

Article

A Lightweight Barcode Detection Algorithm Based on Deep Learning

Jingchao Chen , Ning Dai * , Xudong Hu and Yanhong Yuan 

Key Laboratory of Modern Textile Machinery & Technology of Zhejiang Province, Zhejiang Sci-Tech University, Hangzhou 310018, China

* Correspondence: zstudn@zstu.edu.cn

Abstract: For existing situations of missed detections, false detections, and repeated detections in barcode detection algorithms in real-world scenarios, a barcode detection algorithm based on improved YOLOv8 is proposed. The EfficientViT block based on a linear self-attention mechanism is introduced into the backbone of the original model to enhance the model's attention to barcode features. In the model's neck, linear mapping and grouped convolution are used to improve the C2f module, and the ADown convolution block is utilized to modify the model's downsampling, which reduces the model's parameters and computational cost while improving the efficiency of model feature fusion. Finally, the reconstruction of the model's detection head and the modification of the loss function are implemented to enhance the model's training quality and reduce the model's error in barcode detection. Experimental results indicate that the improved model exhibits an increase of 1.8% in recall rate and 1.9% in mAP50:95 for barcode localization and classification. The FPS is improved by 40 frames per second. The model's parameter count is reduced by 74.2%, and FLOPs are decreased by 79.6%. Furthermore, the proposed model outperforms other models in terms of model size and barcode detection accuracy.

Keywords: barcode detection; YOLOv8; linear self-attention mechanism; linear mapping; grouped convolution



Citation: Chen, J.; Dai, N.; Hu, X.; Yuan, Y. A Lightweight Barcode Detection Algorithm Based on Deep Learning. *Appl. Sci.* **2024**, *14*, 10417. <https://doi.org/10.3390/app142210417>

Academic Editor: Luis Javier Garcia Villalba

Received: 5 October 2024
Revised: 8 November 2024
Accepted: 8 November 2024
Published: 12 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As an exceptionally efficient method for information storage and transmission, one-dimensional codes and two-dimensional codes have profoundly simplified the procedures involved in logistics tracking, inventory management, and commodity certification, thereby bringing unprecedented convenience and operational efficiency to various sectors and industries across diverse fields [1–7]. These advanced coding technologies not only achieve rapid data entry and automatic identification with a remarkably high degree of accuracy and reliability, but they also have the capacity to encapsulate a vast array of content, ranging from straightforward commodity information to intricate data links, through their innovative and unique coding methodologies. This demonstrates their impressive information processing capabilities. Relevant and in-depth research has clearly shown that the advancement of barcode recognition technology holds immense significance for improving the efficiency and accuracy of information processing, accelerating the digital transformation of various industries, optimizing supply chain management processes, fostering the application and development of emerging technologies, and expanding the realm of potential application fields.

In recent times, an increasing number of researchers have started to direct their attention towards the practical applications of barcodes in real-world scenarios. Identification technology must not only possess the capability to detect barcodes in real-time with precision and accuracy, but it must also ensure that the applications utilizing this technology consume minimal power. With this in mind, this paper introduces a novel barcode recognition algorithm specifically designed to address the challenges associated with barcode

detection and analysis in real-life situations. The primary contributions of this paper can be summarized as follows:

1. This paper proposes a high-speed and accurate barcode recognition algorithm based on a lightweight barcode detection model. The model achieves real-time barcode detection with a small number of parameters, low computational complexity, high accuracy, and rapid inference capabilities.
2. While focusing on minimizing both parameter count and computational complexity of the model, this paper also enhances the model's ability to dynamically adjust contextual feature relationships. To achieve this, we introduce a linear attention mechanism and efficient convolutional operations into the network structure, thereby improving the feature extraction capabilities of the feature fusion convolutional blocks and pooling layers, as well as the operational efficiency of the convolutional modules. Furthermore, we reconstruct the detection head and adjust the model's loss function to enhance the convergence speed and training quality of the model.
3. The effectiveness of the proposed algorithm and model is verified in this paper. Comparative experimental results demonstrate that our model outperforms other models in terms of detection accuracy, model size, and detection speed.

2. Related Work

Common barcode recognition technologies can be divided into two types: laser scanner-based and camera-based. The former is mostly used for one-dimensional barcode recognition, with slower recognition speed and shorter device recognition distance. Therefore, this paper only considers camera-based barcode detection. Camera-based detection methods can further be categorized into traditional digital image processing-based and deep learning-based approaches.

Initially, researchers employed traditional digital image processing techniques to achieve barcode recognition, yielding some promising research advancements. Ohbuchi [8] simplified the image information by setting a gray threshold for the input barcode image and then employed a spiral scanning strategy to precisely locate the critical black bars. By calculating the directions perpendicular to these black bars, they effectively sampled the barcode area. Wachenfeld [9] developed a one-dimensional barcode recognition algorithm specifically designed for camera phones and validated its performance on public datasets, achieving an accuracy rate exceeding 90% and demonstrating high robustness against common image distortions. Katona [10] innovatively proposed a barcode detection algorithm combining bottom-hat filtering and morphological operations. They matched potential barcode areas by calculating the Euclidean distances between non-zero pixels, showing significant advantages compared to previous techniques.

