Portable IoT device for tire text code identification via integrated computer vision system

Received: 7 November 2024 | Accepted: 31 January 2025

DOI: 10.1111/mice.13438

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

RESEARCH ARTICLE

Portable IoT device for tire text code identification via integrated computer vision system

Haowei Zhang[1,2] | **Kang Gao**[1,2] | **Yue Hou**[3] | **Marco Domaneschi**[4] | **Mohammad Noori**[5]

[1]Key Laboratory of Concrete and Prestressed Concrete Structures of the Ministry of Education, Southeast University, Nanjing, China

[2]National and Local Joint Engineering Research Center for Intelligent Construction and Maintenance, Southeast University, Nanjing, China

[3]Department of Civil Engineering, Faculty of Science and Engineering, Swansea University, Wales, UK

[4]Department of Structural, Geotechnical and Building Engineering, Politecnico di Torino, Turin, Italy

[5]College of Engineering, California Polytechnic State University, San Luis Obispo, California, USA

Correspondence
Kang Gao, Key Laboratory of Concrete and Prestressed Concrete Structures of the Ministry of Education, Southeast University, Nanjing 211189, China.
Email: gaokang@seu.edu.cn

Yue Hou, Department of Civil Engineering, Faculty of Science and Engineering, Swansea University, Wales SA2 8PP, UK.
Email: yue.hou@swansea.ac.uk

Funding information
Fundamental Research Funds for the Central Universities, Grant/Award Number: 2242023K5006; National Natural Science Foundation of China, Grant/Award Numbers: 52208151, 52127813

Abstract

The identification of tire text codes (TTC) during the production and operational phases of tires can significantly improve safety and maintenance practices. Current methods for TTC identification face challenges related to stability, computational efficiency, and outdoor applicability. This paper introduces an automated TTC identification system founded on a robust framework that is both user-friendly and easy to implement, thereby enhancing the practical use and industrial applicability of TTC identification technologies. Initially, instance segmentation is creatively utilized for detecting TTC regions on the tire sidewall through You Only Look Once (YOLO)-v8-based models, which are trained on a dataset comprising 430 real-world tire images. Subsequently, a computationally efficient rotation algorithm, along with specific image pre-processing techniques, is developed to tackle common issues associated with centripetal rotation in the TTC region and to improve the accuracy of TTC region detection. Furthermore, a series of YOLO-v8 object detection models were assessed using an independently collected dataset of 1127 images to optimize the recognition of TTC characters. Ultimately, a portable Internet of Things (IoT) vision device is created, featuring a comprehensive workflow to support the proposed TTC identification framework. The TTC region detection model achieves a segmentation precision of 0.8812, while the TTC recognition model reaches a precision of 0.9710, based on the datasets presented in this paper. Field tests demonstrate the system's advancements, reliability, and potential industrial significance for practical applications. The IoT device is shown to be portable, cost-effective, and capable of processing each tire in 200 ms.

1 | INTRODUCTION

Vehicle tire information provides vital information for transportation and infrastructure maintenance (Sitton et al., 2024; Y. Yang et al., 2024). Among all, tire text codes (TTC) identification offers the most intuitive access to vehicle information (Kazmi et al., 2019), such as "imprinting quality of TTC" for online tire production quality inspection (Cheng et al., 2024), which is primarily used in indoor application, and "manufacturing date" for evaluating the remaining service life of on-road vehicle (Ju et al., 2024; Kazmi et al., 2020) and "size information" for estimating the gross weight of service truck (M. Q. Feng et al., 2020; Kong et al., 2022; D. Li et al., 2024), applicable in outdoor scenarios. TTC also contains speed and load index, which can be used for traffic information management (Kamjoo et al., 2024; Yeum et al., 2016). However, to efficiently and accurately identify TTC is one of the most challenging optical character recognition (OCR) tasks due to inherent properties like small size, abrasion, low contrast, irregular geometric arrangement, varied shapes and sizes, and external factors like vehicle speed, illumination, detection distance. Therefore, the traditional tire OCR methods, such as template matching (Scholz & Koehler, 2012), laser scanning (Sukprasertchai & Suesut, 2016), digital image processing (C. Liu et al., 2022), and edge operator detection (Ham et al., 1995) are transforming into intelligent character recognition (ICR) utilizing deep learning-based image analysis for enhanced robustness and precision. Yet, current outdoor tire ICR tasks face limitations, including complex multi-stage processing, high computing resource consumption, and inefficient device workflows. Therefore, it is essential to develop a fully upgraded, economical Internet of Things (IoT; Garrido-Hidalgo et al., 2023) TTC identification system (D. Liu et al., 2024) that is easily intelligible, deployable, and repeatable, demonstrating strong industry potential.

TTC region detection is the crucial initial step that determines the reliability and stability of the TTC identification task. Kazmi et al. (2019, 2020) developed a region of interest (RoI; Girshick, 2015) based convolutional neural network (CNN) to detect the "Department of Transportation" (DOT) TTC region on the tire images that have been straightened using histogram of oriented gradients algorithm so as to detect tire circularity. However, this method involves multiple image processing steps that require significant computing resources, particularly due to its reliance on a binocular vision system. This dependency limits the size and speed of the vehicles that can be measured and introduces instability in performance. F. Gao et al. (2021) utilized YOLO-v4 object detection means (Pan et al., 2023) for initial TTC region detection and subsequently trained a CRAFT network for refined TTC region detection. However, due to the curved and multi-directional nature of the TTC, the bounding box cannot accurately delineate the edges of the TTC region. This limitation often results in bounding boxes that encompass extraneous areas around the TTC region, which can significantly impact the accuracy of TTC recognition following the initial detection. Moreover, secondary TTC detection methods involve higher computing costs and reduced robustness. Segmentation-based TTC region detection, which achieves pixel-level accuracy, offers a solution (Z. He et al., 2024). Although many studies have demonstrated success with curved text detection in common scene applications (Zhuotao Tian et al., 2019; Xu et al., 2019), curved OCR detection methods, such as Progressive Scale Expansion Network (PSE-Net) (W. Wang et al., 2019), require complex dataset annotation process as evidenced by CTW1500 datasets. Hence, a pixel-level precision TTC region detection method that simplifies dataset creation is needed to improve TTC detection accuracy and reduce the need for complex image post-processing.

TTC regions are not only curved but can also be completely reversed and have irregular centripetal arrangement. Centripetal rotation of the TTC region presents another significant challenge following the initial detection. Current button-up approaches require computationally intensive post-processing to extract recognizable text. Yin et al. (2015) adopted clustering methods to generate readable text regions for slant fonts, involving three types of clustering, which may result in reduced robustness. Shi et al. (2017) used a segmentation approach to detect each word individually and then connected each target into words using a proposed directional adjacent link method, which needed huge computation resources. F. Gao et al. (2020) further applied similar techniques on metal OCR tasks, generating character bounding points and fitting connected curve to recertify rotated characters. F. Gao et al. (2021) then implanted this centripetal character curves fitting method on TTC identification. However, the proposed rotation strategy failed to align the TTC into an absolute upright position due to the misalignment of YOLO's bounding boxes, necessitating multiple post-processing steps. This resulted in costly text-by-text recertification. Another costly solution, Kazmi et al. (2019, 2020) employed the unwrapped-segment method by combining two cameras to work as a binocular vision system capturing tire image twice to separate tire rubber from the rim. However, this process took 500 ms per image and required complex multi-system linkage, and fixed size tires can only be detected at a fixed distance. Therefore, a computationally effective, stable, and general TTC rotation algorithm is essential to streamline the overall TTC

identification workflow and to facilitate the creation of a high-quality, readable TTC recognition dataset.

There is currently no end-to-end method for TTC recognition due to the limitations imposed by the scarcity and specificity of available database. Existing scene text detection and recognition methods are not well-suited for TTC identification (F. Gao et al., 2021). In the field of vision-based non-contact vehicle weight-in-motion, Feng et al. (M. Q. Feng & Leung, 2021) utilized open-sourced OCR Application Programming Interface (API) to recognize four types of TTC after background filtering. However, this approach demonstrated poor detection stability. Kazmi et al. (2019, 2020) developed a lightweight CNN network to categorize TTC characters into 39 classes, training it on synthetic images one word at a time. This recognition method focused solely on identifying "DOT," and word-by-word training approach hindered the model to understand the global context of TTC. Similarly, F. Gao et al. (2021) compared ASTER, MORANv2, and CRNN models to develop an effective TTC recognition system, expanding its application beyond "DOT" to include other types of TTC (Cheng et al., 2024). Nevertheless, "image-to-sequence" datasets used in the methods described above are difficult to produce and are not publicly available, which hindered the repeatability of TTC identification methods. In the authors' opinion, TTC is not merely scene optical character text; rather, each TTC character can be regarded as a distinct object category. Given the state-of-the-art performance of YOLO models (Dan et al., 2024; Jocher et al., 2023; Redmon & Farhadi, 2018), adapting TTC recognition to handwriting classification (Lecun et al., 1998) based on object detection tasks could yield significant results even with relatively small datasets. This approach would circumvent the need for extensive pre-processing techniques typically required for scene text recognition. Consequently, it would be valuable to establish an object detection-based benchmark TTC recognition database and to exploit an object detection model for TTC recognition tasks (Rafiei & Adeli, 2017), facilitating ease of replication and future upgrade.

In conclusion, outdoor TTC recognition is of important value for transportation maintenance. However, complex character shapes, multi-directional curvature, low contrast, and sensitivity to lighting and vehicle speed have limited the performance of OCR methods. Existing ICR approaches still face high computational costs, complex workflows, limited generalization, and difficulties in outdoor deployment. Therefore, this study proposes a simplified, standardized dataset production method for deep learning-based TTC identification framework, along with a specifically designed TTC angle correction algorithm, all integrated into IoT devices to address the current insufficient TTC research in outdoor application of estimating remaining service life and gross weight. Consequently, this paper aims to advance outdoor TTC identification technology to a more practical level of industrialization by developing a cost-effective and portable IoT device and establishing an efficient, easily deployable three-stage algorithm framework, which is beneficial for real-time transportation management.

1. A dataset of 430 high-quality-annotated tire sidewall images (TSIs) is collected, and instance segmentation is employed for pixel-level detection of the TTC region in the first instance. An optimal YOLO-seg model is trained.
2. A novel and efficient TTC rotation algorithm is proposed to work seamlessly with segmentation-based TTC detection, eliminating the need for complex and unnecessary character-by-character recertification.
3. Building on the concept of handwriting OCR, object detection is introduced to enhance TTC recognition, providing greater robustness and facilitating easier data production. A dataset of 1127 high-quality-annotated TTC recognition images was generated using the proposed framework to comprehensively train an optimal YOLO-obj model.
4. An automatic TTC identification system, which is compact, efficient, and user-friendly, has been developed by integrating the proposed three-stage algorithm framework into a portable IoT device. This device features complete workflow and can be easily deployed in various application scenarios. The reliability of the system has been evaluated through field tests.

This study is presented in the following order. Section 2 introduces the overview of the proposed methods, Section 3 details the proposed framework, Section 4 illustrates the experiments and validations, Section 5 describes the developed IoT device and system, and Section 6 gives the conclusion.

2 | METHODOLOGY OF PROPOSED AUTOMATIC TTC IDENTIFICATION SYSTEM

TTC identification is a complex task due to several challenges, including low contrast, small size, irregular morphology, and centripetal, as shown in Figure 1. To identify TTC, the TSI is needed as the basic element for extracting the essential tire sidewall character image (TSCI) data, which contain the required character data that needs to be recognized.

