Vehicular Abandoned Object Detection Based on VANET and Edge AI in Road Scenes

Gang Wang, Mingliang Zhou, Xuekai Wei, and Guang Yang, *Senior Member, IEEE*

***Abstract*— Rapid processing of abandoned objects is one of the most important tasks in road maintenance. Abandoned object detection heavily relies on traditional object detection approaches at a fixed location. However, detection accuracy and range are still far from satisfactory. This study proposes an abandoned object detection approach based on vehicular ad-hoc networks (VANETs) and edge artificial intelligence (AI) in road scenes. We propose a vehicular detection architecture for abandoned objects to achieve task-based AI technology for large-scale road maintenance in mobile computing circumstances. To improve detection accuracy and reduce repeated detection rates in mobile computing, we propose a detection algorithm that combines a deep learning network and a deduplication module for high-frequency detection. Finally, we propose a location estimation approach for abandoned objects based on the World Geodetic System 1984 (WGS84) coordinate system and an affine projection model to accurately compute the positions of abandoned objects. Experimental results show that our proposed algorithm achieves an average accuracy of 99.57% and 53.11% on the two datasets, respectively. Additionally, our whole system achieves real-time detection and high-precision localization performance on real roads.**

***Index Terms*— Abandoned objects, object detection, VANET, deep learning, deduplication module.**

Manuscript received 7 September 2022; revised 22 February 2023 and 2 May 2023; accepted 12 July 2023. Date of publication 28 July 2023; date of current version 29 November 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62176027; in part by the China Scholarship Council (CSC) under Grant 202008330537; in part by the Ningbo Natural Science Foundation under Grant 2023J280; in part by the Ningbo Key Research and Development Program, Digital Twin Project; in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LQ20F020006; in part by the General Program of the National Natural Science Foundation of Chongqing under Grant cstc2020jcyj-msxmX0790; in part by the Human Resources and Social Security Bureau Project of Chongqing under Grant cx2020073; in part by the Guangdong Oppo Mobile Telecommunications Corporation Ltd., under Grant H20221694; and in part by the UK Research and Innovation (UKRI) Future Leaders Fellowship under Grant MR/V023799/1. The Associate Editor for this article was I. U. Din. *(Corresponding author: Mingliang Zhou.)*

Gang Wang is with the School of Computing and Data Engineering, NingboTech University, Ningbo 315100, China, and also with the Chongqing Key Laboratory of Image Cognition, School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: smile588@sina.com).

Mingliang Zhou is with the School of Computer Science, Chongqing University, Chongqing 400044, China (e-mail: mingliangzhou@cqu.edu.cn).

Xuekai Wei is with the State Key Laboratory of Internet of Things for Smart City, University of Macau, Macau, China, and also with the School of Computer Science, Chongqing University, Chongqing 400044, China (e-mail: xuekaiwei2-c@my.cityu.edu.hk).

Guang Yang is with the Cardiovascular Research Centre, Royal Brompton Hospital, SW3 6NP London, U.K., and also with the National Heart and Lung Institute, Imperial College London, SW7 2AZ London, U.K. (e-mail: g.yang@imperial.ac.uk).

Digital Object Identifier 10.1109/TITS.2023.3296508

I. Introduction

THE maintaining roads is crucial to ensure the safety and comfort of drivers and their vehicles [1], [2], [3]. In addition to potholes and cracks, abandoned objects have become a new road disease that poses a serious threat to road traffic safety [4]. These objects have led to a surge in traffic accidents, as drivers have been unable to avoid them. As a result, it is of utmost importance to identify and remove abandoned objects from roads.

Over the past few decades, abandoned object detection has advanced significantly, thanks to the rapid development of object detection and classification analysis in the computer vision field [5], [6]. Some traditional object detection methods have been applied for abandoned object detection.Traditional methods [7], [8], such as background modeling [9], [10], [11], have been used for abandoned object detection, but they heavily rely on background modeling algorithms and are more suited for static scenes.

With the rise of deep learning approaches, many outstanding deep learning network models, such as VGG [12], ResNet [13], and RNN [14], have been developed and applied successfully in many common scenes, such as face recognition and license plate recognition. However, deep learning approaches require vast amounts of data and computing power to achieve better performance [15].

One of the main challenges in abandoned object detection in road scenes is achieving high-accuracy detection and high-precision localization in mobile circumstances [16]. Traditional abandoned object detection methods are usually integrated into the intelligent vision system on a static location, thus limiting their scope [17], [18], [19]. Moreover, the computing architecture of traditional detection is unbalanced for its centralized computation, leading to high communication and computation costs [20].

To overcome these challenges, vehicular ad-hoc networks (VANETs) offer a suitable solution for abandoned object detection in road scenes [21], [22]. The addition of complementary communication techniques [23], such as Long Term Evolution (LTE) and Super Wi-Fi, and the utilization of Mobile Edge Computing (MEC) technology can effectively solve the problems of capacity, computation, and latency in traditional detection architecture [24], [25]. By deploying AI models on edge devices, VANETs and edge AI can achieve high-accuracy detection and high-precision localization for abandoned objects in mobile circumstances.

In summary, this paper addresses the problem of abandoned object detection in mobile circumstances and proposes the use of VANETs and edge AI to overcome the challenges of traditional detection methods. Our study makes three key scientific contributions.

- Firstly, we propose a vehicular detection architecture for abandoned objects, which comprises mobile hardware module components and a detection algorithm that utilizes a deep learning model. Our architecture allows for all detection computations to be completed in a vehicular edge computing AI device by combining VANET, making it a task-based AI technology that can significantly improve efficiency in detecting and computing for a new application scenario.
- Secondly, we propose a detection algorithm that combines a deep learning network with a deduplication module for high-frequency detection computation in mobile computing circumstances. Unlike prior related works that focus on traditional image processing approaches for road surfaces, our algorithm considers duplicate detection, which is often not considered in mobile computing circumstances. To train our deep learning network, we establish two abandoned object datasets by collecting images from the Internet and marking all images with typical abandoned object classes in road scenes. This is a significant contribution, as there are currently no abandoned object datasets for deep learning approaches in existing works.
- Lastly, to accurately compute the positions of abandoned objects, we propose a location estimation approach based on the World Geodetic System 1984 (WGS84) coordinate system and an affine projection model. Our approach can be overlaid into a map service and helps reduce localization errors, which can be a significant issue in prior related works.

The rest of this paper is organized as follows. In Section II, we provide a brief review of related works. Section III presents an overview of our proposed vehicular detection approach. In Section IV, we describe our high-frequency detection computation for mobile computing circumstances. In Section V, we detail our location estimation approach for abandoned objects in WGS84. Section VI presents our experimental results. Finally, Section VII concludes our work.

II. Related Works

Traditional abandoned object detection methods face challenges with detection accuracy, object localization, and detection scope. Developing a vehicular detection approach combining VANET and edge AI can help address these limitations and improve the effectiveness of abandoned object detection.

A. Traditional Abandoned Object Detection

Traditional abandoned object detection approaches mainly adopt background modeling methods to generate the background picture. Then, these approaches extract foreground regions by comparing current frames with the background picture. These approaches depend heavily on background modeling algorithms. Representative background modeling algorithms include GMM, Vibe, Codebook, etc. These algorithms are suitable for static scenes.

After foreground regions are extracted from pictures, common traditional object detection algorithms will further extract features for classification. Common traditional object detection algorithms have many applications for computer vision tasks. The classification steps are mainly classical feature extraction first and then combining some features classifiers for detection [26], [27], [28]. However, traditional object detection algorithms suffer from many drawbacks. There exist problems with region selection strategies, high time complexity, window redundancy, and hand-designed features.

B. Deep Learning-Based Detection

With the rapid development of deep learning, more powerful tools have been introduced, which are capable of learning semantic, high-level, and more in-depth features [29]. Particularly, automatically detecting objects in images and videos can be satisfactory.

For the processing stage of deep learning detection, there are two main categories: One-stage and Two-stage. One-stage is a direct regression of class probability and location coordinate values of an object. However, accuracy is low, and speed is fast. The classical one-stage object detection networks are YOLO [30], [31], [32], [33] and SSD [34].

Another category of deep learning detection algorithms is Two-stage. These algorithms first generate a series of candidate frames as samples, and then classify these samples by convolutional neural networks. The classical Two-stage object detection networks are R-CNN [35], Fast R-CNN [36], Faster R-CNN [37], and Mask R-CNN [38].

