Searching for activation functions, ArXiv. preprint (2017)
21. Kumar, S.S., Likhachev, S., Kaess, M.F.: PointNetLK: Robust & efficient point cloud registration using PointNet, [arXiv:1908.09852](https://arxiv.org/abs/1908.09852) [cs] (2019)
22. Dai, Q., Lee, Y., Liao, H.: Deformation analysis of shape memory alloys using structured light and 3D point cloud registration. *Int. J. Adv. Manufact. Technol.* **106**(5–6), 1697–1713 (2020)
23. Zhang, Z.: A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(11), 1330–1334 (2000). <https://doi.org/10.1109/34.888718>
24. Song, S., Xiao, J.: Deep sliding shapes for a modal 3D object detection in RGB-D images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 808–816, (2016). <https://doi.org/10.1109/CVPR.2016.94>
25. Derpanis, K.G.: Overview of the RANSAC algorithm. *Image Rochester NY* **4**(1), 2–3 (2010)
26. Yang, K., Wang, K., Hu, W., Bai, J.: Expanding the detection of traversable area with RealSense for the visually impaired. *Sensors* **16**(11), 1954 (2016). <https://doi.org/10.3390/s16111954>
27. de Cheveigné, A., Simon, J.Z.: Denoising based on spatial filtering. *J. Neurosci. Methods* **171**(2), 331–339 (2008). <https://doi.org/10.1016/j.jneumeth.2008.03.015>
28. Mederos, B., Velho, L., de Figueiredo, L.H.: Robust smoothing of noisy point clouds. In: Proceeding SIAM Conference on Geometric Design and Computing, vol. 2004, no. 1, p. 2, Philadelphia, PA, USA: SIAM (2003)
29. Sun, Z., Li, Z., Liu, Y.: An improved lidar data segmentation algorithm based on Euclidean clustering. In: Wang, R., Chen, Z., Zhang, W., Zhu, Q. (eds.) Proceedings of the 11th International Conference on Modelling, Identification and Control (ICMIC2019). LNEE, vol. 582, pp. 1119–1130. Springer, Singapore (2020). [https://doi.org/10.1007/978-981-15-0474-7\\_105](https://doi.org/10.1007/978-981-15-0474-7_105)
30. Li, C., et al.: YOLOv6: A single-stage object detection framework for industrial applications. arXiv preprint [arXiv:2209.02976](https://arxiv.org/abs/2209.02976) (2022)
31. Wang, C.Y., Bochkovskiy, A., Liao, H.Y.M.: YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, 7464–7475. arXiv preprint [arXiv:2207.02696](https://arxiv.org/abs/2207.02696) (2022)