This study presents a straightforward and robust strategy for TTC identification, and the establishment process

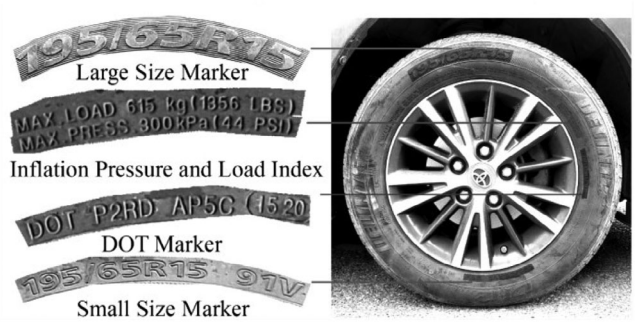

FIGURE 1 Pictures of tire sidewall character image (TSCI) and tire sidewall image (TSI). DOT, Department of Transportation.

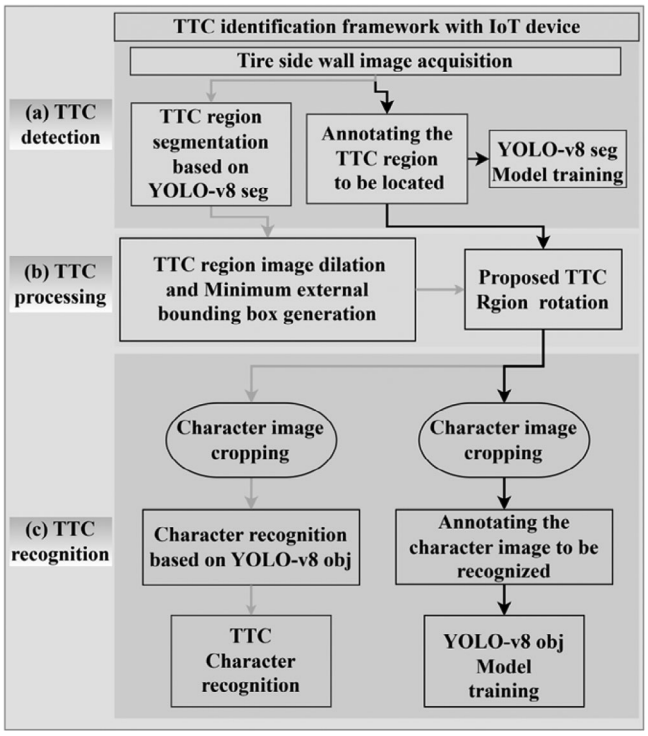

FIGURE 2 Establishment of the proposed tire text codes (TTC) identification framework: (a) TTC detection, (b) pre-processing of TTC image, and (c) TTC recognition. IoT, Internet of Things.

of the proposed TTC identification framework is shown in Figure 2.

Here, "TTC detection" refers to identifying the region of tire character region while "TTC recognition" involves interpreting these characters. Initially, TSI datasets, encompassing various tire types, were collected to train a YOLO-v8-based segment model for detecting four types of TTC. Utilizing image pre-processing techniques and the proposed TTC rotation algorithm, the TTC is oriented positively for accurate character recognition. Meanwhile, TSCI datasets were generated for training the TTC recognition model. A YOLO-v8-based object detection model was trained across the four types. In Figure 2, the black line flowchart depicts the establishment of the datasets and models, while the light line flowchart shows the complete inference stage for TTC identification. Finally, an innovative IoT roadside device was invented for automated TTC identification of passing vehicles.

In conclusion, an automatic TTC identification system was developed by integrating AI (Artificial Intelligence)-based framework into an IoT device for multi-transportation scene application as shown in Figure 3. This system features a comprehensive workflow and has been deployed in field tests to enhance current TTC identification solutions. The detail of the proposed framework is illustrated in Figure 4. The IoT device facilitates TTC identification by capturing tire images at the roadside. These images are processed by a development board using integrated deep learning models and algorithms, allowing for the recognition of four types of TTC.

3 | METHOD OF DETECTING, ROTATING, AND RECOGNIZING TTC

3.1 | Segmentation-based TTC detection method

This study aims to develop high-performance methods for TTC identification to extract tire sidewall character information, including tire/rim specifications, speed levels, standard pressure, and so forth.

Given that different types of TTC are randomly positioned on tire sidewalls, and varied in shapes and sizes, accurate TTC region detection is crucial. Unlike previous research that directly employed object detection networks for TTC region detection, this study introduces a novel strategy that combines image segmentation methods with computational algorithms to achieve pixel-level detection of the TTC region. This segmentation approach effectively avoids the detection of redundant negative samples and irrelevant regions (H. Li et al., 2024) and provides high-quality TSCI for TTC recognition. To ensure rapid and accurate TTC region detection, this study utilizes a YOLO-v8 seg-based network to develop a robust and efficient TTC detection model. YOLO-v8 represents an advanced integration of object detection and segmentation, marking a significant improvement within the YOLO series. The open-source available YOLO family has facilitated its widespread application in intelligence transportation research. Due to the significant advancement in the YOLO-v8 model, this study posits that (1) the robust feature extraction capabilities of the Cross Stage Partial Network (CSPNet) backbone (Bochkovskiy et al., 2020; C.-Y. Wang et al., 2020), enhanced by the introduction of C2f convolution model, can effectively facilitate TTC region detection;

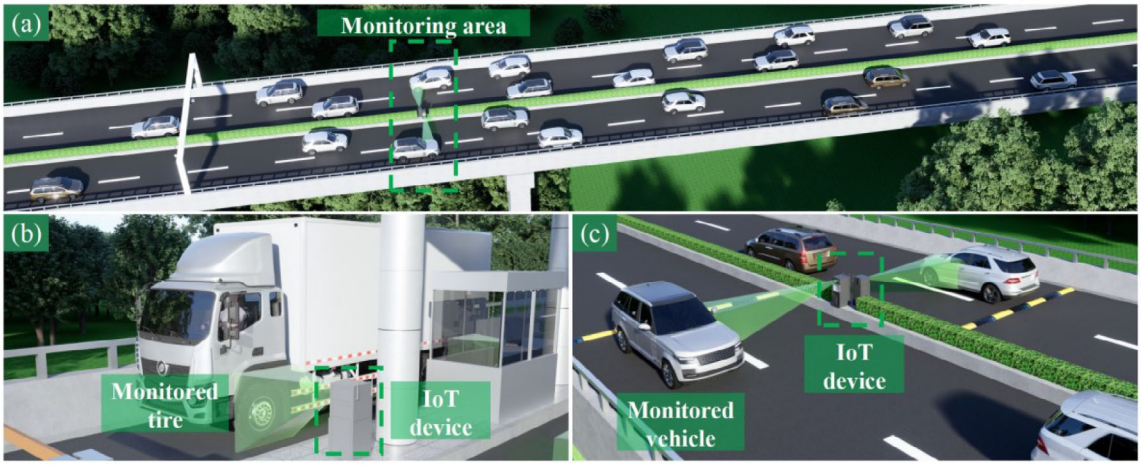

FIGURE 3 Application scenarios of the proposed TTC identification system: (a) bridge, (b) toll station, and (c) road. IoT, Internet of Things.

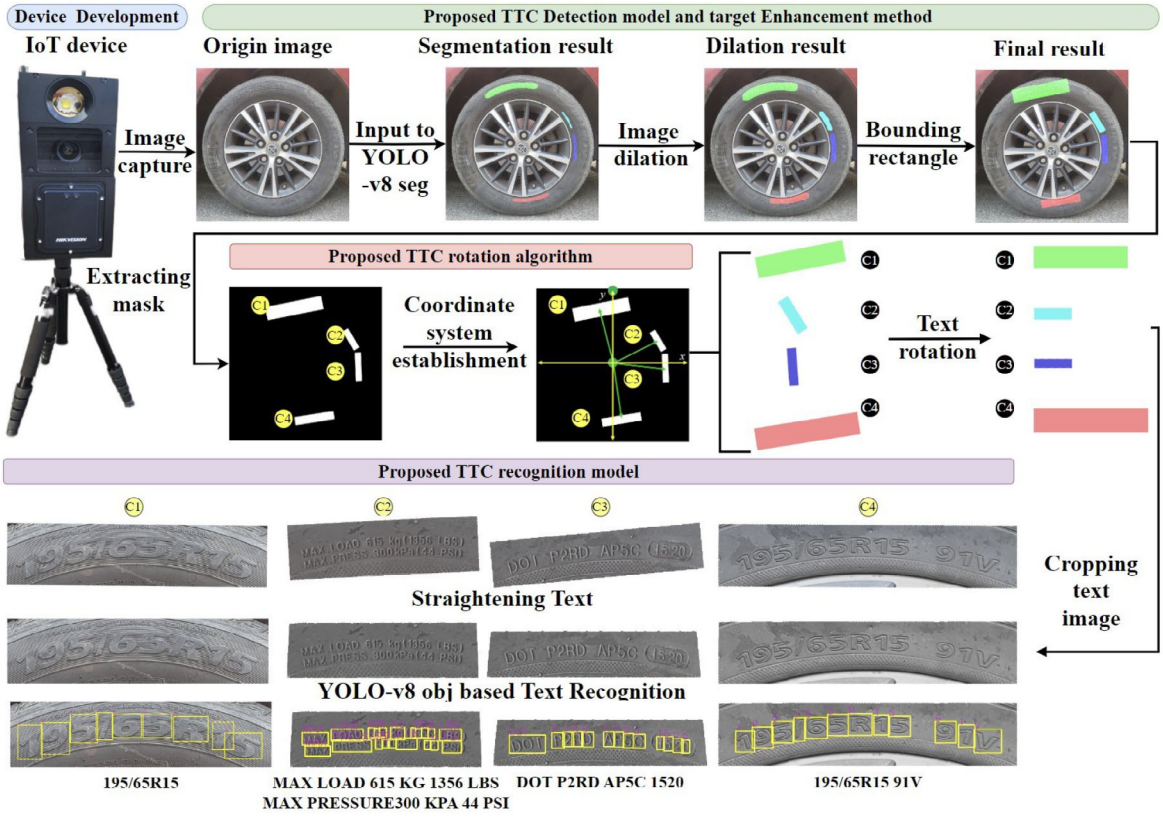

FIGURE 4 Overview of the proposed TTC identification system. IoT, Internet of Things.

(2) integration of the feature pyramid network (FPN; Lin et al., 2017) with path aggregation network (S. Liu et al., 2018) structure can balance global and local information, which is advantageous for detecting TTC regions at various scales; and (3) dynamic allocation of negative and positive samples allows the model to concentrate more on the interested TTC targets while avoiding additional inference costs for the proposed pixel-level task. However, selecting the optimal model still requires comparing YOLO-v8 seg-based models with different parameters, data augmentation techniques, and detecting heads to adapt to the proposed dataset.