In recent years, with the excellent accuracy and inference speed demonstrated by deep learning in multiple fields such as drones, road defects, and pedestrian detection [11–14], more and more researchers have begun to consider applying detection algorithms based on deep learning to barcode detection tasks. Hansen [15] adopted a combined strategy integrating the YOLO detection framework with a regression network based on Darknet19 to achieve efficient barcode detection and accurate prediction of orientation angles. Experimental results showed that this integrated method significantly outperformed many previous traditional techniques. Tian [16] designed a two-stage neural network architecture that cleverly integrates the Barcode Region Proposal Network (BRPN) with the barcode detection network, sharing deep convolutional feature maps generated by VGG16 as the core backbone network, optimizing computational resources, and achieving precise barcode detection and reliable orientation angle prediction. Jia [17] modified the Region Proposal Network (RPN) of Faster-RCNN by introducing oriented anchors, enhancing the network's robustness to distorted barcodes and making region predictions more accurate. Zhang [18] focused on optimizing the fully connected layers in Faster-RCNN by directly performing bounding box regression on the coordinates of quadrilateral vertices, further improving the accuracy and adaptability of barcode detection results. Xu [19] used the improved

YOLOv4 model and OCR to extract the barcode information and three-segment code of packages, respectively, and the package information recognition rate reached 98.5%, but the experimental sample size was too small and the data set was not open source.

Compared to detection methods based on traditional digital image processing, detection methods based on deep learning networks do not need to set threshold parameters that are sensitive to environmental factors, thereby significantly reducing the impact of environmental factors on the practical application of detection methods. However, current deep learning-based detection methods are constrained in their real-world application due to factors such as the high complexity of the algorithms and the need for improvement in detection speed.

With advancements in computing devices, the powerful parallel processing capabilities of GPUs can significantly accelerate model training and inference speeds, handling more data and more complex model structures. Therefore, research on deep learning-based barcode detection algorithms is of great significance. Based on this, this paper designs a high-speed barcode recognition algorithm combining deep learning-based object detection models. Considering that two-stage models generate candidate object bounding boxes in the first stage and classify and regress the bounding boxes in the second stage, they are slower in detection speed compared to one-stage models. Therefore, this paper adopts the one-stage model YOLOv8 as the baseline model. By improving its network structure, reducing model size, and enhancing model inference speed, it aims to meet barcode recognition tasks in complex and diverse real-world scenarios.

3. Method

3.1. The Framework of Barcode Recognition Algorithm

Due to the different encoding and reading methods of various barcodes, this paper utilizes Zbar for 1D barcodes and ZXing for 2D barcodes for decoding, respectively, and then integrates the obtained barcode information. The barcode recognition algorithm framework is illustrated in Figure 1.

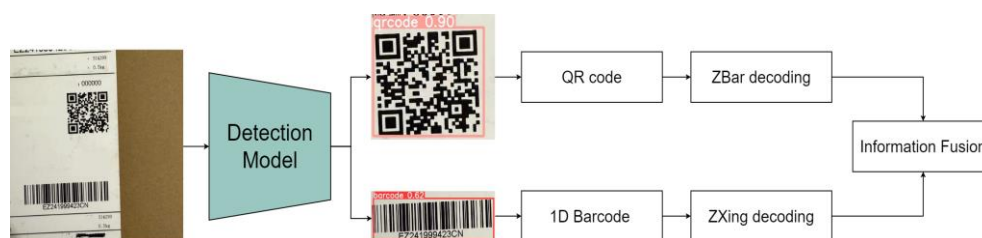


Figure 1. Comprehensive framework of a barcode recognition algorithm. The algorithm is mainly composed of the bar code detection model and the bar code decoding library.

The YOLOv8 network model can be functionally divided into three main parts: the backbone network, the neck network, and the detection head, with each part consisting of convolutional layers, C2f feature fusion modules, pooling layers, and other components. The detection accuracy and model size optimization of the original model leave room for improvement. Therefore, this paper modifies the model's network structure in three primary areas, as illustrated in Figure 2, which depicts the improved network structure. To address the issue of successful barcode recognition but failed decoding by the model, this paper modifies the original network structure in terms of barcode classification detection and barcode area delineation, aiming to enhance the model's barcode recognition accuracy and improve the decoding rate of subsequent decoding algorithms.