One-stage network is more suitable for real-time mobile computation for its fast computation speed. Therefore, many improved YOLO series networks are introduced for mobile circumstances. Some works have been proposed to use One-stage network. Researchers proposed a complex YOLO based on YOLOv2 3D object detection for autonomous driving [39]. Choi et al. [40] proposed a Gaussian YOLOv3 model based on localization uncertainty for autonomous driving. Cai et al. [41] proposed YOLOv4-5D for autonomous driving. Aboah et al. [42] proposed a YOLOv5-based deep learning decision tree traffic anomaly detection system. Sarmiento et al. [43] proposed a pavement breakage detection and segmentation scheme on road surfaces using YOLOv4 and DeepLabv3.

Benefiting from the high-accuracy performance of deep learning-based object detection, deployment of AI models on edge devices in vehicles will improve the accuracy of abandoned object detection. In this paper, we propose to achieve a vehicular abandoned object detection approach based on VANET and edge AI in road scenes. In contrast to existing methods that mainly focus on the improvement of detection performance at a fixed location, we incorporate VANET and edge AI into the vehicular abandoned object detection in mobile computing circumstances to improve the accuracy and the range of the detection for road maintenance.

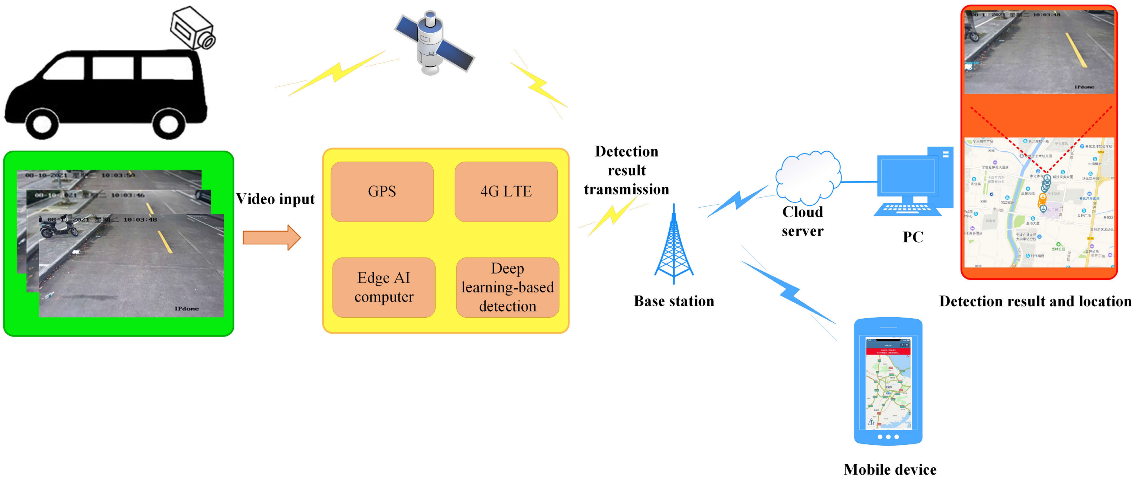

Video input
GPS
4G LTE
Edge AI computer
Deep learning-based detection
Detection result transmission
Base station
Cloud server
PC
Detection result and location
Mobile device

Fig. 1. Overview of the proposed vehicular detection approach.

III. Overview of Proposed Approach

The proposed vehicular detection approach can be seen in Fig.1. Our proposed approach can recognize and localize abandoned objects on city roads. The whole architecture includes four types of devices:

1) Vehicular vision capturing device. An industrial vehicular camera, called HIKVISION iDS-TCC225, is used in our approach. The camera is two mega pixels, which is 1/2.8” Progressive Scan CMOS. The camera supports two frame rates: 50HZ:25fps(1920 × 1080) and 60HZ:30fps(1920 × 1080). The optic of the camera is 30x optical zoom with 65.5-2.11 degrees, which can capture a lane of around 2-3m.

2) Vehicular edge AI device. To achieve rapid processing for video streams on a vehicular edge, our approach adopts an industrial computer, called ADVANTECH MIC-7700, with antiseismic and GPU computational characteristics as our vehicular edge AI device. The industrial computer provides intel CPU i7-6700TE, NVIDA GPU Geforce RTX 2080TI, dual network port, and Windows 7 pro operation system. Therefore our vehicular edge AI device can support decoding video streams and deep learning-based detection for edge AI tasks.

3) Vehicular Geo-location device. Our approach uses a dual antenna RTK inertial integrated navigation system, which integrates industrial MEMS gyroscope, accelerometer and dual frequency GNSS receiver [44]. Through an embedded multi-sensor fusion algorithm and a full temperature domain calibration method, the navigation system can output stable and continuous location, direction, speed and altitude information in complex environments (elevated, underground garage, tunnel, urban road, port, tree shelter, etc.). This device supports access to RTK differential signal to achieve centimeter-level high-precision localization and 0.15° orientation accuracy through dual antennas. Through the integrated navigation fusion algorithm, it can be better than 0.4m when satellites are out of lock for 10 seconds.

4) Communication and accessible system. Our communication system uses a 4G-LTE route as a gateway to achieve a rapid transmission of detection results [45]. Its processor is powerful enough to handle full ranges of LTE communications capabilities, including video streams. Its internal memory provides ample storage for custom scripts, software applications and a wide variety of protocols.

To achieve the detection for abandoned objects, we integrate four parts: 1) the vehicular vision capturing part, 2) the vehicular edge AI part, 3) the vehicular Geo-location part, and 4) the communication and accessible system. All parts of implementation in our approach use C++, JAVA, and Python. Moreover, some classic libraries (e.g., OpenCV, et al.) are adopted in our implementation.

IV. Mobile Detection Algorithm for High-Frequency Computation

A. Algorithm Framework

Fig.2 shows our deep learning-based detection algorithm. We use the YOLOv5 module as a basic detection network module for its simple deployment and stability. Particularly, we design a deduplication module for high-frequency detection computation in mobile computing circumstances. The YOLO network includes three parts: the backbone, neck, and head. The backbone is a CNN network that could combine different features of multi-scale images. The neck is the network layer that could combine image features and transit image features to the predicted layer. The head is the predicted layer which can predict bounding boxes and classes. YOLOv5 uses CSPDarknet53 as its backbone framework. Its neck includes PANET and SPP. Its head network is from YOLOv3. Also, YOLOv5 is different from YOLOv4. YOLOv5 designs two CSP structure network CSP1_X for the backbone and CSP2_X for the neck. However, YOLOv4 designs only a CSP structure network, which can be used in the backbone.

Moreover, YOLOv5 uses generalized intersection over union (GIoU) loss as its loss function, which can be formulated by

$$\ell_{GIOU} = 1 - GIOU \tag{1}$$

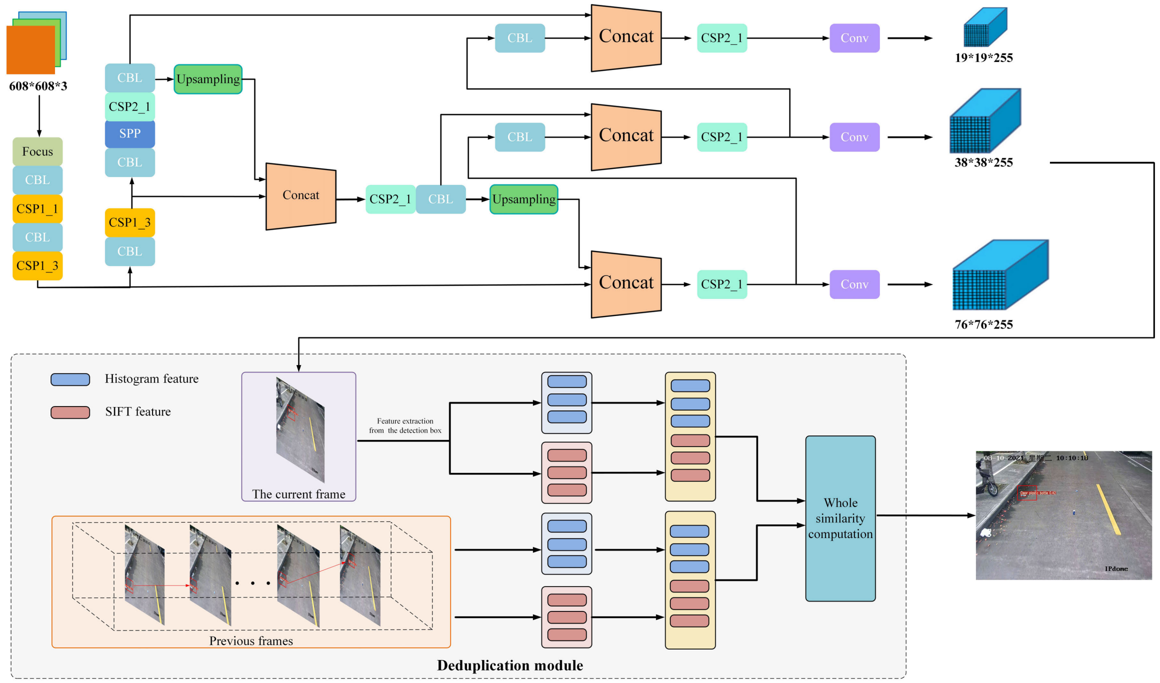


Fig. 2. Framework of our method.