As shown in Figure 5a, the YOLO-v8 architecture processes images through five convolutional layers (P1–P5), enhancing feature depth from 64 to 1024 while reducing resolution. P modules include Conv2d, BatchNorm2d,

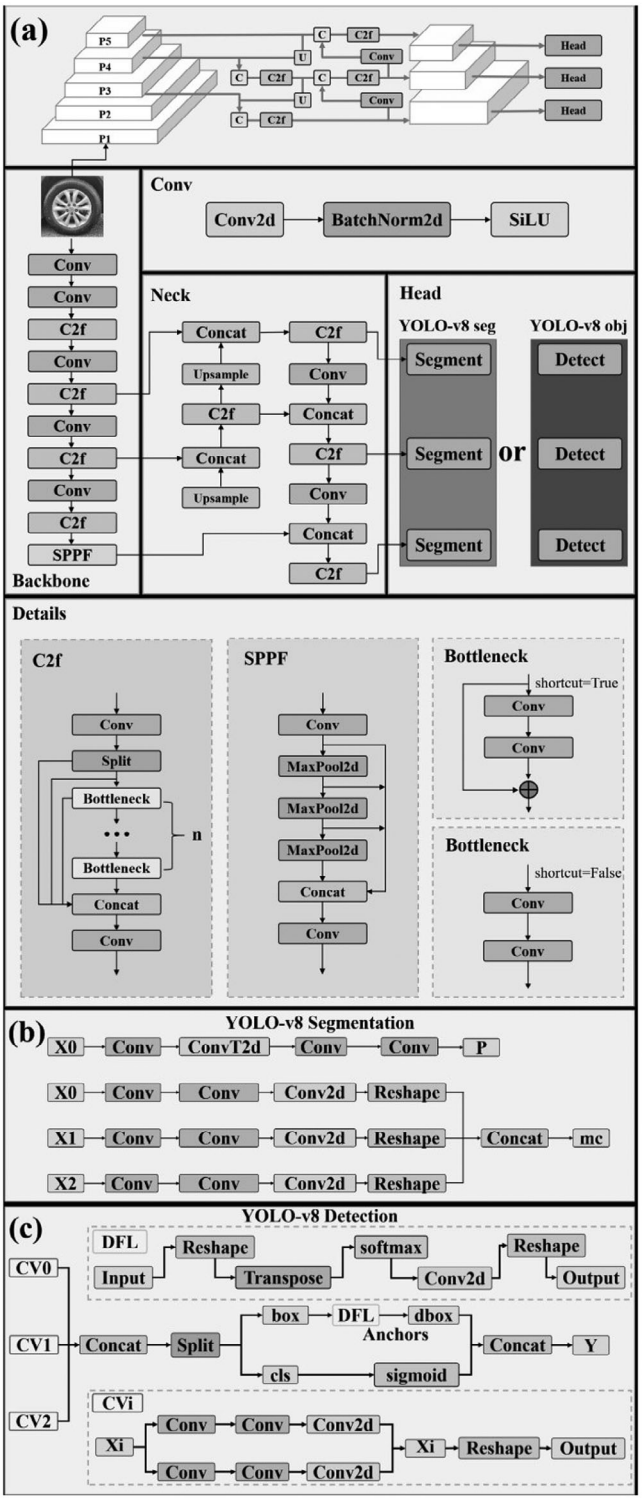

FIGURE 5 Architecture of YOLO-v8: (a) YOLO-v8 baseline, (b) YOLOv8 Segmentation head, and (c) YOLO-v8 Detection head.

and Sigmoid Linear Unit (SiLU) layers, transitioning from low-level details to high-level characteristics. C2f modules stack bottleneck layers for improved feature extraction. The FPN combines high- and low-level features for multi-scale outputs (X0, X1, X2), and the Spatial Pyramid Pooling - Fast (SPPF) module enriches contextual information. The segmentation head combines features from three different levels, assigning pixel-level labels to detected targets. Feature X0 is up-sampled to create a prototype mask, while X1, X2, and X3 are processed to predict anchor box confidence scores, classes, and bounding box coordinates, as shown in Figure 5b.

To enhance the segmentation performance of the model in the TTC region detection task, this paper employs an instance segmentation approach. Specifically, the four types of TTC targets, "Large Size Mark," "Inflation Pressure and Load Index," "DOT Marker," and "Small Size Marker" are categorized into distinct classes. This segmentation scheme incorporates a classification task to increase the model's accuracy. If all four types of TTC regions are regarded as the same category, the model might incorrectly recognize TTC regions not of interest, as the non-target regions could be visually like the targeted ones. The initial positioning of the TTC region is vital, and the bounding box generated by YOLO may not align with the centripetal direction of the TTC region. Consequently, the YOLO bounding box is frozen to save certain computing resources. Instead, a new bounding box re-establishment strategy, which is more advantageous for TTC recognition, will be introduced in the next section. Once the model roughly locates the TTC region, if the segmentation is insufficient to cover all TTC regions, image pre-processing techniques can be applied to enhance segmentation and provide high-quality TSCI. However, if positioning errors occur, they may preclude further processing steps. Additionally, TTC region detection results cannot be directly used for TTC recognition without orientation correction. Therefore, this study has specifically designed a simplified and intuitive general pre-processing algorithm to improve TTC detection results as opposed to text-by-text rectification.

3.2 | General pre-processing algorithm for TTC detection

The automatic recognition of tire characters poses significant challenges due to their tilted angles and irregularity. Moreover, the segmentation results from the TTC detection model might fail to completely detect the TTC region or cover the entire target area. To address these issues, a novel algorithm has been proposed to rotate the tire characters into a standard orientation. This is complemented by image noise filtering, dilation of the TTC target area, and bounding box generation to improve the quality of TCSI. These enhancements aim to optimize the previous methods and improve the accuracy of character recognition. The proposed rotation algorithm and three image

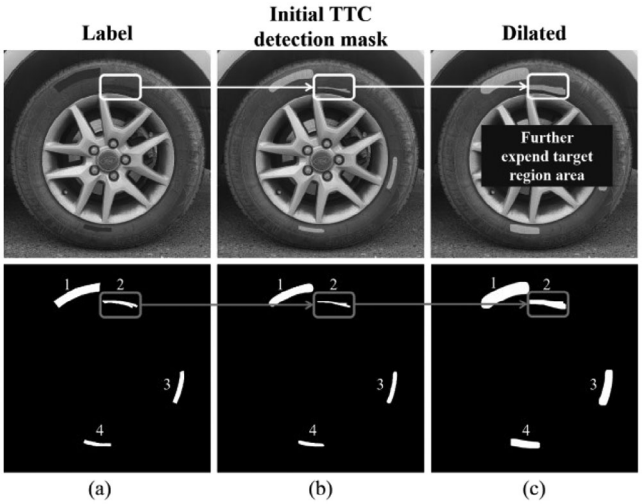

FIGURE 6 Comparison of image dilation effect: (a) label TTC region, (b) initial TTC detection region, and (c) dilated TTC region.

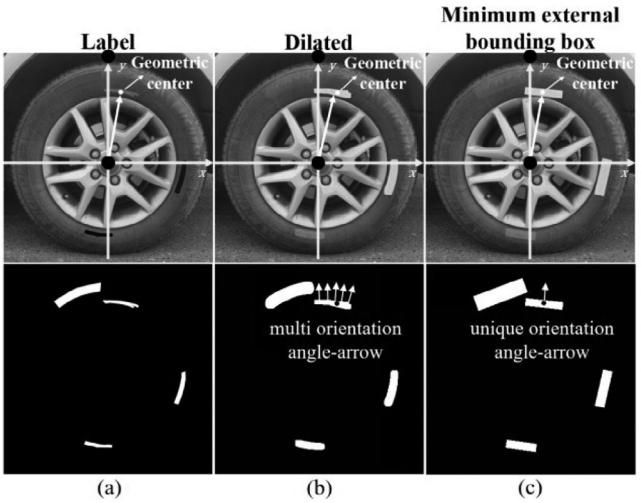

FIGURE 7 Comparison of the geometric center points: (a) label TTC region, (b) dilated TTC region, and (c) Minimum External Bounding Box (MEBB) region.

pre-processing techniques are designed to refine the segmented output from the TTC detection model. The process involves the following steps.

3.2.1 | Pre-process of TTC detection mask

First, the number of pixel blocks in the connected regions of all character segments is calculated using "cv2.contourArea()" function from the OpenCV API (Pulli et al., 2012). Subsequently, these regions are thus referred to as TTC "target regions." To ensure the integrity of all target regions and to prevent the omission of any characters within the target regions that were not adequately segmented, an image dilation algorithm from the OpenCV API (Gil & Kimmel, 2002; cv2.dilate(mask, (15, 15), iterations = 3)) is employed within a single connected character region. This approach is utilized to expand the area of the initial TTC detection polygon, thereby ensuring that the character regions adjacent to the segmented "target regions" that were not fully covered are more comprehensively included within the positioning polygon as shown in Figure 6.

In Figure 6, the dilation process enhances the segmentation results of TTC detection. The original results with limited segmentation effectiveness do not fully cover the entire TTC target region. However, the image dilation optimization algorithm expands the recognition area, ensuring that all the characters within the regions are included, which is conducive to the subsequent character recognition. To successfully utilize the proposed rotation method, it is imperative to accurately calculate the "geometric center coordinates" of the connected region and the "unique normal orientation angle" within the TTC target region

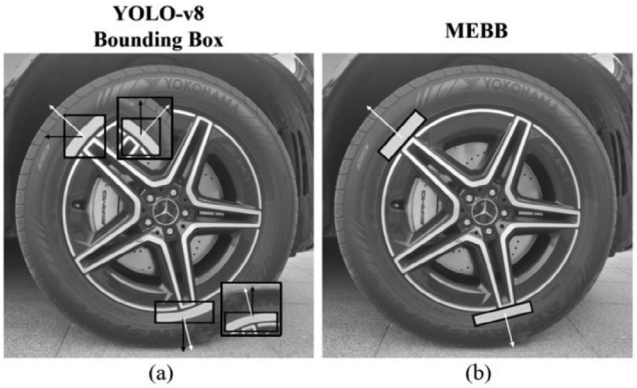

FIGURE 8 Comparison of MEBB and YOLO-v8 detected bounding boxes: (a) YOLO BB and (b) MEBB.

as shown in Figure 7c with white arrow. In contrast, the dilated TTC region in Figure 7b exhibits multiple normal orientation angles, which cannot be uniquely defined.

Moreover, in Figure 7b, the area of the dilated TTC region is not uniformly distributed, with more than half of the area concentrated on the right side. Consequently, the geometric center is skewed to the right. However, after generating a bounding box, the geometric center realigns closely with the true geometric center of the label TTC region in Figure 7a.

It should be noted that the proposed MEBB is more appropriate than the YOLO-v8 detected bounding box in this study. In Figure 8a, the bounding box generated by the YOLO model contains many unnecessary non-character areas and can only be a horizontal box, and its normal orientation angle is either 0 degrees or 90 degrees. Therefore, if the TTC region is rotated based on the YOLO

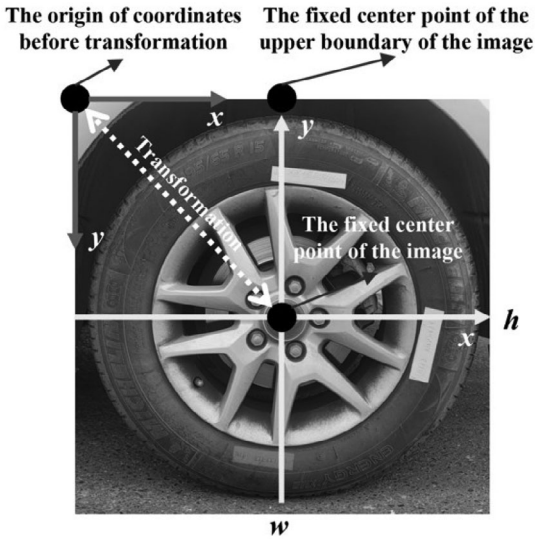

FIGURE 9 The image coordinate system of TSI.