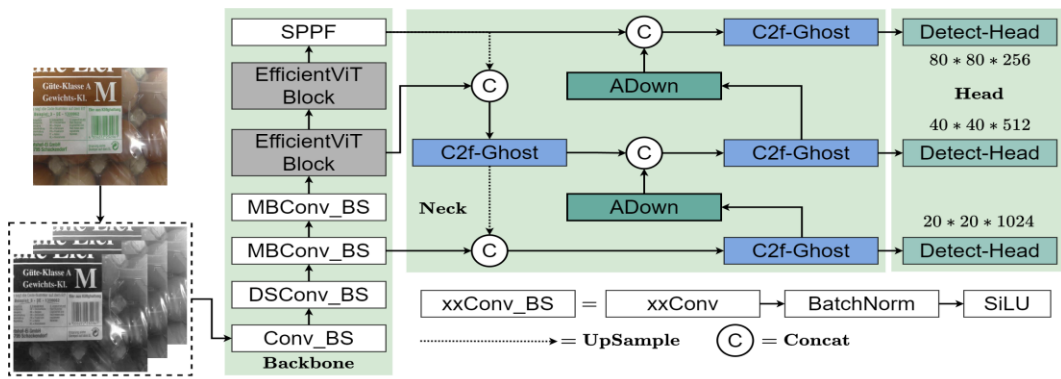


Figure 2. The network structure of the improved model.

In this paper, a linear attention mechanism and efficient downsampling modules are introduced into the backbone and neck of the original model, respectively, to achieve dimensionality reduction of the image while retaining the main features of the barcode as much as possible. To improve the operational efficiency of the model’s convolutional blocks and reduce the consumed computational resources, this paper optimizes the ordinary convolutions in the backbone and the convolutions in the C2f module. Furthermore, by reconstructing the model’s detection head and modifying the loss function, this paper further reduces the model’s parameter count while improving the quality of the model’s barcode prediction boxes.

3.2. EfficientViT Block and Multi-Scale Linear Attention

Deploying advanced object detection models in hardware devices often necessitates considering issues such as model detection accuracy and computational cost. A common practice is to introduce modules that contribute to improving accuracy into lightweight models, which, while enhancing model precision, may also increase the model’s size and computational requirements.

To improve the accuracy of barcode detection models and reduce their computational cost, this study introduces the EfficientViT [20] block (as shown in Figure 3) into the backbone of YOLOv8. The EfficientViT block employs the non-linear function ReLU in its self-attention mechanism and replaces the traditional softmax attention mechanism with a linear attention mechanism. Compared to the backbone of YOLOv8, the improved backbone incorporates a global attention mechanism, achieving higher barcode detection accuracy with fewer parameters and computational requirements.

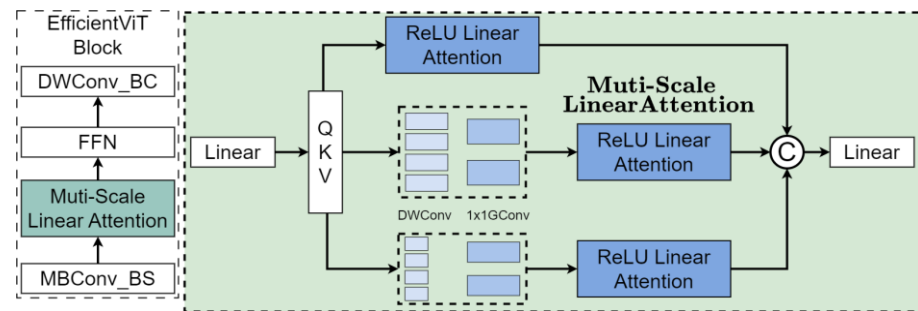


Figure 3. EfficientViT block (left) and multi-scale linear attention (right).

Specifically, given an input represented as $x \in R^{N \times f}$, the generalized form of the attention mechanism can be written as follows:

$$Attention(Q, K, V)_i = \sum_{j=1}^N \frac{Sim(Q_i, K_j)}{\sum_{j=1}^N Sim(Q_i, K_j)} V_j \tag{1}$$

where $Q = xW_Q, K = xW_K, V = xW_V$, and $W_Q/W_K/W_V \in R^{N \times d} (N \gg d)$ are learnable linear projection matrices. $Attention(Q, K, V)_i$ is row i of the matrix $Attention(Q, K, V)$ and $Sim(Q, K)$ is the similar equation.

In the original softmax attention mechanism [21], $Sim(Q, K) = exp(QK^T / \sqrt{d})$. However, in the EfficientViT block, a ReLU-based linear attention mechanism is adopted as the similarity function. Leveraging the associativity of matrix multiplication, Equation (1) can be progressively rewritten as shown in Equation (2). Compared to the attention mechanism based on the softmax function, the computational complexity of the ReLU-based linear attention mechanism is reduced from $O(N^2)$ to $O(N)$. Additionally, the ReLU function consumes fewer computational resources than the softmax function, making it more hardware-friendly for real-world intelligent logistics centers. Each Feed-Forward Network (FFN) layer in the EfficientViT block is followed by a depthwise convolution, which enhances the ReLU linear attention by leveraging convolution to increase the model's focus on spatial local information. Furthermore, group convolutions are utilized within the block to reduce the total number of operations for information aggregation, further improving the efficiency of barcode feature extraction by the model.

$$\begin{aligned} Sim(Q, K) &= ReLU(Q)ReLU(K)^T \\ Attention(Q, K, V)_i &= \frac{\sum_{j=1}^N [ReLU(Q_i)ReLU(K_j)^T]V_j}{\sum_{j=1}^N [ReLU(Q_i)ReLU(K_j)^T]} \\ &= \frac{ReLU(Q_i)\sum_{j=1}^N ReLU(K_j)^T V_j}{ReLU(Q_i)\sum_{j=1}^N ReLU(K_j)^T} \end{aligned} \quad (2)$$