B. Deduplication Module

Because high-frequency detection would result in extensive repeated detection for the same object in mobile computing circumstances, we propose a deduplication module for high-frequency detection computation. In our proposed deduplication module for high-frequency detection computation, we extract two features of co-located GIoU areas between continuous frames. One feature is the histogram feature, which has a color similarity. The histogram similarity computation of GIoU areas can be deduced as follows:

$$h(G, S) = \sum_{i=1}^{N} \left(1 - \frac{|g_i - s_i|}{Max\,(g_i, s_i)}\right) \tag{2}$$

where N represents the pixel sample number, g_i and s_i represent the co-located GIoU area between the continuous frames G and S, respectively.

To consider angles and deformation, we introduce the SIFT feature as the other feature. The SIFT similarity computation of GIoU areas can be given as follows:

$$d(G, S) = \frac{1}{m} \sum_{n=1}^{m} \left(\min_{1 \le j \le n} \sqrt{\sum_{k=1}^{128} \left(g_{ik} - s_{jk}\right)^2} \right) \tag{3}$$

where m is the SIFT feature number of G, n is the SIFT feature number of S, g_{ik} is the k-th dimension in the i-th feature vector, and s_{jk} is the k-th dimension in the j-th feature vector.

To obtain a whole similarity of co-located GIoU areas between continuous frames, we can utilize a weighted fusion method to combine the histogram similarity and the SIFT similarity, which can be given by

$$\zeta(G, S) = \gamma h(G, S) + \lambda d(G, S) \quad \text{s.t.} \quad \gamma + \lambda = 1 \tag{4}$$

where γ and λ represent weight values, $h(G, S)$ represents the histogram similarity, and $d(G, S)$ represents the SIFT similarity.

In our actual experiment, we set $\gamma = 0.5$ and $\lambda = 0.5$ to compute the whole similarity ζ in Eq.(4). Also, we set $\zeta = 0.7$ as a similarity threshold between continuous frames. When the whole similarity is greater than the similarity threshold, the detection area would be the same. At last, the prediction box will vanish. The procedure of the proposed detection algorithm is given in Algorithm 1.

V. Location Estimation for Abandoned Objects

A. Model Hypothesis

The high-precision locations of abandoned objects would process promptly abandoned objects on roads to reduce traffic accidents. So it is significant for our detection approach to obtain high-precision positions of abandoned objects. To accurately compute positions of abandoned objects, the proposed location estimation approach for abandoned objects is based on the WGS84 coordinate system and our affine projection model, which is shown in Fig.3. The WGS84 coordinate system is used by the GPS satellite navigation system, and can overlay into a map service. In the proposed location estimation approach, the affine projection model is proposed to decrease the localization error and rapidly correct the location.

B. Affine Projection Modeling

To obtain the real-world location of abandoned objects captured from our vision device, the affine projection model is proposed to establish the relationship between the real world and video frames. In the proposed affine projection model, two factors should be considered. One factor is the vision-capturing device's height. The other is the lens distortion of

Algorithm 1 Detection Box Computational Procedure in Mobile Computing

```
Input: Prediction box $B^p$;
       Ground truth box $B^g$.
Output: Unrepeated detection box $B^*$.
Initialization:
$B^p = (x_1^p, y_1^p, x_2^p, y_2^p)$ , $B^g = (x_1^g, y_1^g, x_2^g, y_2^g)$
while True do
    $\hat{x}_1^p = min(x_1^p, x_2^p)$ , $\hat{x}_2^p = \max(x_1^p, x_2^p)$ ,
     $\hat{y}_1^p = min(y_1^p, y_2^p)$ , $\hat{y}_2^p = \max(y_1^p, y_2^p)$ ;
    $\hat{B}^p = (\hat{x}_1^p, \hat{y}_1^p, \hat{x}_2^p, \hat{y}_2^p)$ ensuring $x_2^p > x_1^p$ and
     $y_2^p > y_1^p$.
    Compute area of $B^g$ and $\hat{B}^p$:
    $A^g = (x_2^g - x_1^g) * (y_2^g - y_1^g)$ ,
    $A^p = (\hat{x}_2^p - \hat{x}_1^p) * (\hat{y}_2^p - \hat{y}_1^p)$.
    if max $(\hat{x}_1^p, x_1^g)$ > max $(\hat{x}_2^p, x_2^g)$ and
     max $(\hat{y}_1^p, y_1^g)$ > max $(\hat{x}_2^p, x_2^g)$ then
        $x_1^\Psi = \max(\hat{x}_1^p, x_1^g)$ , $x_2^\Psi = \max(\hat{x}_2^p, x_2^g)$,
         $y_1^\Psi = \max(\hat{y}_1^p, y_1^g)$ , $y_2^\Psi = \max(\hat{y}_2^p, y_2^g)$.
        Compute intersection $\psi$ between $\hat{B}^p$ and $B^g$:
         intersection: $\Psi = (x_2^\Psi - x_1^\Psi) * (y_2^\Psi - y_1^\Psi)$.
    else
        Compute intersection $\psi$ between $\hat{B}^p$ and $B^g$:
         intersection: $\Psi = 0$.
    end if
    Finding the coordinate of smallest enclosing box:
     $B^c : x_1^c = min(\hat{x}_1^p, x_1^g)$ , $x_2^c = \max(x_2^p, x_2^g)$ ,
    $y_1^c = min(\hat{y}_1^p, y_1^g)$ , $y_2^c = \max(\hat{y}_2^p, y_2^g)$ .
    Compute area of $B^c$ : $A^c = (x_2^c - x_1^c) * (y_2^c - y_1^c)$ .
    $IoU = \frac{\Psi}{\mu}$, where $\mu = A^p + A^g - \Psi$.
    $GIoU = IoU - \frac{A^c - \mu}{A^c}$.
    Compute $h(G, S)$,$d(G, S)$,$\zeta(G, S)$ using Eq.(2),
     Eq.(3), Eq.(4), respectively.
if $\zeta(G, S) > threshold$ then
    Generate $B^*$.
end if
end while
```

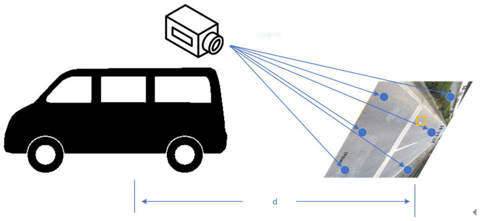


Fig. 3. Location estimation for abandoned objects.

CCD in the vision-capturing device. To calculate a real point location (x', y') of an abandoned object for a pixel at (x, y) in a video frame, this affine projection model can be expressed by

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix} \tag{5}$$

where S_x and S_y represent scale factors, T_x and T_y represent translation factors.

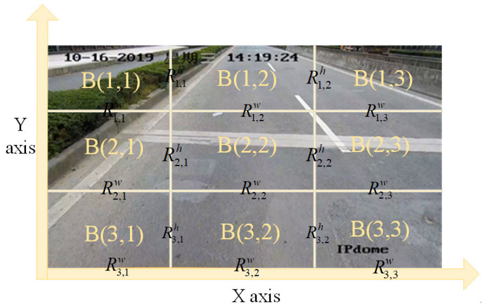


Fig. 4. Grid quantization approach.

To efficiently compute projection parameters, we propose a grid quantization approach in Fig.4. In our approach, we will divide a video frame into 3*3 grid blocks to reduce the error of distortion. H^p and W^p are the height and width of the video frame, respectively. H^c and W^c are the height and width of the real world, respectively. Each grid block in Fig.4 has a pair of affine parameters (R^h, R^w) as distortion correction of the location. The affine parameters $(R^h_{i,j}, R^w_{i,j})$ of a grid block can be expressed by

$$\begin{cases} R^h_{i,j} = \frac{H^p_{i,j}}{H^c_{i,j}} \\ R^w_{i,j} = \frac{W^p_{i,j}}{W^c_{i,j}} \end{cases} \quad 1 \le i \le 3, 1 \le j \le 3 \tag{6}$$

where $R^h_{i,j}$ represents the height ratio parameter of the (i, j)-th block, $R^w_{i,j}$ represents the weight ratio parameter of the (i, j)-th block, $H^p_{i,j}$ represents the height of the (i, j)-th block in the video frame, $H^c_{i,j}$ represents the height of the (i, j)-th block in the real world, $W^p_{i,j}$ represents the height of the (i, j)-th block in the video frame, and $W^c_{i,j}$ represents the height of the (i, j)-th block in the real world.