BB's orientation by black arrow, it cannot remain upright after rotation, which negatively impacts the computational efficiency of the subsequent TTC recognition model.

3.2.2 | TTC detection mask rotation algorithm

After noise removal, image dilation, and MEBB generation, the tire characters are completely enclosed within the TTC target regions. Hence, TTC target regions that are already correctly oriented meet the basic recognition conditions. However, TTC regions that remain inclined cannot yet be recognized. This challenge is mainly due to the peculiarity of tire centripetal characters. Currently, no existing algorithm or model in the field of OCR is capable of directly recognizing inverted or tilted characters. To address this problem, this paper proposes a general algorithm to rotate the tire character to the correct orientation. This method is straightforward and avoids the computational cost and instability associated with complex transformations of TTC regions used in previous studies. The proposed TTC rotation method contains three steps.

First, the pixel width w and pixel height h of the image are calculated to determine "the fixed center point of the upper boundary of the image" and "the fixed center point of the image." The line connecting these two points is taken as the y-axis, while the line perpendicular to this is designated as the x-axis, thereby establishing the image coordinate system, as shown in Figure 9. Then, to facilitate the implementation of the algorithm, a transformation of the coordinate system is required for the coordinates of the fixed center point of the upper boundary $C_{\text{up}} = (w/2,0)$ and the fixed center point of the image $C_{\text{center}} = (w/2,h/2)$ in the original image coordinate. The transformation equations are:

$$x' = (x - \Delta x) \tag{1}$$

$$y' = -(y - \Delta y) \tag{2}$$

where $\Delta x = +w/2$, $\Delta y = +h/2$, and C_{up} and C_{center} are:

$$C_{\text{up}} = \left(0, \frac{h}{2}\right) \tag{3}$$

$$C_{\text{center}} = (0,0) \tag{4}$$

Finally, based on the transformed coordinate system, the central coordinate of each bounding box of the TTC target regions is calculated as shown in Figure 10a. Next, as depicted in Figure 10b, a line is drawn connecting the center of the TTC target region to a fixed central point. The clockwise angle between this line and the image's vertical medial y-axis is then measured as shown in Figure 10c.

Following the obtained rotation angle, the characters can be easily rotated to the top by determining the quadrant of the coordinate system, with the angle calculated using basic trigonometric function and quadrantal angle calculations. The pseudo-code for the rotation algorithm is shown in Algorithm 1. The inputs are tire_image_center_x (ticx): x coordinates of the center point of the tire image; tire_ image_center_ y (ticy): y coordinates of the center point of the tire image; mask_center_x (mcx): x coordinates of the rectangular mask character area; mask_center_ y (mcy): y coordinates of the rectangular mask character area, and the output is φ: the clockwise

ALGORITHM 1 TTC region rotation

for ***mcx***, ***mcy*** **in masks**:
$\Delta y = mcy - ticy$
$\Delta x = mcx - ticx$
$L = \sqrt{(\Delta x)^2 + (\Delta y)^2}$
$\cos\alpha = \frac{\Delta x}{L}$
$\sin\alpha = \frac{\Delta y}{L}$
If $\sin\alpha \geq 0$and $\cos\alpha \geq 0$ (1 quadrant)
$\alpha = 270^\circ + \arctan(|\frac{\Delta y}{\Delta x}|)$
elseif $\sin\alpha \geq 0$and $\cos\alpha < 0$ (2 quadrant)
$\alpha = 90^\circ - \arctan(|\frac{\Delta y}{\Delta x}|)$
elseif $\sin\alpha < 0$and $\cos\alpha < 0$ (3 quadrant)
$\alpha = 90^\circ + \arctan(|\frac{\Delta y}{\Delta x}|)$
elseif $\sin\alpha < 0$and $\cos\alpha \geq 0$ (4 quadrant)
$\alpha = 270^\circ - \arctan(|\frac{\Delta y}{\Delta x}|)$
end

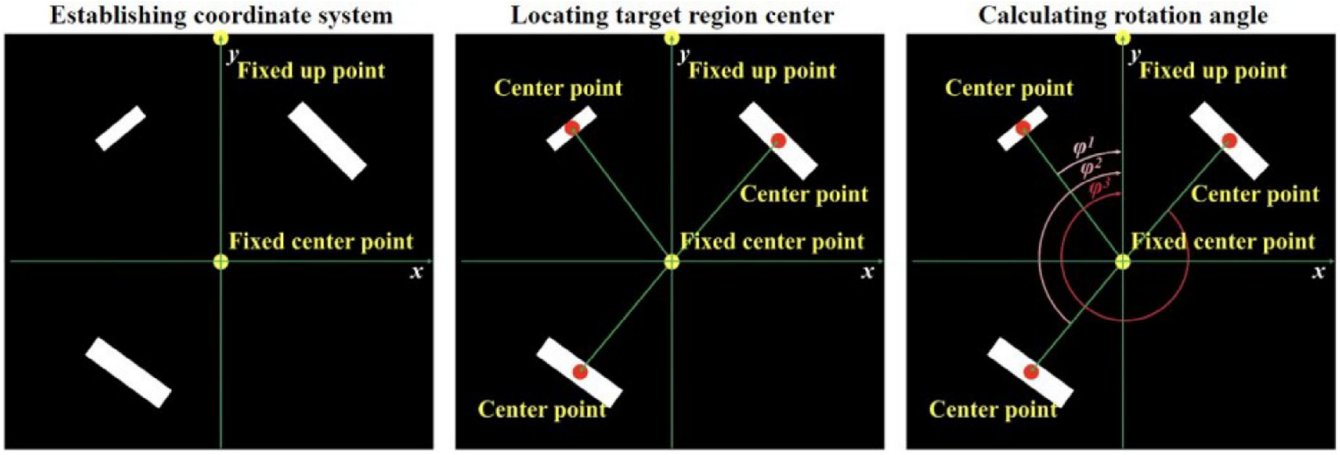

FIGURE 10 Schematic diagram of the character rotation algorithm: (a) calculating center points, (b) locating target center region, and (c) calculating rotation angle.

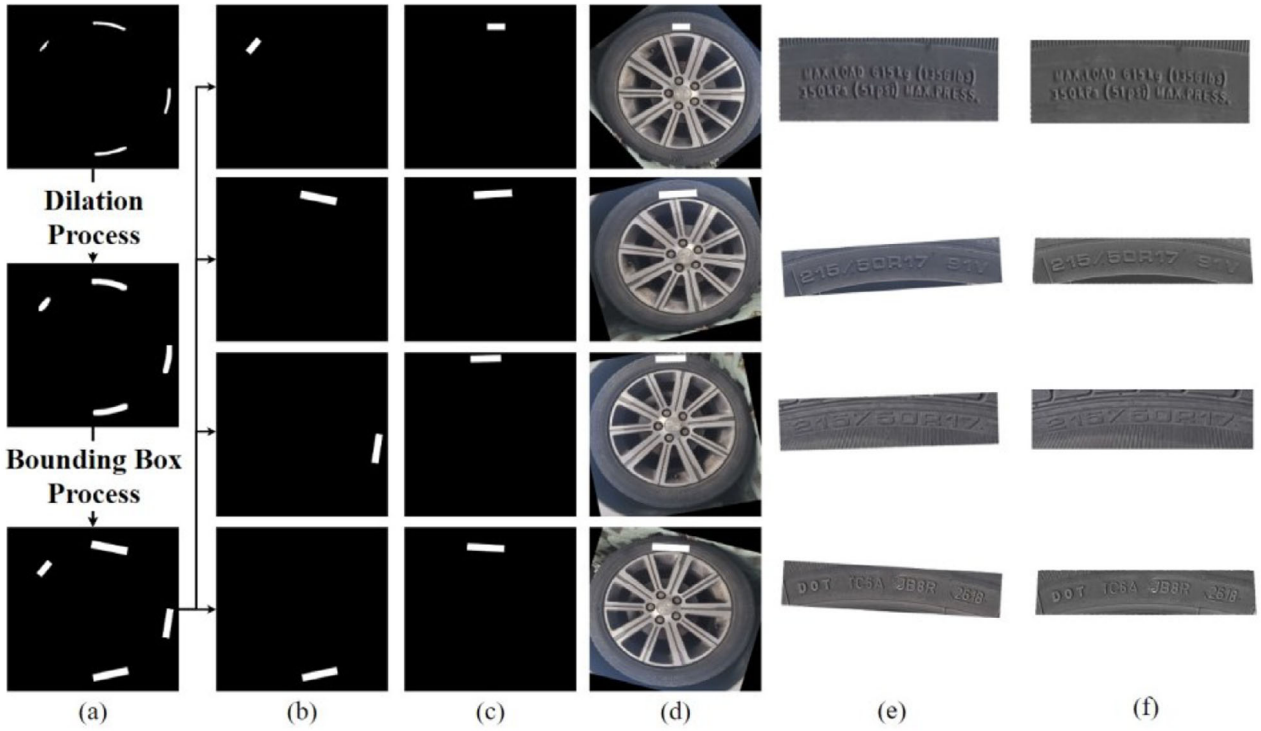

FIGURE 11 TTC rotation algorithms: (a) mask image pre-processing, (b) mask of each TTC target region, (c) rotated mask of TTC target region, (d) rotated image of TTC target region, (e) skewed TSCI, and (f) rectified.

rotation angle required to align the mask region with the positive y-axis.

The overall flowchart of the TTC detection algorithm is shown in Figure 11. In Figure 11c,d, each TTC region is precisely rotated clockwise by angle φ and aligned as closely as possible with the horizontal axis before being further cropped for TTC recognition. However, some TTC target regions remain slightly skewed as shown in Figure 11e. To correct this, an image rectification method based on horizontal projection (cv2.warpAffine (RoI, image size)) is used, aligning the TTC region to the horizontal axis, as shown in Figure 11f. By combining the image segmentation method with the proposed rotation strategy, the prerequisite for TTC recognition technology is satisfied and improved within a straightforward framework.

3.3 | TTC recognition based on object detection

As presented above, the processes of TTC detection, noise filtering, target region dilation, bounding box generation, and positive rotation of TTC regions provide a fundamental dataset for the development of the TTC recognition model. The shapes or fonts of characters on the tire sidewalls can vary significantly, even within the same character category, due to differences in brand, manufacturing, and tire type. Hence, the conventional OCR techniques, which are designed for uniform fonts, are not well-suited for recognizing tire characters. However, the variety of tire characters is limited, consisting only of Arabic numerals, alphabets, and a few special symbols, giving a total of 48 classes. This paper contends that the proven performance of the YOLO series network in the classification tasks is sufficient to enable YOLO to accurately classify different types of TTC, and its generalization ability is robust enough to correctly detect various shapes or fonts within the same type of TTC. Furthermore, the size of tire character varies significantly across different types of TTC, especially on different tires. Fortunately, YOLO-v8's anchor-free mechanism (Zhi Tian et al., 2020) facilitates effective training and detection of TTC data of varying sizes and shapes. In contrast, the traditional anchor-based mechanism hinders the model's generalization across objects of different scales and consumes more computational resources (Z. He et al., 2022). This efficiency allows YOLO to be easily deployed on the proposed IoT device with minimal edge inference costs. Moreover, the "Decoupled-Head" detection structure, combined with distributional focal loss (DFL), improves the model's accuracy by minimizing learning conflicts between object "Classify" and "Detection" (Dong. Liu & Jiang, 2024; W. Liu et al., 2018; Zhang et al., 2023), which is beneficial to the 48 categories of object detection task used in this study. YOLOv8's detection head uses multi-scale feature maps (CV0, CV1, CV2). It splits into "box" and "cls" branches, with the "box" branch using DFL for better bounding box accuracy and the "cls" branch for class probabilities. This design merges predictions from all scales for enhanced performance as shown in Figure 5c.