Given the diversity and complexity of real-world scanning scenarios, input images often contain a mixture of different objects, varying lighting conditions, and barcodes occupying different areas. Traditional single-scale processing windows struggle with such multi-scale input data, potentially leading to the omission of critical information. To significantly enhance the model's efficiency in extracting barcode feature information in complex operational scenarios, we retain the Spatial Pyramid Pooling Fast (SPPF) module at the end of the network backbone. Thanks to the multi-scale spatial container embedded in the SPPF module, the model can more effectively process and integrate spatial feature information of different sizes compared to single-scale sliding window pooling methods. This avoids the imbalance of spatial feature information embedded in the image caused by cropping or distortion operations (as shown in Figure 4).



Figure 4. After many iterations of processing in the network, the input image may suffer from distortion and deformation. (a) cropped. (b) warped.

3.3. Efficient Multi-Scale Feature Fusion Network

In traditional convolutional neural networks, each convolutional layer performs complex convolutional operations on the input data to generate new feature maps. However, these feature maps often contain a large amount of redundant information, with many of them being similar or related. In the YOLOv8 network structure, the C2f module serves as the feature fusion module and is abundantly present in both the backbone and neck. To fur-

ther reduce the computational cost of the model and improve its operational efficiency, this study introduces the Ghost module [22] into the C2f module, improving the convolutional operation efficiency of the C2f module from the convolutional level and thus enhancing the detection speed of the barcode detection model.

In the Ghost module (as shown in Figure 5), one of a set of similar feature maps is designated as the Intrinsic feature map, while the others are termed Ghost feature maps. These Ghost feature maps can be obtained by applying linear transformations to the Intrinsic feature map. Given an input $F_{in} \in R^{h \times w \times c}$, the convolutional filters $f_C \in R^{k \times k \times c \times n}$ of the layer are evenly divided into s groups. One group of convolutional filters ($f_i \in R^{k \times k \times c \times m}, m = n/s$) is selected to perform standard convolutional operations, resulting in m Intrinsic feature maps. Each time, m Intrinsic feature maps are grouped together and subjected to a linear transformation. To ensure uniformity in calculations and reduce resource consumption, the same linear transformation is used to compute m Ghost feature maps for that group. Finally, m Intrinsic feature maps and $(n - m)$ Ghost feature maps are concatenated to produce the final output $F_{out} \in R^{h_1 \times w_1 \times n}$. Assuming that all convolutional kernels have a size of $k \times k$, the theoretical speedup ratio r_S and compression ratio r_C achieved by using the Ghost module are calculated as follows:

$$\begin{aligned}
 r_S &= \frac{n \times h_1 \times w_1 \times c \times k^2}{\frac{n}{s} \times h_1 \times w_1 \times c \times k^2 + (s-1) \times \frac{n}{s} \times h_1 \times w_1 \times k^2} \\
 &\approx \frac{s \times c}{s+c-1} \approx s, \\
 r_C &= \frac{n \times c \times k^2}{\frac{n}{s} \times c \times k^2 + (s-1) \times \frac{n}{s} \times k^2} \approx \frac{s \times c}{s+c-1} \approx s
 \end{aligned}
 \tag{3}$$

where $s \ll c$.

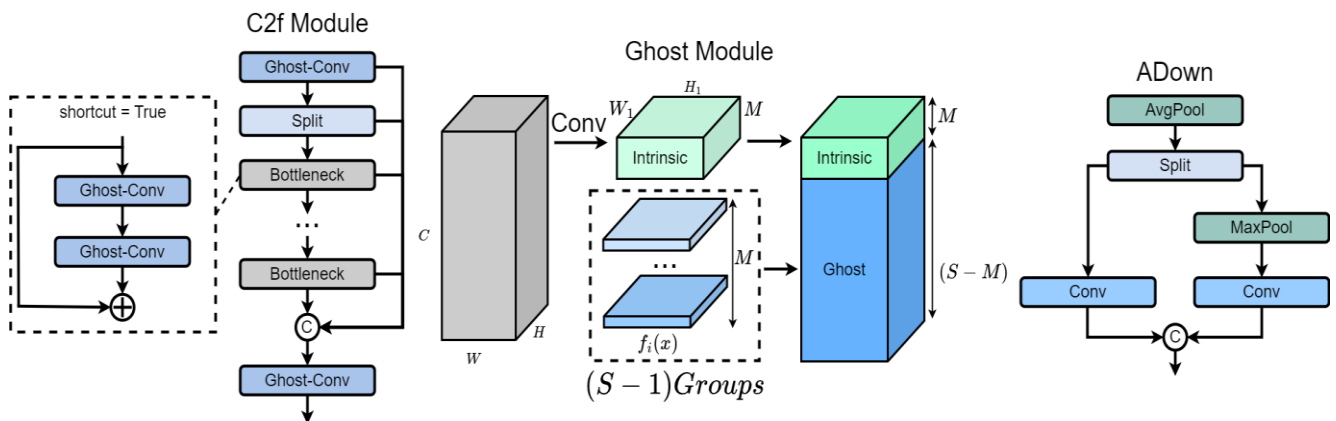


Figure 5. C2f-Ghost module (left), ghost module (middle), and ADown module (right).