By using the typical checkerboard calibration method in experiments considering the camera position and lens distortion, we can obtain an affine coefficient matric as follows:

$$C^h_R = \begin{bmatrix} 2.7 & 2.7 & 2.7 \\ 1.55 & 1.55 & 1.55 \\ 1.1 & 1.1 & 1.1 \end{bmatrix}$$
$$C^w_R = \begin{bmatrix} 5.2 & 5.2 & 5.2 \\ 2.8 & 2.8 & 2.8 \\ 1.35 & 1.35 & 1.35 \end{bmatrix} \tag{7}$$

where C^h_R and C^w_R represent the height ratio coefficient and weight ratio coefficient for affine parameters $(R^h_{i,j}, R^w_{i,j})$ in nine gird blocks in a frame, respectively.

Then, the scale factors of each grid block can be formulated by

$$\begin{aligned} S_x &= R^h_{i,j} * x \quad x \in B(i, j) \\ S_y &= R^w_{i,j} * y \quad y \in B(i, j) \end{aligned} \tag{8}$$

where $B(i, j)$ represents the (i, j)-th block in a frame.

Similarly, the translation factors of each grid block can be formulated by

$$T_x = \sum_{i,j} \tau_{i,j} * R^w_{i,j} * W^p_{i,j} \quad x \in B(i,j)$$
$$T_y = \sum_{i,j} \lambda_{i,j} * R^h_{i,j} * H^p_{i,j} \quad y \in B(i,j) \tag{9}$$

where $\tau_{i,j}$ is the coefficient factor of the X-axis, $\lambda_{i,j}$ is the coefficient factor of the y-axis, $W^p_{i,j} = \varepsilon * W^p$ is the width of the (i,j)-th block with $\varepsilon = 1/3$, $H^p_{i,j} = \kappa * h^p$ with $\kappa = 1/3$.

By combining coefficient factors, Eq.(9) can be further described with

$$T_x(i,j) = \begin{bmatrix} 0 & R^w_{1,1} * W^p_{1,1} - R^w_{1,2} * W^p_{1,2} & R^w_{1,1} * W^p_{1,1} + R^w_{1,2} * W^p_{1,2} - R^w_{1,3} * W^p_{1,3} \\ 0 & R^w_{1,1} * W^p_{1,1} - R^w_{1,2} * W^p_{1,2} & R^w_{1,1} * W^p_{1,1} + R^w_{1,2} * W^p_{1,2} - R^w_{1,3} * W^p_{1,3} \\ 0 & R^w_{1,1} * W^p_{1,1} - R^w_{1,2} * W^p_{1,2} & R^w_{1,1} * W^p_{1,1} + R^w_{1,2} * W^p_{1,2} - R^w_{1,3} * W^p_{1,3} \end{bmatrix}$$
$$T_y(i,j) = \begin{bmatrix} R^h_{3,1} * h^p_{3,1} + R^h_{2,1} * h^p_{2,1} - 2 * R^h_{1,1} * h^p_{1,1} & R^h_{3,1} * h^p_{3,1} + R^h_{2,1} * h^p_{2,1} - 2 * R^h_{1,1} * h^p_{1,1} & R^h_{3,1} * h^p_{3,1} + R^h_{2,1} * h^p_{2,1} - 2 * R^h_{1,1} * h^p_{1,1} \\ R^h_{3,1} * h^p_{3,1} - R^h_{2,1} * h^p_{2,1} & R^h_{3,1} * h^p_{3,1} - R^h_{2,1} * h^p_{2,1} & R^h_{3,1} * h^p_{3,1} - R^h_{2,1} * h^p_{2,1} \\ 0 & 0 & 0 \end{bmatrix} \tag{10}$$

Utilizing a satellite heading angle and Eq.(5), the longitude and latitude of the abandoned object can be given by

$$x'' = \phi * \cos\alpha * d + x'$$
$$y'' = \varphi * \sin\alpha * d + y' \tag{11}$$

where α represents the satellite heading angle, d represents the distance between the GPS device and the camera field of view, x' represents a real point location of an abandoned object region for a corresponding pixel at x in a video frame, y' represents a real point location of an abandoned object region for a corresponding pixel at y in a video frame, and ϕ and φ represent the longitude and latitude of the GPS device. respectively. Summarily, our proposed location approach is given by Algorithm 2.

VI. Experiments

A. Datasets and Training Set

Establishing datasets is fundamental to train and verify our proposed detection algorithm. Because existing datasets for abandoned object detection can not be found, we consider several garbage images from the Internet to establish our experimental datasets. Two public garbage datasets are collected from the Internet. These two datasets are TACO dataset [46] and Xi'an AI competition dataset [47].

Algorithm 2 Proposed Location Approach for Abandoned Objects in WGS84

```
Input: Location $P(x, y)$ of a detection box B in the video frame.
Output: Location $P(x'', y'')$ of an abandoned object in WGS84.
for each block do
    Compute $(x, y)$ using Eq.(5);
    Compute $S_x$ and $S_y$ using Eq.(8);
    Compute $T_x$ and $T_y$ using Eq.(9);
    Compute $W^p_{i,j} = \varepsilon * W^p$ with $\varepsilon = 1/3$,
      $H^p_{i,j} = \kappa * h^p$ with $\kappa = 1/3$;
    Compute $P(x'', y'')$ using Eq.(11);
    Return $P(x'', y'')$.
End For
```

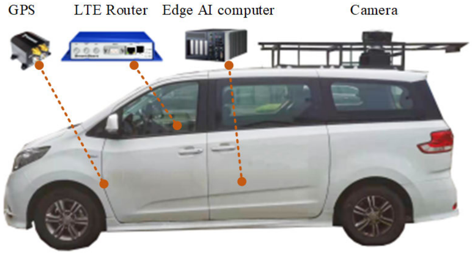


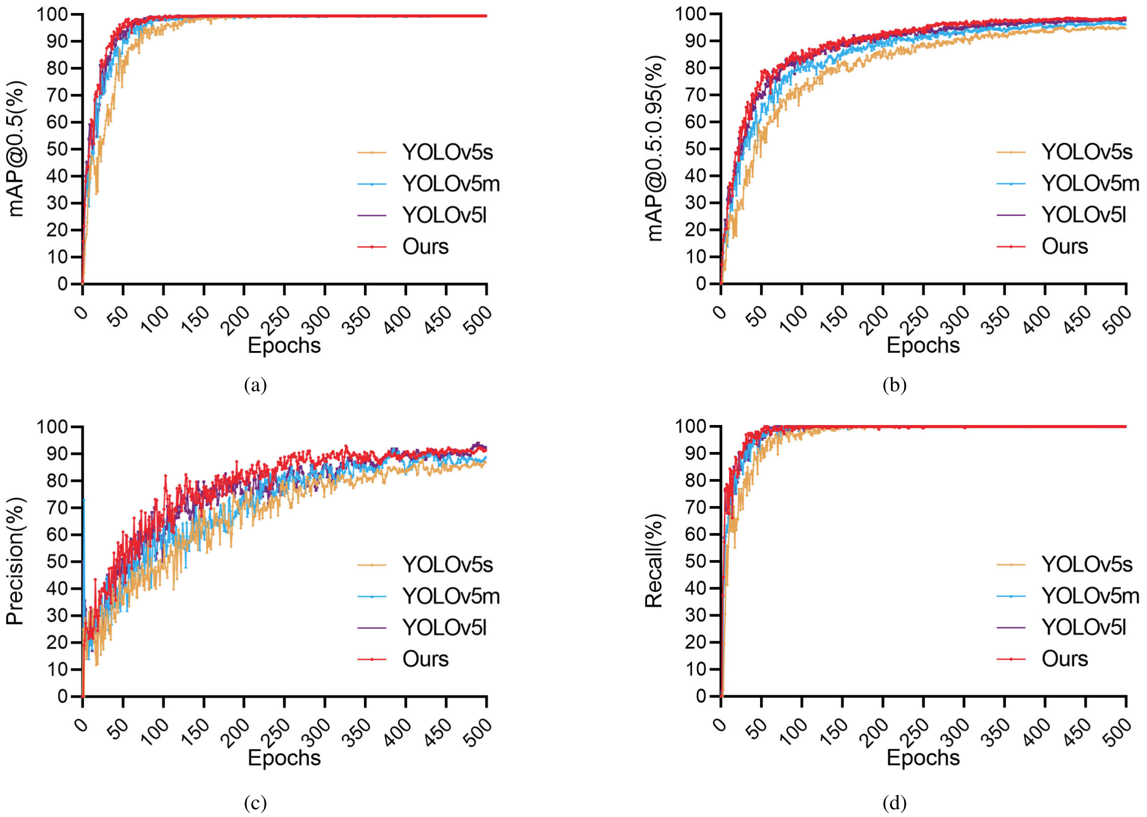
Fig. 5. Vehicular devices setting.