Therefore, by utilizing YOLO-v8 obj model, this study leverages the superior speed and performance of the YOLO network for object detection tasks. The TTC recognition problem is thus transformed into a more straightforward handwriting classification or object detection problem, which reduces the development costs. This approach effectively diminishes the complexity of character recognition tasks and the need for large datasets. This approach avoids complex label annotation, enhances the model's relevance to the task, and endows high scalability for testing various methods. This facilitates dataset publication of the subsequent model performance optimization.

4 | VALIDATIONS

4.1 | TTC data acquisition and generation

To validate the proposed framework, experiments were carried out by field datasets collection and model comparison, and two sets of data were established for training the YOLO-v8 seg-based TTC detection and YOLO-v8 obj-based TTC recognition model.

4.1.1 | TTC detection datasets

To perform a comprehensive data collection, a total of 430 high-resolution (5120 × 5120) TSIs, with focal lengths from 8 to 12 mm using industrial cameras at three highway toll stations were used to make sure of the diversity of tires. The pass by vehicle travels at different speeds and distances, and only the vehicle traveling under 5 m/s and ranging within 1–1.5 m can be captured, and the focal lengths were manually changed by the distance. The TSI datasets encompass different backgrounds, outdoor lighting conditions, brand, types, sizes, and wear levels. Then, the "labelme" API (Russell et al., 2008) was used to accurately annotate four types of TTC regions: DOT Marker (Class 0), Large Size Mark (Class 1), Inflation Pressure and Load Index (Class 2), and Small Size Mark (Class 3). Second, these annotations were then converted from "json" files into a grayscale label image. In addition, data augmentation methods, including "Horizontal flipping," "Vertical flipping," "Gaussian blur," "Affine transformation," and "random deformation" were adopted to expand the dataset and the corresponding label by a factor of five times in 2150 original images and 2150 label images, which enhanced the variety of the data. The statistical information of TTC detection datasets before augmentation is shown in Figure 12.

In Figure 12, it can be observed that the samples of "Inflation Pressure and Load Index" are the most numerous. This is because this category consists of two different strings: one indicating the tire standard inflation pressure and the other indicating the information of the tire standard load index. Therefore, these two different strings in "Class 2" were segmented and labeled separately but are considered the same type of TTC. The "Small Size Mark" is an indispensable mark, but wear has caused "Class 3" to be missing in a few samples. The "DOT"

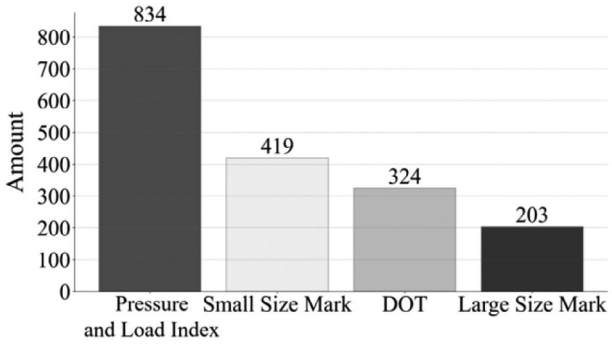

FIGURE 12 Statistical information of TTC detection datasets. DOT, Department of Transportation.

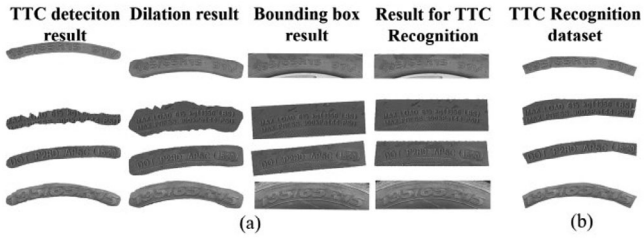

FIGURE 13 Comparison of TSCI: (a) TSCI during the TTC detection process and (b) TSCI data for model training.

mark is only applicable to the United States; therefore, tire data collected from other countries may miss this mark. Finally, "Large Size Mark" is not present on every tire, leading to its lower number of samples.

Since each type of TTC has a considerable number of samples, the instance segmentation method can be used to improve the model's understanding of different types of TTC regions and improve its accuracy rather than treating all character blocks as the same category. The datasets were divided into 8:1:1 for model training, validation, and testing.

4.1.2 | TTC recognition datasets

The TTC recognition datasets, or TSCI, are extracted spontaneously from TTC detection datasets, known as TSI. The TSCI is directly cropped from the original segmentation labels of TSI after being pre-processed using the proposed rotation algorithm as shown in Figure 13b. Here, a simple and high-quality method for generating TTC recognition datasets is achieved through a designed rotation algorithm.

After rotating and cropping 1376 TSCI data from 430 TSI images, the "labelimg" tool was utilized to annotate these images and categorize them into different groups. Specifically, special words like "DOT," "PRESSURE," and "LOAD" were marked as a whole object to decrease training cost and increase the model's discernment ability on

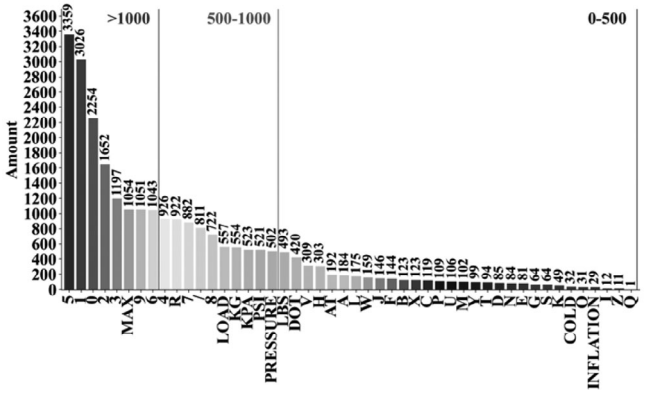

FIGURE 14 Statistical information of TTC recognition datasets.

these special words, while numbers and other characters were categorized separately. The categories are in following orders: "0," "1," "2," "3," "4," "5," "6," "7," "8," "9," "A," "B," "C," "D," "E," "F," "G," "H," "I," "J," "K," "L," "M," "N," "O," "P," "Q," "R," "S," "T," "U," "V," "W," "X," "Y," "Z," "/," "AT," "COLD," "DOT," "INFLATION," "KG," "KPA," "LOAD," "LBS," "MAX," "PRESSURE," "PSI." The categories include numerals, alphabets, abbreviations, and special tire-related terms. The dataset is organized into these 48 categories, and the statistical information of TTC recognition datasets is shown in Figure 14. In this study, plentiful data of numbers and special words have been collected. Except for a very rare letter, the distribution of collected letters is relatively uniform. The datasets were randomly divided with a ratio of 8:1:1 for model training and evaluation.

4.2 | Training results and discussion

4.2.1 | TTC detection validation

The training of TTC detection models is performed on a computer system with Ubuntu system, Intel(R) Xeon(R) Gold 6248R CPU at 3.00 GHz, Nvidia RTX 4080 GPU, Python, and PyTorch-based environment. Speed evaluations were performed on i9-13980HX and Nvidia RTX 4050 GPU. The training parameters were set as follows: image size of 1024, batch size of 16, initial learning rate of 0.01, and "Adam" optimizer. To build an optimal TTC detection model, five different parameters size baseline YOLO-v8 official models (X, L, M, S, and N) were trained for 100 epochs without data augmentation. These models were compared based on mean average precision (mAP), precision (P), recall (R), time (T), and model size to select the baseline model that provides the best balance between performance and speed, as shown in Table 1.

TABLE 1 Comparison of tire text codes (TTC) detection baseline model.

No.	Mean average precision (50) (mAP(50))	Precision (P)	Recall (R)	Time (T)/ms	Giga Floating-point Opeartion Per Second (GFLOPs)
X	0.6060	0.7274	0.6242	111.0	343.7
L	0.6073	0.7221	0.5973	63.5	220.1
M	0.5840	0.7731	0.5457	47.6	110.0
S	0.5348	0.6872	0.5736	42.2	42.4
N	0.5025	0.6514	0.5315	38.2	12.0
L-P6	0.2479	0.3787	0.3194	63.3	222.3
L-EFF	0.5842	0.7172	0.5452	66.0	228.5
L-Opt	0.8623	0.8812	0.8282	63.5	220.1

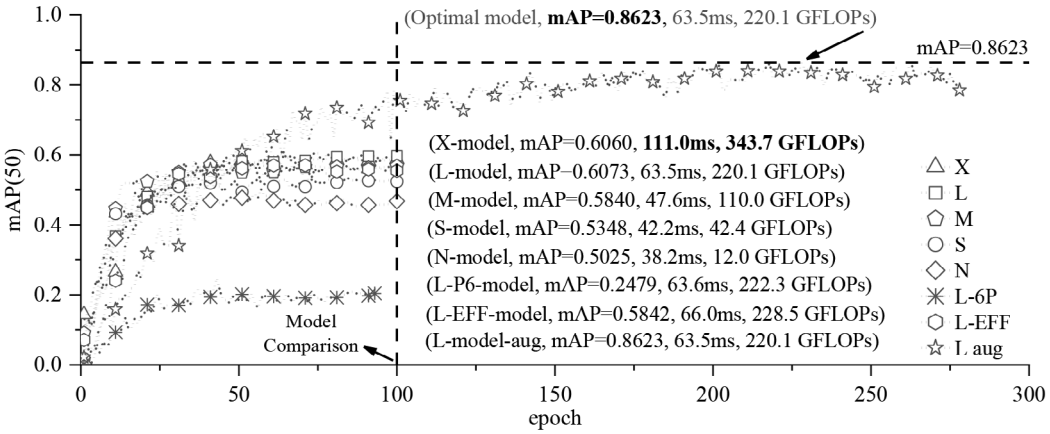

FIGURE 15 The mean average precision (50) (mAP(50)) comparison of TTC detection models.

The initial comparison results show that all the models were converged, with model L achieving the best performance, having an mAP(50) of 0.6073 and an acceptable inference speed of 63.5 ms. Model X exhibited similar performance but had a slower speed of 111.0 ms. Hence, model L was chosen for further comparison using different detection heads. Then, another 100 epochs were carried out to compare YOLO-v8-seg-L-P6 (an additional detection head output modification for small targets, based on P2-P5) and YOLO-v8-seg-L-Efficient head, aiming to identify a more computationally effective model. However, the efficient head showed limited improvement, and P6's performance was unsatisfactory. Therefore, the YOLO-v8-seg-L model is then chosen for 400 epochs training, repeated three times with online augmentation strategies. Here, the selected augmentation parameters are image rotation = +10°, translate = 0.1, scale = 0.5, flip up = 0.5, flip left and right = 0.5, copy and paste, and mosaic. The training results of the optimal TTC detection model on 400 epochs are shown in Figure 15.