The reduction in computation, according to calculation Formula (3), hinges on the grouping of convolutional filters within convolutional layers. In this paper, the Ghost module is employed in each convolutional layer of the C2f module, achieving similar feature extraction effects as traditional convolutional layers with fewer computational resources.

Furthermore, considering the concatenation operations of features at different scales within the model’s neck structure, this paper selects the ADown [23] module as the convolutional block for the model’s downsampling operations. Compared to conventional convolutional blocks, the ADown module boasts reduced parameters and computational complexity, significantly contributing to improving the model’s operational efficiency and hardware compatibility.

3.4. Lightweight Detection Head

In the field of object detection, there are mainly two types of detection heads: anchor-free and anchor-based. The anchor-based detection head can generate dense anchor boxes, enabling the network to directly classify objects and perform bounding box regression,

thereby enhancing the model’s recall rate. However, this method also leads to the generation of numerous redundant anchor boxes and requires careful setting of several hyperparameters during model training, which increases the model’s computational load and memory consumption. Based on this, this paper opts for an anchor-free detection head. To further reduce the number of parameters and computational complexity of the decoupled head structure, modifications are made to the detection head: two grouped convolutions are utilized at the head position to achieve parameter sharing, and then the target location and category information are separately extracted to form the decoupled head. The improved detection head boasts fewer parameters, lower computational complexity, and higher parameter utilization efficiency. The specific structure is illustrated in Figure 6.

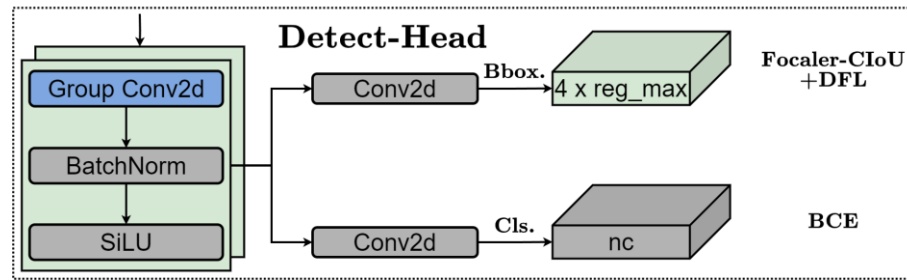


Figure 6. The integral architecture of the detection head. After Group Convolution, the input images enter two branches respectively.

To mitigate the adverse effects of sample imbalance during the model training phase and improve training quality by treating easy and hard samples differently, this paper selects Focaler-CIoU [24] as the loss function for the model. Given $[u, d] \in [0, 1]$, the value of $IoU^{Focaler}$ can be adjusted through modifying u and d to focus on different samples. Since the true distribution of target barcodes typically does not deviate significantly from their annotated positions, the model should prioritize values near the annotated positions during training. Therefore, Distribution Focal Loss (DFL) [25] is chosen to be paired with the Focaler-CIoU loss function, which together act on the bounding box regression of the barcode detection model presented in this paper. The calculations for the Focaler-CIoU and DFL loss functions are provided in Equations (4), (5), and (6), respectively.

$$L_{Focaler-CIoU} = L_{CIoU} + IoU - IoU^{Focaler} = 1 + \frac{\rho^2(b, b^{st})}{c^2} + \alpha v - IoU^{Focaler}, \tag{4}$$

$$IoU = \frac{A \cap B}{A \cup B}, \alpha = \frac{v}{(1 - IoU) + v}, v = \frac{4}{\pi^2} \left(\arctan \frac{w^{st}}{h^{st}} - \arctan \frac{w}{h} \right)^2, \tag{5}$$

$$IoU^{Focaler} = \begin{cases} 0, & IoU < d \\ \frac{IoU - d}{u - d}, & d \ll IoU \ll u \\ 1, & IoU > u \end{cases} \tag{5}$$

where, A and B are the area occupied by the prediction box and the real box, respectively, b and b^{st} are the coordinates of the center point of the prediction box and the real box, respectively, $\rho(\bullet)$ is the Euclidean distance, c is the diagonal distance of the smallest external rectangle, w and w^{st} are the width of the prediction box and the real box, respectively, and h and h^{st} are the height of the prediction box and the real box, respectively.

$$DFL(S_i, S_{i+1}) = -(y_{i+1} - y) \log(S_i) - (y - y_i) \log(S_{i+1}), \tag{6}$$

$$S_i = \frac{y_{i+1} - y}{y_{i+1} - y_i}, S_{i+1} = \frac{y - y_i}{y_{i+1} - y_i}$$

where S_i is the probability of predicting label y_i , S_{i+1} is the probability of predicting label y_{i+1} , and y_i and y_{i+1} are the two labels closest to label y , $y_i \leq y \leq y_{i+1}$.