1) TACO Dataset: The dataset is a growing image dataset of waste in the wild. It contains 1500 images with 60 classes of garbage objects. There are diverse environments: woods, roads and beaches. These images are manually labeled and segmented according to a hierarchical taxonomy to train and evaluate object detection algorithms.

2) Xi'an AI Competition Dataset: The dataset is an AI competition dataset for municipal waste sorting collection. It contains about 10000 images. There are six classes: Bottle, Cloth, Kitchen Waste, Metal, Paper, Plastic.

Because of their label format and sample imbalance, these two datasets can not be used directly for training and testing. We select firstly those images under the road environment from the TACO dataset. So we establish a road abandoned object dataset called TACOsub as our first experimental dataset. TACOsub contains 1194 images for road scenes with four classes. We re-label 4 classes: Clear plastic bottle, Drink can, Other plastic and Plastic. This can support more deep learning models. For the Xi'an AI competition dataset, we re-label six classes and balance its quantity of each class. Thus we establish a more reasonable dataset called AIcomp dataset. AIcomp dataset contains 1200 images with six classes. Each class has 200 sample images.

In the training stage, our proposed algorithm will be trained on our two datasets (TACOsub and AIcomp). We divide each dataset into a train set, a validation set, and a test set by 8:1:1. To ensure training and performance independence, we train our model on each dataset. Therefore we can effectively evaluate the gain of sample data for our model.



(a)

(b)

(c)

(d)

Fig. 6. Comparison of training detailed curves for four methods on the TACOsub dataset: (a) YOLOv5s, (b) YOLOv5m, (c) YOLO5l, and (d) Ours.

B. Experimental Setting

We adopt common indicators to evaluate the proposed algorithm and its actual application. Then, we set an experimental training and test environment.

1) Algorithm Evaluation: Precision and recall are two performance measurement indicators selected for algorithm evaluation experiments:

$$Percision = \frac{tp}{tp + fp}$$
$$Recall = \frac{tp}{tp + fn} \tag{12}$$

where tp represents the number of abandoned objects belonging to one class assigned to the same one by the algorithm, fp represents the number of abandoned objects classified as a wrong class, and fn represents the number of abandoned objects of one class not appearing in the correct class in the algorithm's output.

To evaluate the performance of the algorithm on all classes, we adopt the mAP indicator as the measurement when the threshold of score $= 0.5$. Benefiting from the GPU, it takes around 40 hours to train our deep learning model with 500 epochs and 300 epochs on two datasets, respectively.

2) Actual Application Evaluation: In actual application evaluation, we adopt the detection rate, detection cost time and deduplication rate to evaluate the running performance of our vehicular system. We select city roads as experimental roads. The speed of our vehicular system is about 20 km/h. The weather and illumination are normal. The detection rate indicator is the detection accuracy on city roads. The detection cost time is directly received from our vehicular system. The deduplication rate is given by

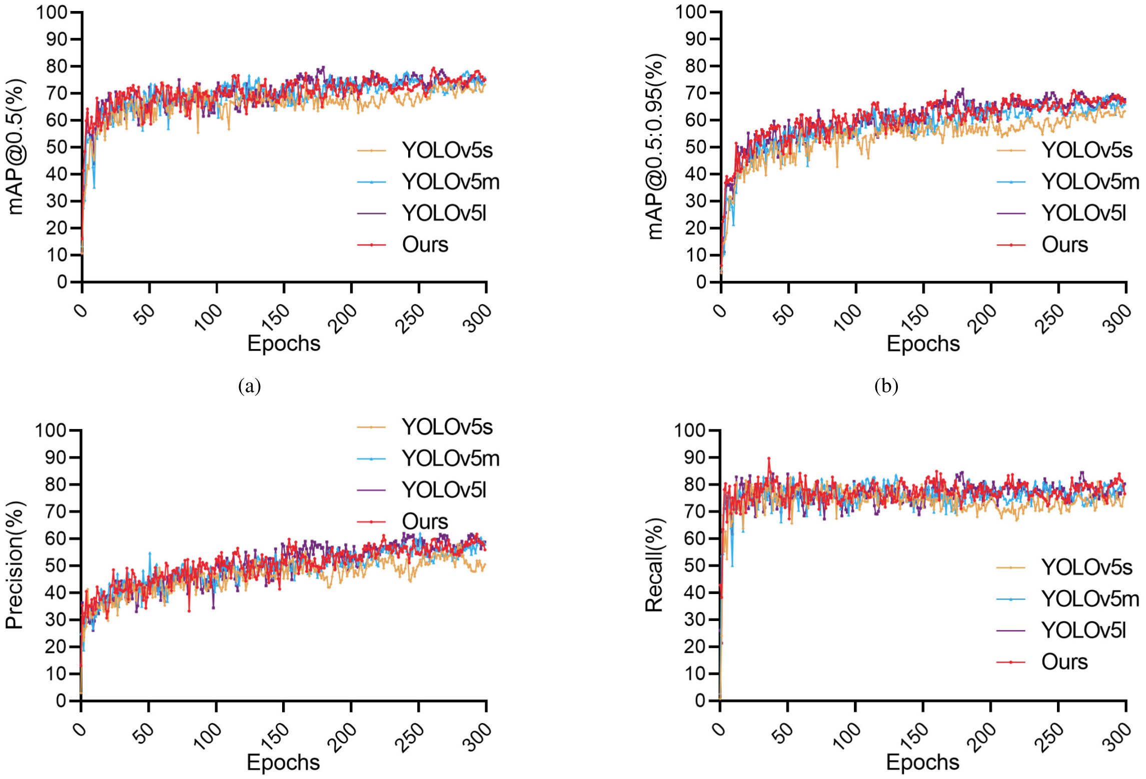
TABLE I
Details of Vehicular Devices

Device type	Description
Vehicle camera	HIKVISION iDS-TCC225
Edge AI computer	ADVANTECH MIC-7700
GPS device	RTK and GNSS receiver
LTE router	4G-LTE

$$\rho = \frac{S_{pre} - S_{after}}{S_{pre}} * 100\% \tag{13}$$

where S_{pre} and S_{after} represent the number of detected abandoned objects, respectively.

In addition, all detection images will be shown on an AutoNavi electronic map online according to location information from our vehicular system. The vehicular system works on the edge AI device, called ADVANTECH MIC-7700, including an intel i7-6700TE CPU@2.4GHz, 16 GB RAM and a single NVIDA Geforce RTX 2080TI GPU. Alibaba and AutoNavi cloud servers provide the electronic map online for location display. Our vehicular devices setting is shown in TABLE I and Fig.5.



(a)

(b)

(c)

(d)

Fig. 7. Comparison of training detailed curves for four methods on the AIcomp dataset: (a) YOLOv5s, (b) YOLOv5m, (c) YOLOv5l, and (d) Ours.

TABLE II

COMPARISON WITH FOUR METHODS FOR TWO DATASETS

Methods	**TACOsub dataset**	**AIcomp dataset**
	mAP(%)	mAP(%)
YOLOv5s [48]	99.55	51.49
YOLOv5m [48]	99.56	48.2
YOLOv5l [48]	99.57	51.87
Ours	99.57	53.11

C. Evaluation on Two Datasets

In this section, we compare our method with other state-of-the-art methods on our established two datasets in terms of mAP, precision and recall indicator. Moreover, we visualize training detail parameters and prediction results for extensive comparisons.

We first evaluate our method and the other YOLOv5 serial models on our established two datasets. TABLE II lists the results of mAP on two datasets. As shown in TABLE II, our method achieves 99.57%, and 53.11% in terms of mAP on two datasets, respectively, which performs significantly better than the other YOLOv5 serial models. Especially, the mAP of our method is 53.11%, whereas the other YOLOv5 serial models only have mAP of 51.49%, 48.2%, and 51.87%. The above results perform considerably better than the other YOLOv5 serial models.