The training process stopped early due to the patience parameter set at 100, indicating no further improvement, The results show that the YOLO-v8-seg-L model achieved the best performance with mAP(50) of 0.8632 and an acceptable speed, making it suitable for subsequent TTC detection. Figure 16 illustrates the effectiveness of the optimal model in TTC detection on selected test samples and demonstrates the image expansion algorithm's capability to optimize any imperfection in TTC detection. In Figure 16, the TTC region area of each sample was progressively compared to the area of the original label after detection by the model and continuous enhancement through dilation and bounding box generation as marked by auxiliary mark lines.

4.2.2 | TTC recognition validation

Base on the optimal TTC detection model and proposed highly stable pre-process and rotation methods, the optimal TTC recognition model is further established by comparing seven models. The datasets were resized to 640, with the Adam optimizer, an initial learning rate of 0.01, and a batch size of 16. The X, L, M, S, and N

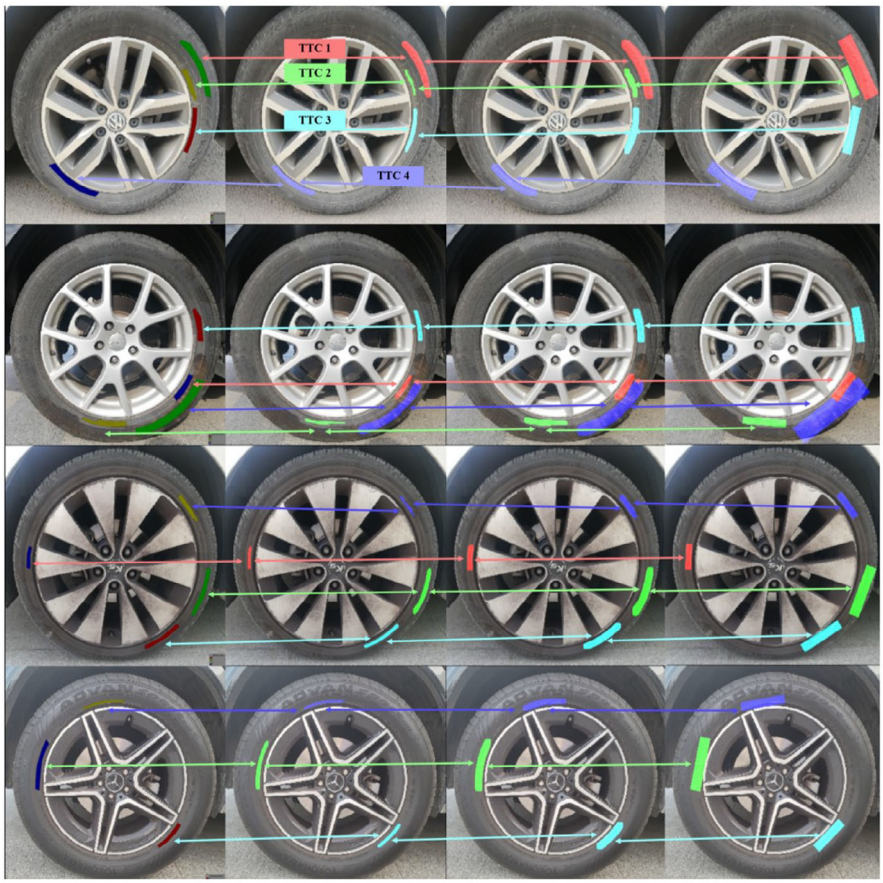

FIGURE 16 Effectiveness of the TTC detection model and enhanced results on selected testing datasets.

TABLE 2 Comparison of TTC recognition baseline model.

No.	mAP(50)	P	R	T/ms	Model size (GFLOPs)
X	0.9557	0.9451	0.9062	13.9	257.6
L	0.9320	0.9665	0.8785	10.4	165.0
M	0.9076	0.9402	0.8323	6.9	78.8
S	0.8269	0.893	0.7912	5.4	28.5
N	0.6917	0.8304	0.6387	5.2	8.1
X-AFPN	0.8310	0.9041	0.7990	18.0	229.5
X-EFF	0.9255	0.9635	0.8803	14.2	270.7
X-Opt	0.9710	0.9682	0.9449	13.9	257.6

baseline models were trained for 100 epochs to compare mAP and speed performance as shown in Table 2. The initial comparison shows that the speed differences among the baseline models were minimal. Model X shows the best performance in terms of mAP and recall, while model L shows the highest precision but has low recall, indicating overfitting. Hence, the Asymptotic Feature Pyramid Network (AFPN) structure (G. Yang et al., 2023), which may have stronger multi-scale TTC recognition capabilities and an efficient head, known for its computational advantages, was further tested on model X baseline over 100 epochs. As shown in Figure 17, the convergence of models shows that YOLO-v8-obj-X-EFF performs similarly to model L, while YOLO-v8-obj-X-AFPN shows lower capacity, comparable to model N, which could be due to scene non-adaptation. The YOLO-v8-obj-X model was then trained for 400 epochs, repeated three times, using online augmentation strategies. Here, the selected augmentation parameters included image rotation = +5°, translate = 0.2, scale = 0.5, flip up = 0.5, flip left and right = 0.5, copy and paste, mosaic, and another erasing parameter set to 0.4 to help the model focus on training targets. After training, YOLO-v8-obj-X achieved significant results with the highest mAP of 0.9710, precision of 0.96827, and recall of 0.9449.

The effectiveness of the optimal model in recognizing TTC is demonstrated in Figure 18 using selected untrained samples, which illustrate correctly recognized characters on various TTCs. The proposed approach efficiently and accurately recognizes TTCs by leveraging YOLO for object detection tasks. Transforming of the character recognition problem reduces complexity, resulting in high accuracy and recall rates. Combined with TTC detection (63.5 ms) and pre-processing (20 ms), the average speed of the whole framework is 100.5 ms.

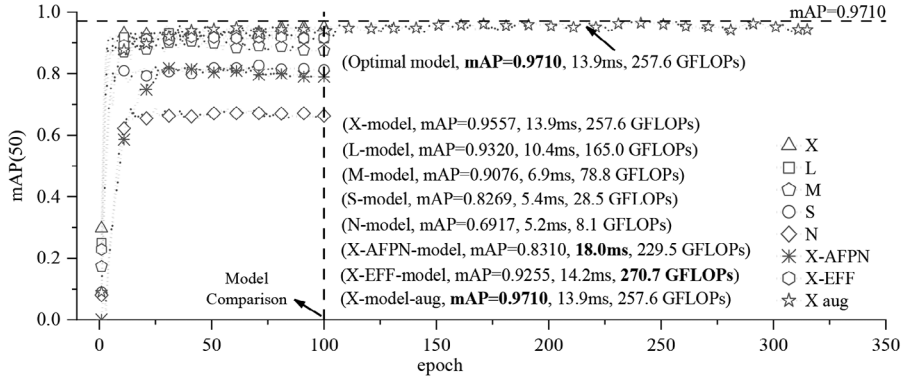

FIGURE 17 mAP(50) comparison of TTC recognition models.

FIGURE 18 Effectiveness of TTC recognition models. DOT, Department of Transportation.

Moreover, transforming the TTC recognition problem into an object detection task reduces the image quality requirement for the TSCI. In previous studies, researchers needed to develop a secondary object detection model to extract high-quality TSCI from rough TSCI. Yet, the proposed object detection-based TTC recognition system can directly perform satisfactory ICR tasks on rotated raw TSCI. This approach significantly reduces the computational overhead associated with TTC identification, facilitates edge deployment of the framework on IoT devices, minimizes pre-processing steps for TTC detection, and enhances the overall stability of identification tasks.

5 | AUTOMATIC SYSTEM FOR TTC IDENTIFICATION

5.1 | Portable IOT device

In a previous study, due to the limitation of the algorithms, the roadside devices (binocular devices) were required to take multiple photographs of the tires in the TTC identification task (Kazmi et al., 2020). Another study achieved TTC identification with monocular vision. However, due to the complexity of the computational models involved, outdoor deployment on edge devices was not pursued (F. Gao et al., 2021). Instead, the focus shifted toward a costly optical scanning method for indoor tire detection applications (Cheng et al., 2024). On the contrary, this paper presents a fully automated IoT device, named the "Portable TTC Identification Device," designed for outdoor application scenarios using monocular vision. The proposed device is equipped with basic waterproof and splash-proof features and is designed to be portable, easy to install and disassemble. To support effective field TTC identification, the device allows remote adjustment of image acquisition parameters, continuous triggering and acquisition, synchronized multi-trigger operations, and includes a portable power source for outdoor operation, along with comprehensive program error protection (Peng et al., 2023).

Figure 19 illustrates the overall appearance and key components of the device. It comprises components such as the light source, camera, trigger radar, and development board. (1) To realize automatic detection, a 20 Hz trigger radar of 25 watt is installed, capable of detecting passing vehicles at around 1.5 m. (2) A lens of 25 million pixels is utilized to ensure precise imaging within the same working distance as the radar. (3) An industrial camera powered by USB (Universal Serial Bus) with a high resolution of 5120 × 5120 was chosen to work seamlessly with the camera underexpose time of 300 ms. (4) A trigger strobe LED

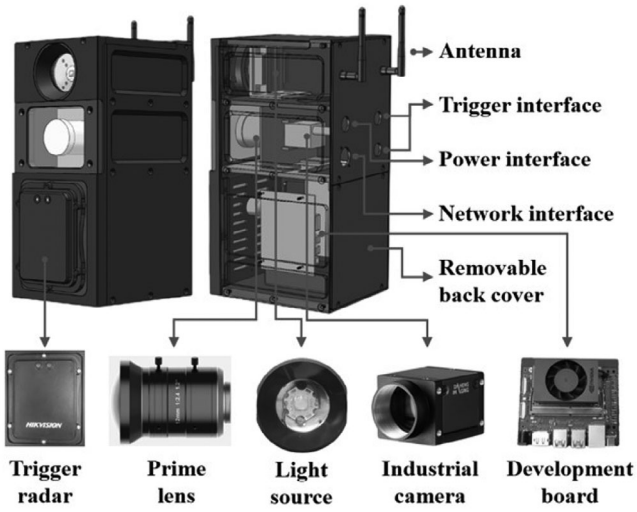

FIGURE 19 Diagram of proposed Internet of Things (IoT) device.

(Light-Emitting Diode) light of 50 watt, capable of emitting light with a diameter of 1.5 m at 1 m away, is incorporated to make the tire target in the image clear, facilitating the proposed algorithm and allowing the system to adapt to various lighting conditions. Testing showed that with the assistance of LED light, the device can capture vehicles traveling at a speed of 20 km/h without compromising the algorithm's detection accuracy. (5) A Jetson Orin NX NVIDIA development board of 25 watt with a 150-watt mobile power source set on the baseplate is exploited to process the TTC identification framework. The cost of this device is about $4000.