4. Experiments

4.1. Preparation of the Dataset and Configuration of the Experimental Environment

Due to the limited availability of public datasets related to barcodes, all barcode images used in the experiments were sourced from the internet, and we annotated the images in the experiment. To address the scarcity of original images, data augmentation techniques such as cropping, rotation, stitching, noise addition, and contrast adjustment were applied to expand the dataset. Ultimately, 2741 images were selected as the total sample for the barcode detection experiments conducted in this paper. The barcode dataset, as illustrated in Figure 7, features images with a resolution of 416×416 . It encompasses both one-dimensional and two-dimensional barcodes, covering real-world scenarios such as retail merchandising, logistics management, and invoice and receipt processing.



Figure 7. Barcode dataset. It includes real-world scenarios such as commodity retail, logistics management, and invoice receipt handling.

To validate the effectiveness of the barcode detection algorithm proposed in this paper, as well as the outcomes of model training and validation, the open-source labeling tool labellingm was used to objectively convert the images into the YOLO dataset format. The images were then randomly divided into training, validation, and test sets in a 6:1:1 ratio, with the training set containing 2039 images and both the validation and test sets containing 351 images each.

In order to ensure the fairness of the experiments, all experiments are conducted on the same server for training and reasoning processes. Table 1 shows the configuration of the experimental environment.

Table 1. Experimental environment.

Parameter	Configuration
Programming language	Python
Deep learning framework	Pytorch1.8.1
CPU	Intel (R) Xeon Gold 6248R
GPU	NVIDIA RTX A6000
CUDA	11.8
Batch size	16
Initial learning rate	0.01
Epoch	200
SGD momentum	0.937
Weight decay	0.0005

4.2. Evaluation Metrics for Model Performance

Since the object detection model studied in this paper will be applied in real-time barcode detection and localization in practical scenarios, the evaluation criteria of significance include detection accuracy, inference speed, model size, and computational complexity.

The experiments in this paper primarily select recall rate (R) and mAP50:95 as evaluation metrics for model accuracy, with their calculations provided in Formula (6).

$$R = \frac{TP}{TP + FN}, mAP = \frac{\sum_i^C AP_i}{C} \quad (7)$$

In ablation experiments, we utilize metrics such as localization error, background error, missed GT (ground truth) error, and False Positive error, as defined in TIDE [26], to intuitively reflect the discrepancy between the predicted bounding boxes of the barcode localization model and the ground truth boxes. Under the assumption that a predicted box and a ground truth box share the same category and have an IOU (Intersection over Union) overlap greater than a set threshold t , we select the one with the highest overlap as a True Positive, and the rest are considered False Positives. Consequently, the four aforementioned errors can be defined as follows:

Localization error (E_{Loc}): The prediction box is correctly classified but mislocated.

Background error (E_{Bkg}): Background is falsely detected as foreground.

Missed GT error (E_{Miss}): All ground truth boxes that are not detected, excluding classification errors and localization errors.

False Positive error (E_{FP}): It can arise from mislocalization, background confusion, duplicate detections, and other reasons.

4.3. Ablation Experiment

The results of ablation experiments are shown in Table 2. The improvement methods proposed in this paper all contribute to reducing model size and improving detection accuracy. Specifically, when the model's backbone network employs the EfficientViT block, the model achieves good lightweight results, with a recall rate of 93.3%, mAP50:95 of 73.8%, 6.8 M parameters, and 17.4 G FLOPs. When the model uses the improved C2f module, it has 7.8 M parameters, 5.2 G FLOPs, a recall rate of 93%, and mAP50:95 of 73.5%. When the model adopts the ADown convolutional block in downsampling, it achieves the most significant improvement in detection accuracy, with a recall rate of 93.4% and mAP50:95 of 73.8%, which are 0.7% and 0.4% higher than the original model, respectively. When the model applies all the improvement strategies proposed in this paper, it demonstrates the best overall performance, with 2.86 M parameters, 5.8 G FLOPs, a recall rate of 94.5%, and mAP50:95 of 75.3%. The experimental results verify the effectiveness of the improvement methods proposed in this paper. Compared with the pre-improvement model, the improved model achieves better lightweight performance while enhancing detection accuracy, which is of great practical significance for deploying the model on resource-constrained platforms such as mobile devices and embedded systems.

Table 2. Ablation experiment results on barcode dataset.

Methods				Params (M)	FLOPs (G)	Recall (%)	mAP50:95 (%)
Backbone-im *	C2f-Ghost	ADown	Head-im *				
				11.1	28.4	92.7	73.4
✓				6.8	17.4	93.3	73.8
	✓			7.8	19.0	93	73.5
		✓		9.33	21.3	93.4	73.8
			✓	10.06	27.8	92.9	73.6
✓	✓			5.2	13.6	94.2	73.9
✓	✓	✓		4.66	12.9	94.3	74.6
✓	✓	✓	✓	2.86	5.8	94.5	75.3
Improvement (%)				(−74.2%)	(−79.6%)	(+1.8%)	(+1.9%)

* Backbone-im and Head-im represent the improved backbone and head, respectively.