Fig.6 and Fig.7 show the comparison of training detailed curves for four methods on the TACOsub dataset and AIcomp dataset, respectively. As shown in Fig. 6, our model achieves higher mAP, Precision, and Recall scores compared to other YOLOv5 models on the TACOsub dataset. However, our model utilizes a deeper and wider CNN network, leading to a slower convergence rate than the other YOLOv5 models. Fig. 7 shows the performance of the four models on the AIcomp dataset. Similar to Fig. 6, we can observe that the overall scores are slightly better than the other YOLOv5 models. However, our model still has a slower convergence rate compared to the other YOLOv5 models, due to the same reason as on the TACOsub dataset.

To evaluate the sophisticated classification ability of our method, we further compare the performance of abandoned object detection by our method with those of the other YOLOv5 serial methods on our established two datasets. TABLE III shows the classification detection amount of true positives and false positives on the TACOsub dataset. Moreover, precision and recall rates are exposed for the same classification results on the TACOsub dataset. Note the classification detection ratio of our method is overall more balanced than the other three methods on the TACOsub dataset. Similarly, we further verify the performance of our method on the AIcomp dataset in TABLE IV, which means the mAP of our method performs better than the other three methods in TABLE II. Moreover, Fig.8 shows visual comparisons of TABLE III and IV.

Fig.9 and Fig.10 show visual comparisons of labeled images with predicted images for our method on the TACOsub dataset and AIcomp dataset, respectively. From the visual comparison

TABLE III
CLASSIFICATION RESULTS OF FOUR METHODS ON THE TESTING DATASET OF THE TACOSUB DATASET

Methods	Class	TP	TP+FN	TP+FP	Recall (%)	Precision (%)	mAP(%)
YOLOv5s [48]	Clear plastic bottle	16	37	17	43.24	94.12	66.05
	Drink can	11	20	17	55	64.71	
	Plastic film	5	25	8	20	62.5	
	Unlabeled litter	15	55	35	27.27	42.86	
YOLOv5m [48]	Clear plastic bottle	16	37	22	43.24	72.73	58.18
	Drink can	11	20	12	55	91.67	
	Plastic film	5	25	14	20	35.71	
	Unlabeled litter	15	55	46	27.27	32.61	
YOLOv5l [48]	Clear plastic bottle	16	37	18	43.24	88.89	71.16
	Drink can	11	20	13	55	84.62	
	Plastic film	5	25	9	20	55.56	
	Unlabeled litter	15	55	27	27.27	55.56	
Ours	Clear plastic bottle	16	37	19	43.24	84.21	61.83
	Drink can	11	20	14	55	78.57	
	Plastic film	5	25	12	20	41.67	
	Unlabeled litter	15	55	35	27.27	42.86	

TABLE IV
CLASSIFICATION RESULTS OF FOUR METHODS ON THE TESTING DATASET OF THE AICOMP DATASET

Methods	Class	TP	TP+FN	TP+FP	Recall (%)	Precision (%)	mAP(%)
YOLOv5s [48]	Bottle	12	19	20	63.16	60	62.32
	Cloth	15	20	18	75	83.33	
	Kitchen Waste	16	20	25	80	64	
	Metal	11	20	18	55	61.11	
	Paper	17	20	24	85	70.83	
	Plastic	9	20	26	45	34.62	
YOLOv5m [48]	Bottle	12	19	22	63.16	54.55	58.67
	Cloth	15	20	23	75	65.22	
	Kitchen Waste	16	20	23	80	69.57	
	Metal	11	20	23	55	47.83	
	Paper	17	20	23	85	73.91	
	Plastic	9	20	22	45	40.91	
YOLOv5l [48]	Bottle	12	18	20	66.67	60	66.16
	Cloth	15	25	18	60	83.33	
	Kitchen Waste	16	32	25	50	64	
	Metal	17	23	18	73.91	94.44	
	Paper	9	25	24	36	37.5	
	Plastic	15	25	26	60	57.69	
Ours	Bottle	12	19	19	63.16	63.16	57.15
	Cloth	15	20	20	75	75	
	Kitchen Waste	16	20	27	80	59.26	
	Metal	11	20	24	55	45.83	
	Paper	17	20	23	85	73.91	
	Plastic	9	20	35	45	25.71	

results, the performance of our method on the TACOsub dataset is better than the AIcomp dataset. This result implies that the classification ability on the AIcomp dataset is more difficult than the classification TACOsub dataset. It is because the objects of the AIcomp dataset are more sophisticated than those of the TACOsub dataset.

TABLE V
TEST RESULTS OF OUR METHOD AND OTHER STATE-OF-THE-ART METHODS ON THE AIcomp DATASET

Methods	Class	AP(%)	Recall(%)	Precision (%)	mAP (%)
YOLOv3 [33]	Bottle	61.12	68.42	61.9	63.72
	Cloth	85.28	66.67	85.71	
	Kitchen Waste	85.79	80	96	
	Metal	40.32	34.78	44.44	
	Paper	69.53	66.67	71.43	
	Plastic	40.29	62.5	39.47	
YOLOv4 [32]	Bottle	37.44	31.58	60	56.32
	Cloth	77.78	77.78	87.5	
	Kitchen Waste	84.97	83.33	96.15	
	Metal	27.77	13.04	15.79	
	Paper	90.92	86.67	81.25	
	Plastic	19.05	0	0	
SSD [34]	Bottle	53.43	42.11	47.06	65.73
	Cloth	94.44	66.67	100	
	Kitchen Waste	90.04	63.33	86.36	
	Metal	42.16	21.74	38.46	
	Paper	67.87	53.33	88.89	
	Plastic	46.42	29.17	53.85	
CenterNet [49]	Bottle	19.4	0	0	25.29
	Cloth	13.17	0	0	
	Kitchen Waste	16.96	0	0	
	Metal	20.55	4.35	100	
	Paper	70.04	6.67	100	
	Plastic	11.63	0	0	
YOLOv7 [50]	Bottle	60.7	58.32	54.28	57.62
	Cloth	70.7			
	Kitchen Waste	63			
	Metal	52			
	Paper	79.9			
	Plastic	19			
YOLOv5s [48]	Bottle	-	77.02	38.2	51.49
	Cloth				
	Kitchen Waste				
	Metal				
	Paper				
	Plastic				
YOLOv5m [48]	Bottle	-	79.46	34.17	48.2
	Cloth				
	Kitchen Waste				
	Metal				
	Paper				
	Plastic				
YOLOv5l [48]	Bottle	-	81.13	31.64	51.87
	Cloth				
	Kitchen Waste				
	Metal				
	Paper				
	Plastic				
Ours	Bottle	-	78.34	29.21	53.11
	Cloth				
	Kitchen Waste				
	Metal				
	Paper				
	Plastic				

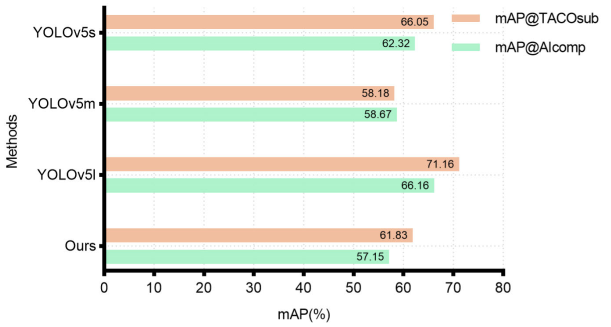
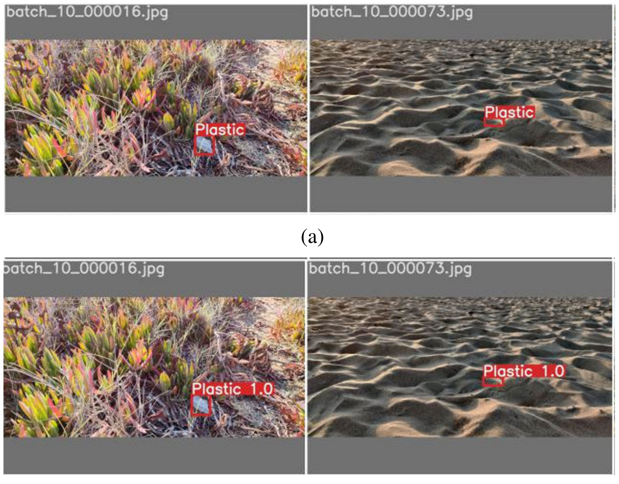


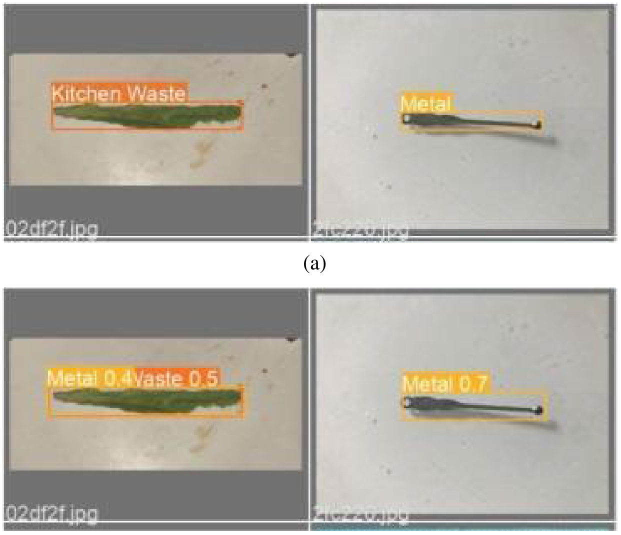
Fig. 8. Classification results in comparison of TABLE III and IV.