In practical scenarios, utilizing a trigger mechanism in the device is imperative to automate the identification process. A two-tiered triggering logic is employed to achieve this. The detailed implementation process is shown in Figure 20 to elucidate the operational principles. Initially, when a vehicle enters the testing area, the primary trigger radar detects its presence and counts the number of vehicles. As the vehicle's tires enter the detection zone, the secondary triggering mechanism is activated. During the preliminary experiment, a pulse trigger was chosen, and a single pulse was transmitted to the development board to control the camera for image capture. These images are then transmitted to the development board for edge computing using the previously proposed deep-learning framework. The system continuously monitors for new vehicles to test. If no new vehicle is detected, it processes to the next axle of the current vehicle until a new target vehicle is identified. A 2-s buffer period is incorporated to accommodate vehicles with longer axle distances. If no new tire targets are detected within this period, the system saves the recognition results for all axles of the vehicle and simultaneously checks for new incoming vehicles. The

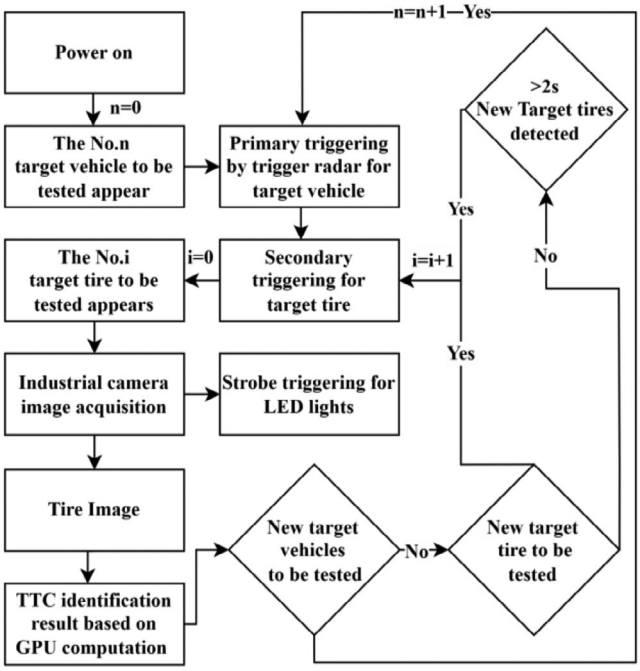

FIGURE 20 Workflow of IoT device.

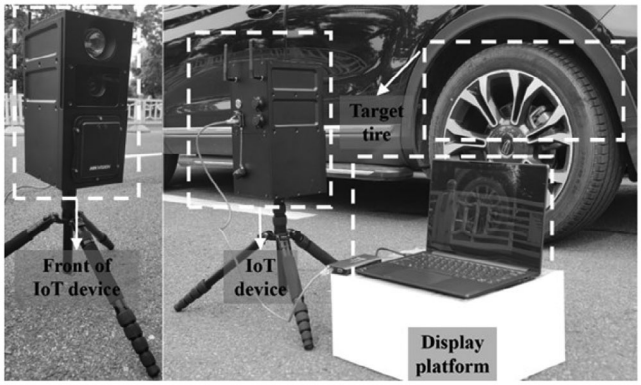

FIGURE 21 Proposed IoT device for field test.

workflow logic outlined in this paper effectively meets the requirements for TTC recognition in various operational scenarios.

5.2 | Field test of the proposed system

A field test was conducted to verify the proposed automatic TTC identification system, which integrated a deep learning-based TTC identification framework compiled in Python. The setup of the field test is shown in Figure 21. A laptop was used to enable image-based secondary triggering of the IoT device, control data acquisition, and display images from the device. The testing platform was positioned on the roadside on a sunny day with good lighting condition to perform TTC identification on passing vehicles. All tested vehicles are traveling at a speed of 5 m/s and within a distance between 1 and 1.5 m. The lens was adjusted to 1×1 m view at a distance of 1 m. The IoT device was on a tripod with a height of 50 cm. The display platform was a laptop with a CPU of i5-1135G7/16G.

The test results show that the computational speed of the integrated deep learning framework on a 16 GB Jetson Orin NX with a mobile power source is approximately: TTC detection took about 100 ms, TTC pre-process and rotation took 20 ms, and TTC recognition averaged 80 ms, resulting in a total average of roughly 200 ms.

This speed performance is superior to Kazmi's 1000–2000 ms (Kazmi et al., 2019, 2020), Gao's 407 ms (F. Gao et al., 2021) and Cheng's 9000 ms (Cheng et al., 2024). The results demonstrate that the system is efficient and reliable for industrial applications, with significant potential to meet real-time detection requirements in many scenarios. Figure 22 illustrates the field test results of the proposed framework on samples that were untrained during the model establishment.

In Figure 22, the green characters represent the targets redundantly identified, and the red characters indicate targets incorrectly identified. The results show that the TTC region detection model can accurately detect the TTC region without generating incorrect or superfluous targets. However, the TTC recognition model exhibits some deficiencies in recognition performance. This study argues that excessive "Dilation and Bounding box generation" may be the main cause of the redundant identification of green characters. Adjusting parameters could reduce the size of the TTC target region and minimize excessive recognition. The incorrectly identified red characters are attributed to inherent challenges in recognizing TTCs that are worn, tilted, small, or densely packed, highlighting the need for more extensive datasets to improve the model. Nevertheless, the TTC recognition model still provides valuable results for the interpretation of four types of tire text code.

6 | REMARKS

The proposed TTC identification system provides a practical, plug-and-play solution that can be readily adopted by engineers in various transportation scenarios. By installing the IoT device at toll stations, bridges, or roadside lanes and leveraging its built-in triggering mechanism, the system can automatically capture and process tire images on the edge, and relay recognized TTCs to a central operations platform in real time. Engineers can then examine the tire information for traffic operations management and maintenance planning. In additional,

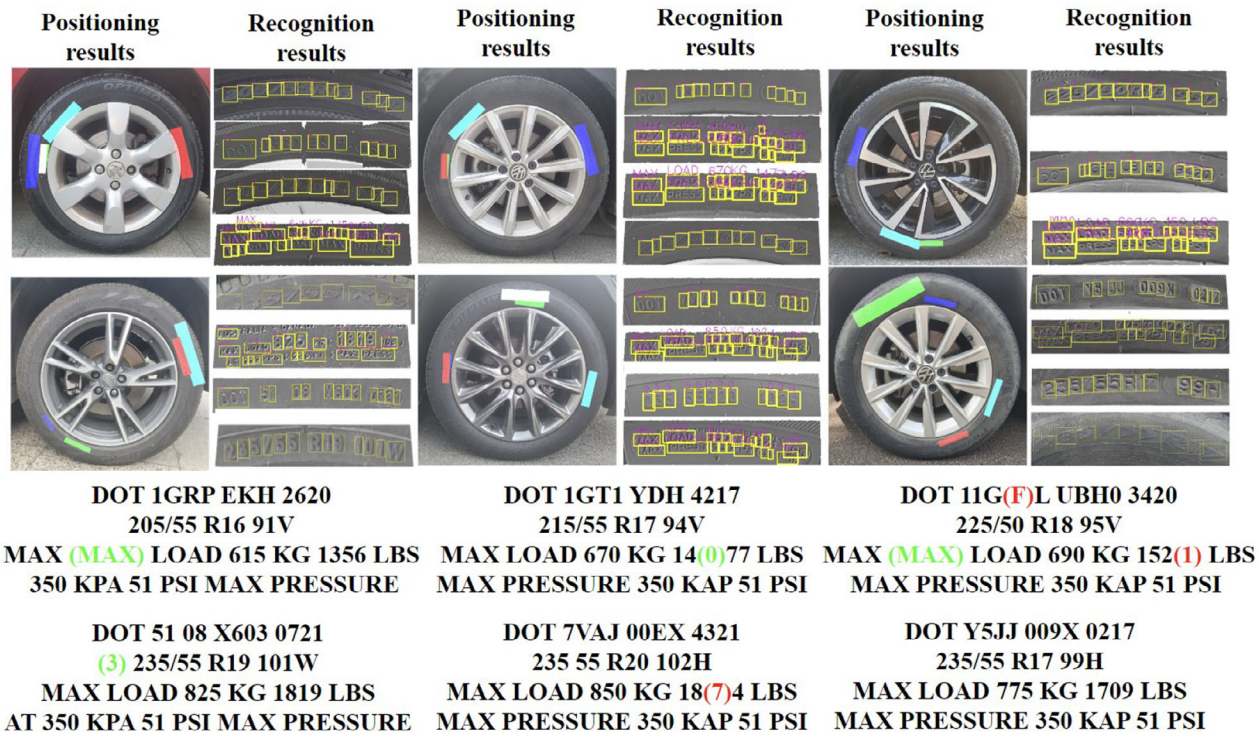

FIGURE 22 Field test results of the proposed framework on selected samples. DOT, Department of Transportation.

every tire has a unique serial number, and the recognized incomplete sequence can be cross-referenced with a tire database to assess character-missing rates, thereby estimating tire abrasion conditions for pass by drivers. Furthermore, in the non-contact weighing field (J. Feng et al., 2024; K. Gao et al., 2024), precise tire character data can significantly improve the accuracy of vehicle weight estimations, making it a valuable upgrade for existing non-contact weigh-in-motion methods. It should be noted that one device can be only used for one traffic lane, blocked vehicle cannot be tested, and toll station can be a solution for multi-monitoring with multi-devices. The proposed system can only monitor vehicles at 1–1.5 m away with vehicle speed under 5 m/s. More advanced camera and autofocus lens could provide better monitoring view with further distance.

Compared in Table 3, the proposed approach stands out in speed (0.2 s per tire), recognition accuracy (96.82% on four TTC categories), ease of deployment (a single monocular IoT camera), and production method of datasets. Different from other methods that either require multi-camera setups, expensive 3-D camera console, extensive pre-processing, or are too slow for practical roadside deployment, this framework offers a current state-of-the-art outdoor solution for TTC identification. Additionally, further enhancements such as enlarging the dataset coverage and combining language model-based TTC restore for extreme wear or heavily faded characters would help refine the accuracy. Moreover, a super-resolution-based method (García-Aguilar et al., 2023) can be adopted to improve the segmentation result.

7 | CONCLUSION

This study proposed an efficient automatic TTC identification framework integrated into a "Portable TTC Identification Device." The training strategies and the dataset production are straightforward to implement, making the framework easy to deploy, propagate, and iteratively upgraded.

1. An instance segmentation method is adequately used to detect four types of TTC regions at a pixel level. The optimal YOLO-v8-seg-L model is selected, demonstrating the best performance with an mAP(50) of 0.8623 and a speed of 63.5 ms, effectively preventing redundant TTC region detection and supporting the subsequent pre-processing and recognition of detected TTC region.
2. A straightforward TTC rotation algorithm is designed to perform general TTC orientation correction by determining each TTC region's quadrant and rotation angle relative to the *y*-axis. This algorithm integrates seamlessly with the proposed TTC detection method and reduces the requirement of model performance and

TABLE 3 Comparison of tire text detection systems.

	Method	Model	Inference speed per tire	Recognition accuracy	Rotation	Device	Dataset	Object
Kazmi **Outdoor**	Intelligent character recognition (ICR) detection	Light convolutional neural network (CNN) with Rectified Linear Unit (ReLU)	1–2 s	80.0% on Department of Transportation (DOT) only	√	Binocular Internet of Things (IoT) camera	1) "DOT"/700k	1 TTC
Gao **Outdoor**	ICR detection	YOLOv4 +CRAFT +MORANv2	0.4 s	94.5% on DOT only	√	Monocular industrial camera	1) TID/NA 2) TCCID NA 3) RTCD/NA Not available	1 TTC
Cheng **Indoor**	ICR detection	CTPN +CRNN	3–9 s	>96% on 4 TTCs	√	3D-camera with rotary console	1) Tire/1.8k 2) TTC/1k Not available	4 TTCs
Proposed method **Outdoor**	ICR segmentation +detection	YOLOv8-Segmentation +YOLOv8-Detection	0.2 s	96.82% on 4 TTCs	√	Monocular IoT camera	1) Tire/430 2) TTC/1.3k Fully available	4 TTCs

Note: Kazmi (Kazmi et al., 2020); Gao (F. Gao et al., 2021); Cheng (Cheng et al., 2024).

datasets coverage, offering stable performance and aiding in the production of TSCI datasets that are very rare.