To intuitively understand the feature extraction and fusion processes of the model when processing input images, this paper analyzes the heatmaps outputted from various layers before and after model improvements, with the results illustrated in the figures. As shown in Figure 8, after partially improving the backbone and neck of the model, the heatmaps display clearer feature boundaries, more accurate feature localization, and stronger feature representation capabilities. This indicates that the improvement strategies proposed in this paper for the model’s network structure enhance its feature extraction and fusion abilities.

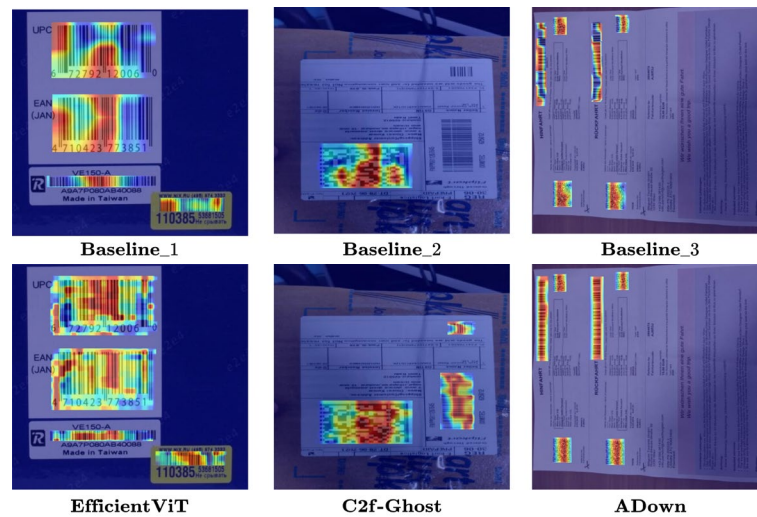


Figure 8. Heatmap outputs before and after model improvement. The top three are heatmaps generated by the baseline model, while the bottom three are heatmaps produced by the improved model.

This paper also records the changes in detection accuracy curves during the training phase of the model before and after network improvements, as shown in Figure 9. Compared to the baseline model, the improved model exhibits higher accuracy in the medium-to-high confidence ranges, indicating a higher recall rate in scenarios close to real-world applications. Additionally, during the training phase, the improved model achieves higher mAP50:95 values, demonstrating better accuracy performance when the model training converges under the same conditions. The experiments show that the improvement strategies proposed in this paper contribute to enhancing the model’s detection accuracy.

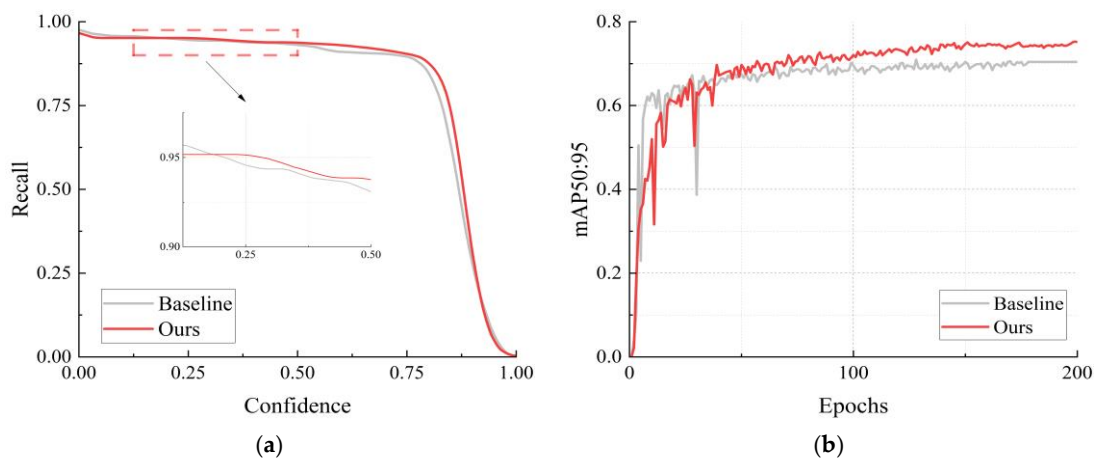


Figure 9. Comparison chart of recall (a) and mAP50:95 (b) Curves between the baseline model and the improved model during the training phase.

Appropriate loss functions often improve a model’s detection performance. To verify the effectiveness and rationality of the loss function selected in this paper for enhancing

model accuracy in real-world barcode detection tasks, various loss functions such as CIoU [27], EIoU [28], SIoU [29], Focaler-EIoU, and Focaler-SIoU were tested, and relevant experimental information was recorded. The experimental results are shown in Table 3. The table indicates that, compared to EIoU and SIoU, when the model uses CIoU as the loss function for bounding box regression, it achieves higher recall and mAP50:95 values of 93.5% and 74.6%, respectively. Furthermore, when the model employs Focaler-IoU, it selectively weights difficult and easy samples during the training phase, resulting in improved model accuracy. Specifically, Focaler-EIoU improves recall and mAP50:95 by 0.5% and 0.5% compared to EIoU, Focaler-SIoU enhances recall and mAP50:95 by 2.5% and 0.4% compared to SIoU, and Focaler-CIoU boosts recall and mAP50:95 by 1.0% and 0.7% compared to CIoU. The experiments demonstrate that the differential treatment of difficult and easy samples in Focaler-IoU helps improve the model's detection accuracy.