(a)

(b)

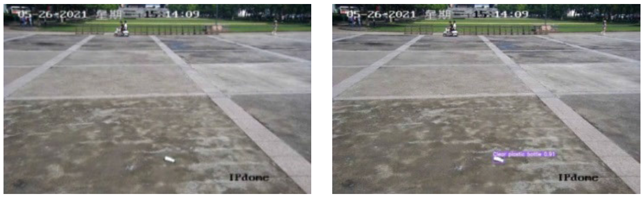
Fig. 9. Visual comparison of our method on the TACOsub dataset: (a) labeled images and (b) predicted images.



(a)

(b)

Fig. 10. Visual comparison of our method on the AIcomp dataset: (a) labeled images and (b) predicted images.



(a) (b)

Fig. 11. Detection performance of our vehicular system in real roads: (a) input video frame, (b) detection results.

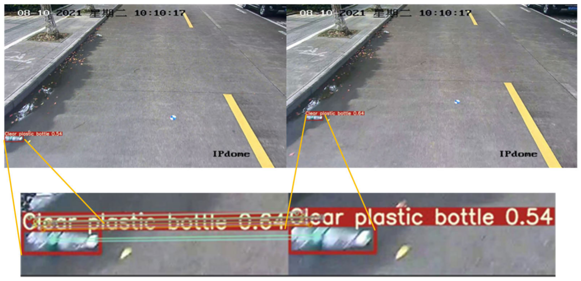


Fig. 12. Similarity matching procedure of the same detected abandoned object from consecutive frames in our deduplication module.

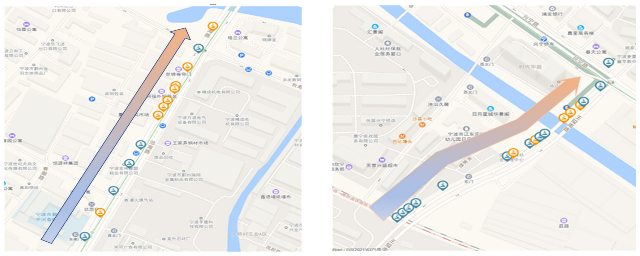
To further verify the detection ability of our method on the AIcomp dataset, we compare our method with other state-of-the-art methods which are recent principal detection models. TABLE V shows the test results of our method and other state-of-the-art methods on the AIcomp dataset. As shown in TABLE V, the best result is that the SSD model only achieves 65.73% at mAP. Although the SSD model is a bit better than the other YOLOv5 serious models, it still has a big gap between the YOLOv5 serious models and our method on the TACOsub dataset.

D. Performance Evaluation in Real Roads

According to the aforementioned results of the algorithm evaluation, we select the deep learning model based on our method from TACO dataset as our actual detection model. In this section, we evaluate the detection ability of our method under real road conditions. Besides, we also test the deduplication and localization performance.

In our road detection experiments, we integrate the proposed deep learning-based algorithm into our vehicular system. Then we run our vehicular system to detect abandoned objects on roads. The accuracy of our detection system on real roads is around 82%. Fig.11 shows the detection performance of our vehicular system on two real roads. As shown in Fig.10, our vehicular system can detect and classify correctly abandoned objects on real roads under daylight and normal weather. Moreover, the cost time of detection takes around 35-41ms per frame. It is sufficient for working on city roads.

The deduplication rate can achieve 98.2% under the static condition, whereas it can achieve 87.6% under the moving condition. Fig.11 shows the similarity matching procedure of the same detected abandoned object from consecutive frames in our deduplication module. As shown in Fig.12, we select a detection region in bounding boxes from consecutive frames to verify the matching performance. Our deduplication module can match directly detected objects. To obtain better matching performance, our deduplication module designs hybrid features to match the detection region in bounding boxes. Moreover, detection frequency is extremely high in mobile circumstances. Therefore, we only search co-located bounding



(a) (b)

Fig. 13. Localization accuracy performance of abandoned objects on the AutoNavi map: (a) direct simple road and (b) complex road.

boxes to match the similarity of repeated objects, which can avoid increasing too much extra computational time.

To verify the localization performance of our vehicular system, we select two city roads as the experimental roads. One road is a direct simple road which is a little easier for localization. The other road is a complex road with a bridge and crossroads which is more difficult for localization. Fig.13 shows the localization accuracy performance of abandoned objects on the AutoNavi map. As shown in Fig.13, our location estimation approach can provide better localization effect for abandoned objects on both of the two roads, which means that management departments are straightforward for finding and clearing abandoned objects. Therefore, this can decrease traffic accidents and keep them tidy.

VII. Conclusion

In this paper, we presented a vehicular detection approach for abandoned objects based on VANET and edge AI for a new application scenario on roads. Firstly, a vehicular detection architecture for abandoned objects was proposed to achieve a task-based AI technology for large-scale road maintenance in mobile computing circumstances. In our proposed architecture, all detection computations can be completed in a vehicular edge AI device.Then, we proposed a detection algorithm combining a deep learning network and a deduplication module for high-frequency detection computation in mobile computing circumstances. We also established two abandoned object datasets by collecting extensive images from the Internet. Finally, a location estimation approach for abandoned objects was proposed to achieve accurate calculation of their positions. Experimental results showed that the proposed algorithm and our whole approach achieved better performance.

For future work, we plan to explore an enhanced algorithm to process foggy and nighttime scenes under different light conditions. In addition, we aim to collect more real-world images from our system for self-training, with the goal of continually optimizing our deep learning model.

References

[1] M. Quintana, J. Torres, and J. M. Menéndez, “A simplified computer vision system for road surface inspection and maintenance,” *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 3, pp. 608–619, Mar. 2016.

[2] R. Fan, U. Ozgunalp, B. Hosking, M. Liu, and I. Pitas, “Pothole detection based on disparity transformation and road surface modeling,” *IEEE Trans. Image Process.*, vol. 29, pp. 897–908, 2020.

[3] Y.-A. Daraghmi, T.-H. Wu, and T.-U. Ik, “Crowdsourcing-based road surface evaluation and indexing,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 5, pp. 4164–4175, May 2022.

[4] J. Lan, D. Yu, B. Ran, Y. Jiang, and Z. Zhang, “A new threat degree analysis method of abandoned objects based on dynamic multifeature fusion in urban traffic,” *J. Intell. Transp. Syst.*, vol. 20, no. 4, pp. 401–412, Jul. 2016.

[5] Z. Wu, C. Liu, J. Wen, Y. Xu, J. Yang, and X. Li, “Selecting high-quality proposals for weakly supervised object detection with bottom-up aggregated attention and phase-aware loss,” *IEEE Trans. Image Process.*, vol. 32, pp. 682–693, 2023.

[6] D.-P. Fan, G.-P. Ji, M.-M. Cheng, and L. Shao, “Concealed object detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 6024–6042, Oct. 2022.

[7] W. J. Kim, S. Hwang, J. Lee, S. Woo, and S. Lee, “AIBM: Accurate and instant background modeling for moving object detection,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 9021–9036, Jul. 2022.

[8] Y. Yang, J. Ruan, Y. Zhang, X. Cheng, Z. Zhang, and G. Xie, “STPNet: A spatial–temporal propagation network for background subtraction,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 2145–2157, Apr. 2022.

[9] C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern*, Jun. 1999, pp. 246–252.