3. The optimal YOLO-v8-obj-X model with an mAP(50) of 0.9710 and a speed of 13.9 ms was selected for recognizing characters of four types of TTC with high accuracy, eliminating the need for secondary TTC region detection and re-certification. Finally, a fully integrated automatic IoT device is developed to cost-effectively compass the proposed TTC identification framework in real scene applications.

However, error missing and redundancy in TTC identification still occur occasionally. Additional datasets are needed to enhance model performance, and other baseline models can be further explored for potential outcomes. Implementing mechanisms like "Self-Attention," "Class-Attention," or "RoI" may improve the model's focusing ability on relevant and rare targets and reduce redundancy and error detections. While the system shows promise, its deployment and performance in more diverse and challenging environments (e.g., high-speed vehicles, different lighting conditions) remain to be thoroughly tested. Future work could also include langue model-based semantic analysis or restoration of TTC recognition results.

ACKNOWLEDGMENTS

All the datasets are available on request.

REFERENCES

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). *YOLOv4: Optimal speed and accuracy of object detection*. arXiv preprint arXiv:2004.10934. https://arxiv.org/abs/2004.10934

Cheng, A., Lu, S., & Gao, F. (2024). Anomaly detection of tire tiny text: Mechanism and method. *IEEE Transactions on Automation Science and Engineering*, *21*(2), 1911–1928.

Dan, H.-C., Yan, P., Tan, J., Zhou, Y., & Lu, B. (2024). Multiple distresses detection for asphalt pavement using improved you only look once algorithm based on convolutional neural network. *International Journal of Pavement Engineering*, *25*(1), 2308169.

Feng, J., Gao, K., Zhang, H., Zhao, W., Wu, G., & Zhu, Z. (2024). Non-contact vehicle weight identification method based on explainable machine learning models and computer vision. *Journal of Civil Structural Health Monitoring*, *14*, 843–860.

Feng, M. Q., & Leung, R. Y. (2021). Application of computer vision for estimation of moving vehicle weight. *IEEE Sensors Journal*, *21*(10), 11588–11597.

Feng, M. Q., Leung, R. Y., & Eckersley, C. M. (2020). Non-contact vehicle weigh-in-motion using computer vision. *Measurement*, *153*, 107415.

Gao, F., Ge, Y., Lu, S., & Weng, L. (2021). Vehicle tire text reader: Text spotting and rectifying for small, curved, and rotated characters. *IEEE Transactions on Instrumentation and Measurement*, *70*, 5016512.

Gao, F., Li, S., You, H., Lu, S., & Xiao, G. (2020). Text spotting for curved metal surface: Clustering, fitting, and rectifying. *IEEE Transactions on Instrumentation and Measurement*, *70*, 5000212.

Gao, K., Zhang, H., & Wu, G. (2024). A multispectral vision-based machine learning framework for non-contact vehicle weigh-in-motion. *Measurement*, *226*, 114162.

García-Aguilar, I., García-González, J., Luque-Baena, R. M., López-Rubio, E., & Domínguez, E. (2023). Optimized instance segmentation by super-resolution and maximal clique generation. *Integrated Computer-Aided Engineering*, *30*(3), 243–256.

Garrido-Hidalgo, C., Roda-Sanchez, L., Fernández-Caballero, A., Olivares, T., & Ramírez, F. J. (2023). Internet-of-Things framework for scalable end-of-life condition monitoring in remanufacturing. *Integrated Computer-Aided Engineering*, *31*(1), 1–17.

Gil, J. Y., & Kimmel, R. (2002). Efficient dilation, erosion, opening, and closing algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(12), 1606–1617.

Girshick, R. (2015). Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile (pp. 1440–1448).

Ham, Y. K., Kang, M. S., Chung, H. K., Park, R.-H., & Park, G. T. (1995). Recognition of raised characters for automatic classification of rubber tires. *Optical Engineering*, *34*(1), 102–109.

He, Z., Chen, W., Zhang, J., & Wang, Y.-H. (2024). Crack segmentation on steel structures using boundary guidance model. *Automation in Construction*, *162*, 105354.

He, Z., Jiang, S., Zhang, J., & Wu, G. (2022). Automatic Damage Detection Using Anchor-Free Method and Unmanned Surface Vessel. *Automation in Construction*, *133*, 104017.

Jocher, G., Chaurasia, A., & Qiu, J. (2023). *Ultralytics YOLOv8*. https://github.com/ultralytics/ultralytics

Ju, H., Shi, H., Shen, W., & Deng, Y. (2024). An accurate and low-cost vehicle-induced deflection prediction framework for long-span bridges using deep learning and monitoring data. *Engineering Structures*, *310*, 118094.

Kamjoo, E., Rostami, A., Fakhrmoosavi, F., & Zockaie, A. (2024). A simulation-based approach for optimizing the placement of dedicated lanes for autonomous vehicles in large-scale networks. *Computer-Aided Civil and Infrastructure Engineering*, *39*(20), 3011–3029.

Kazmi, W., Nabney, I., Vogiatzis, G., Rose, P., & Codd, A. (2020). An efficient industrial system for vehicle tyre (tire) detection and text recognition using deep learning. *IEEE Transactions on Intelligent Transportation Systems*, *22*(2), 1264–1275.

Kazmi, W., Nabney, I., Vogiatzis, G., Rose, P., & Codd, A. (2019). Vehicle tire (tyre) detection and text recognition using deep learning. *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, Vancouver, BC, Canada (pp. 1074–1079).

Kong, X., Zhang, J., Wang, T., Deng, L., & Cai, C. S. (2022). Non-contact vehicle weighing method based on tire-road contact model and computer vision techniques. *Mechanical Systems and Signal Processing*, *174*, 109093.

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Li, D., Liu, M., Yang, L., Wei, H., & Guo, J. (2024). A non-contact identification method of overweight vehicles based on computer vision and deep learning. *Computer-Aided Civil and Infrastructure Engineering*, *39*(22), 3452–3476.

Li, H., Zhang, H., Zhu, H., Gao, K., Liang, H., & Yang, J. (2024). Automatic crack detection on concrete and asphalt surfaces using semantic segmentation network with hierarchical Transformer. *Engineering Structures*, *307*, 117903.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI (pp. 2117–2125).

Liu, C., Dong, Y., Wei, Y., Wang, J., & Li, H. (2022). Image detection and parameterization for different components in cross-sections of radial tires. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, *236*(2-3), 287–298.

Liu, D., & Jiang, M. (2024). *MT2ST: Adaptive multi-task to single-task learning*. arXiv preprint arXiv:2406.18038. https://arxiv.org/abs/2406.18038

Liu, D., Waleffe, R., Jiang, M., & Venkataraman, S. (2024). *GraphSnapShot: Graph machine learning acceleration with fast storage and retrieval*. arXiv preprint arXiv:2406.17918. https://arxiv.org/abs/2406.17918

Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path aggregation network for instance segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT (pp. 8759–8768).

Liu, W., Liu, Z., Yu, Z., Dai, B., Lin, R., Wang, Y., Rehg, J. M., & Song, L. (2018). Decoupled networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT.

Pan, X., Yang, T., Xiao, Y., Yao, H., & Adeli, H. (2023). Vision-based real-time structural vibration measurement through deep-learning-based detection and tracking methods. *Engineering Structures*, *281*, 115676.

Peng, Z., Li, J., & Hao, H. (2023). Development and experimental verification of an IoT sensing system for drive-by bridge health monitoring. *Engineering Structures*, *293*, 116705.

Pulli, K., Baksheev, A., Kornyakov, K., & Eruhimov, V. (2012). Real-time computer vision with OpenCV. *Communications of the ACM*, *55*(6), 61–69.

Rafiei, M. H., & Adeli, H. (2017). A new neural dynamic classification algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(12), 3074–3083.

Redmon, J., & Farhadi, A. (2018). *YOLOv3: An incremental improvement*. arXiv preprint arXiv:1804.02767. https://arxiv.org/abs/1804.02767

Russell, B. C., Torralba, A., Murphy, K. P., & Freeman, W. T. (2008). LabelMe: A database and web-based tool for image annotation. *International Journal of Computer Vision*, *77*, 157–173.

Scholz, O., & Koehler, T. (2012). *Reading DOT codes on passenger car tires*. SAE Technical Paper 2012-01-0797. SAE.

Shi, B., Bai, X., & Belongie, S. (2017). Detecting oriented text in natural images by linking segments. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI (pp. 2550–2558).

Sitton, J. D., Rajan, D., & Story, B. A. (2024). Damage scenario analysis of bridges using crowdsourced smartphone data from passing vehicles. *Computer-Aided Civil and Infrastructure Engineering*, *39*(9), 1257–1274.

Sukprasertchai, S., & Suesut, T. (2016). Real-time surface acquisition of tire sidewall for reading embossed information. *Proceedings of the International MultiConference of Engineers and Computer Scientists*, Hong Kong.

Tian, Z., Shen, C., Chen, H., & He, T. (2020). FCOS: A simple and strong anchor-free object detector. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(4), 1922–1933.

Tian, Z., Shu, M., Lyu, P., Li, R., Zhou, C., Shen, X., & Jia, J. (2019). Learning shape-aware embedding for scene text detection.

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA (pp. 4234–4243).

Wang, C.-Y., Liao, H.-Y. M., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W., & Yeh, I.-H. (2020). CSPNet: A new backbone that can enhance learning capability of CNN. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, Seattle, WA (pp. 390–391).

Wang, W., Xie, E., Li, X., Hou, W., Lu, T., Yu, G., & Shao, S. (2019). Shape robust text detection with progressive scale expansion network. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, CA (pp. 9336–9345).

Xu, Y., Wang, Y., Zhou, W., Wang, Y., Yang, Z., & Bai, X. (2019). TextField: Learning a deep direction field for irregular scene text detection. *IEEE Transactions on Image Processing*, *28*(11), 5566–5579.

Yang, G., Lei, J., Zhu, Z., Cheng, S., Feng, Z., & Liang, R. (2023). AFPN: Asymptotic feature pyramid network for object detection. *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Maui, HI (pp. 2184–2189).

Yang, Y., Xu, W., Gao, A., Yang, Q., & Gao, Y. (2024). Bridge damage identification based on synchronous statistical moment theory of vehicle–bridge interaction. *Computer-Aided Civil and Infrastructure Engineering*, *39*(24), 3741–3768.

Yeum, C. M., Dyke, S. J., Basora Rovira, R. E., Silva, C., & Demo, J. (2016). Acceleration-based automated vehicle classification on mobile bridges. *Computer-Aided Civil and Infrastructure Engineering*, *31*(11), 813–825.

Yin, X.-C., Pei, W.-Y., Zhang, J., & Hao, H.-W. (2015). Multi-orientation scene text detection with adaptive clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(9), 1930–1937.

Zhang, H., Gao, K., Huang, H., Hou, S., Li, J., & Wu, G. (2023). Fully decouple convolutional network for damage detection of rebars in RC beams. *Engineering Structures*, *285*, 116023.

How to cite this article: Zhang, H., Gao, K., Hou, Y., Domaneschi, M., & Noori, M. (2025). Portable IoT device for tire text code identification via integrated computer vision system. *Computer-Aided Civil and Infrastructure Engineering*, 1–20. https://doi.org/10.1111/mice.13438