[10] O. Barnich and M. Van Droogenbroeck, “ViBe: A universal background subtraction algorithm for video sequences,” *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1709–1724, Jun. 2011.

[11] A. Begel, Y. P. Khoo, and T. Zimmermann, “Codebook: Discovering and exploiting relationships in software repositories,” in *Proc. ACM/IEEE 32nd Int. Conf. Softw. Eng.*, vol. 1, May 2010, pp. 125–134.

[12] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, *arXiv:1409.1556*.

[13] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[14] W. Zaremba, I. Sutskever, and O. Vinyals, “Recurrent neural network regularization,” 2014, *arXiv:1409.2329*.

[15] Q. Guo and M. Zhou, “Progressive domain translation defogging network for real-world fog images,” *IEEE Trans. Broadcast.*, vol. 68, no. 4, pp. 876–885, Dec. 2022.

[16] J. Liao et al., “Road garbage segmentation and cleanliness assessment based on semantic segmentation network for cleaning vehicles,” *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 8578–8589, Sep. 2021.

[17] M. C. S. Santana, L. A. Passos, T. P. Moreira, D. Colombo, V. H. C. de Albuquerque, and J. P. Papa, “A novel siamese-based approach for scene change detection with applications to obstructed routes in hazardous environments,” *IEEE Intell. Syst.*, vol. 35, no. 1, pp. 44–53, Jan. 2020.

[18] Z. Chunxiang, Q. Jiacheng, and B. Wang, “YOLOX on embedded device with CCTV & TensorRT for intelligent multicategories garbage identification and classification,” *IEEE Sensors J.*, vol. 22, no. 16, pp. 16522–16532, Aug. 2022.

[19] H. Park, S. Park, and Y. Joo, “Detection of abandoned and stolen objects based on dual background model and mask R-CNN,” *IEEE Access*, vol. 8, pp. 80010–80019, 2020.

[20] Y. Dong, L. Song, R. Xie, and W. Zhang, “An elastic system architecture for edge based low latency interactive video applications,” *IEEE Trans. Broadcast.*, vol. 67, no. 4, pp. 824–836, Dec. 2021.

[21] H. Cao, S. Garg, G. Kaddoum, M. M. Hassan, and S. A. AlQahtani, “Intelligent virtual resource allocation of QoS-guaranteed slices in B5G-enabled VANETs for intelligent transportation systems,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 19704–19713, Oct. 2022.

[22] Y. He, L. Nie, T. Guo, K. Kaur, M. M. Hassan, and K. Yu, “A NOMA-enabled framework for relay deployment and network optimization in double-layer airborne access VANETs,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 22452–22466, Nov. 2022.

[23] H. Cheng, M. Shojafar, M. Alazab, R. Tafazolli, and Y. Liu, “PPVF: Privacy-preserving protocol for vehicle feedback in cloud-assisted VANET,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 9391–9403, Jul. 2022.

[24] G. Luo et al., “Software-defined cooperative data sharing in edge computing assisted 5G-VANET,” *IEEE Trans. Mobile Comput.*, vol. 20, no. 3, pp. 1212–1229, Mar. 2021.

[25] S. P. Xu, K. Wang, M. R. Hassan, M. M. Hassan, and C.-M. Chen, "An interpretive perspective: Adversarial trojaning attack on neural-architecture-search enabled edge AI systems," *IEEE Trans. Ind. Informat.*, vol. 19, no. 1, pp. 503–510, Jan. 2023.

[26] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[27] J. Ning, J. Yang, S. Jiang, L. Zhang, and M.-H. Yang, "Object tracking via dual linear structured SVM and explicit feature map," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4266–4274.

[28] M.-T. Pham, Y. Gao, V.-D. D. Hoang, and T.-J. Cham, "Fast polygonal integration and its application in extending Haar-like features to improve object detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 942–949.

[29] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

[30] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[31] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525.

[32] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[33] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[34] W. Liu et al., "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 21–37.

[35] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[36] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[37] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 91–99.

[38] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[39] M. Simony, S. Milzy, K. Amendey, and H.-M. Gross, "Complex-YOLO: An euler-region-proposal for real-time 3D object detection on point clouds," in *Proc. Eur. Conf. Comput. Vis. (ECCV) Workshops*, Sep. 2018, pp. 1–14.

[40] J. Choi, D. Chun, H. Kim, and H.-J. Lee, "Gaussian YOLOv3: An accurate and fast object detector using localization uncertainty for autonomous driving," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 502–511.

[41] Y. Cai et al., "YOLOv4–5D: An effective and efficient object detector for autonomous driving," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–13, 2021.

[42] A. Aboah, M. Shoman, V. Mandal, S. Davami, Y. Adu-Gyamfi, and A. Sharma, "A vision-based system for traffic anomaly detection using deep learning and decision trees," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 4202–4207.

[43] J.-A. Sarmiento, "Pavement distress detection and segmentation using YOLOv4 and DeepLabv3 on pavements in the Philippines," 2021, *arXiv:2103.06467*.

[44] A. Di Carlofelice, I. Lucresi, and P. Tognolatti, "A geostationary satellite time and frequency dissemination system: A preliminary experiment," *IEEE Trans. Broadcast.*, vol. 68, no. 1, pp. 215–222, Mar. 2022.

[45] J. Ribadeneira-Ramírez, G. Martínez, D. Gómez-Barquero, and N. Cardona, "Interference analysis between digital terrestrial television (DTT) and 4G LTE mobile networks in the digital dividend bands," *IEEE Trans. Broadcast.*, vol. 62, no. 1, pp. 24–34, Mar. 2016.

[46] *TACO: Trash Annotations in Context for Litter Detection*. Accessed: Mar. 17, 2020. [Online]. Available: https://github.com/pedropro/TACO

[47] *Xi'an 2021 High Skilled Talent Skills Competition*. Accessed: Jul. 15, 2021. [Online]. Available: https://biendata.xyz/competition/ai_future_xian

[48] G. Jocher. (2020). *Ultralytics YOLOv5*. [Online]. Available: https://github.com/ultralytics/yolov5

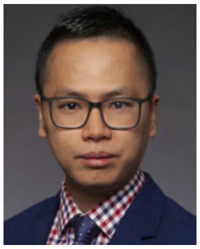
[49] X. Zhou, J. Zhuo, and P. Krähenbühl, "Bottom-up object detection by grouping extreme and center points," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 850–859.

[50] K.-Y. Wong. (Sep. 2022). *Official YOLOv7*. GitHub. [Online]. Available: https://github.com/WongKinYiu/yolov7

Gang Wang received the Ph.D. degree in computer science from Beihang University, Beijing, China, in 2018. He was a Visiting Scholar with the Imperial College London, London, U.K. He is currently a Lecturer with the School of Computing and Data Engineering, NingboTech University, Ningbo, China. He is also a Post-Doctoral Fellow with the Chongqing Key Laboratory of Image Cognition, School of Computer Science and Technology, Chongqing University of Posts and Telecommunications. His current research interests include multimedia compression, image/video processing, multimedia communication, machine learning, and hardware optimization.

Mingliang Zhou received the Ph.D. degree in computer science from Beihang University, Beijing, China, in 2017. He was a Post-Doctoral Researcher with the Department of Computer Science, City University of Hong Kong, Hong Kong, China, from September 2017 to September 2019. He was a Post-Doctoral Fellow with the State Key Laboratory of Internet of Things for Smart City, University of Macau, Macau, China, from October 2019 to October 2021. He is currently an Associate Professor with the School of Computer Science, Chongqing University, Chongqing, China. His research interests include image and video coding, perceptual image processing, multimedia signal processing, rate control, multimedia communication, machine learning, and optimization.

Xuekai Wei received the bachelor's degree in electronic information science and technology and the master's degree in communication and information systems from Shandong University in 2014 and 2017, respectively, and the Ph.D. degree in computer science from the City University of Hong Kong in 2021. His current research interests include video coding, video transmission, and machine learning.



Guang Yang (Senior Member, IEEE) received the M.Sc. degree in vision imaging and virtual environments from the Department of Computer Science, University College London, in 2006, and the Ph.D. degree in medical image analysis jointly from CMIC, Department of Computer Science and Medical Physics, University College London, in 2012. He is currently a tenured Senior Research Fellow with NHLI, Imperial College London, and a Honorary Senior Lecturer with the School of Biomedical Engineering and Imaging Sciences, King's College London. He is also the Head of the Smart Imaging Laboratory funded by UKRI, BHF, and ERC. His research interests include pattern recognition, machine learning, and medical image processing and analysis.