Table 3. Detection accuracy of the improved model under different loss functions.

Methods		Recall (%)	mAP50:95 (%)
IoU	CIoU	93.5	74.6
	EIoU	92.2	73.4
	SIoU	91.7	73.2
Focaler-IoU	EIoU	92.7 (+0.5)	73.9 (+0.5)
	SIoU	94.2 (+2.5)	73.6 (+0.4)
	CIoU(Ours)	94.5 (+1.0)	75.3 (+0.7)

Considering that different loss functions may impact the quality of the model's bounding boxes, to further explore the influence of various IoU metrics on the model's detection accuracy, the TIDE metric is also employed to evaluate the errors between the model's predicted bounding boxes and the ground truth bounding boxes under different loss functions. Experimental results (as shown in Table 4) indicate that when the model's detection head uses Focaler-IoU, the model's errors decrease. Specifically, for EIoU, the Localization error decreases by 0.64, the Background error by 0.14, the Missed GT error by 0.29, and the False Positive error by 0.03; for SIoU, the Localization error drops by 0.11, the Background error by 0.18, the Missed GT error by 0.39, and the False Positive error by 0.14; for CIoU, the Localization error decreases by 0.26, the Background error by 0.36, the Missed GT error by 0.15, and the False Positive error by 0.32. The results suggest that using Focaler-IoU helps improve the model's localization and classification of barcodes, reducing errors. Considering both the detection accuracy of barcodes and the quality of bounding boxes among different IoUs, the model using Focaler-CIoU exhibits the best overall detection performance for barcodes, further validating the effectiveness of the improved loss function strategy proposed in this paper.

Table 4. Comparison of prediction box vs. ground truth box errors for the improved model under different loss functions (using TIDE metrics).

Methods		$E_{Loc} \downarrow$	$E_{Bkg} \downarrow$	$E_{Miss} \downarrow$	$E_{FP} \downarrow$
IoU	CIoU	0.79	0.89	1.43	1.21
	EIoU	0.41	1.10	1.53	1.75
	SIoU	0.51	1.29	1.87	1.79
Focaler-IoU	EIoU	0.15 (−0.64)	0.75 (−0.14)	1.04 (−0.29)	1.18 (−0.03)
	SIoU	0.40 (−0.11)	1.11 (−0.18)	1.48 (−0.39)	1.65 (−0.14)
	CIoU(Ours)	0.53 (−0.26)	0.53 (−0.36)	1.28 (−0.15)	0.89 (−0.32)

4.4. Model Comparison Experiment

To verify the balance between detection performance and model lightweighting achieved by the barcode detection model designed in this paper for barcode detection tasks in real-world scenarios, common models in the current object detection field are selected for comparison experiments. A total of four models, including those from the

YOLO [30,31] series and RT-DETR [32], are compared with the proposed model. The experimental results are shown in Table 5. According to Table 5, RT-DETR has the largest number of parameters and FLOPs, which are 41.9 M and 125.6 G, respectively, but its detection accuracy and speed are relatively poor. Within the YOLO series, the YOLOv7-tiny model has the smallest number of parameters and FLOPs, which are 6.01 M and 13.0 G, respectively, with a recall rate of 91.8% and an mAP50:95 of 66.7%. YOLOv10s has the fastest inference speed, with an FPS value of 244. Compared with other models, the improved model proposed in this paper has the smallest number of parameters and lowest computational cost while achieving the best detection accuracy and speed.

Table 5. Experimental results of comparative performance analysis for barcode detection using different models.

Methods	Params (M)	FLOPs (G)	Recall (%)	mAP50:95 (%)	FPS
Baseline	11.1	28.4	92.7	73.4	210
YOLOv5s	7.02	15.8	91.6	67.8	196
YOLOv7-tiny	6.01	13.0	91.8	66.7	208
RT-DETR-R50	41.9	125.6	90.0	72.8	85
YOLOv9s	9.6	38.7	92.4	71.3	122
YOLOv10s	7.22	21.4	92.1	70.8	244
Ours	2.86	5.8	94.5	75.3	230

To verify the detection accuracy of the model for barcodes in real-world scenarios, some barcode images are selected as samples to evaluate the detection results of different models. Figure 10 shows the comparison results of some models for barcode detection. As can be seen from the figure, Baseline, YOLOv5s, RT-DETR, YOLOv9s, and YOLOv10s exhibit varying degrees of missed detections, false detections, and repeated detections. YOLOv7-tiny has poor boundary determination for barcode bounding boxes. Compared with other models, the improved model, due to the incorporation of a linear attention mechanism, is able to extract more feature maps related to barcodes, leading to more accurate barcode recognition and reducing the probability of repeated detections.

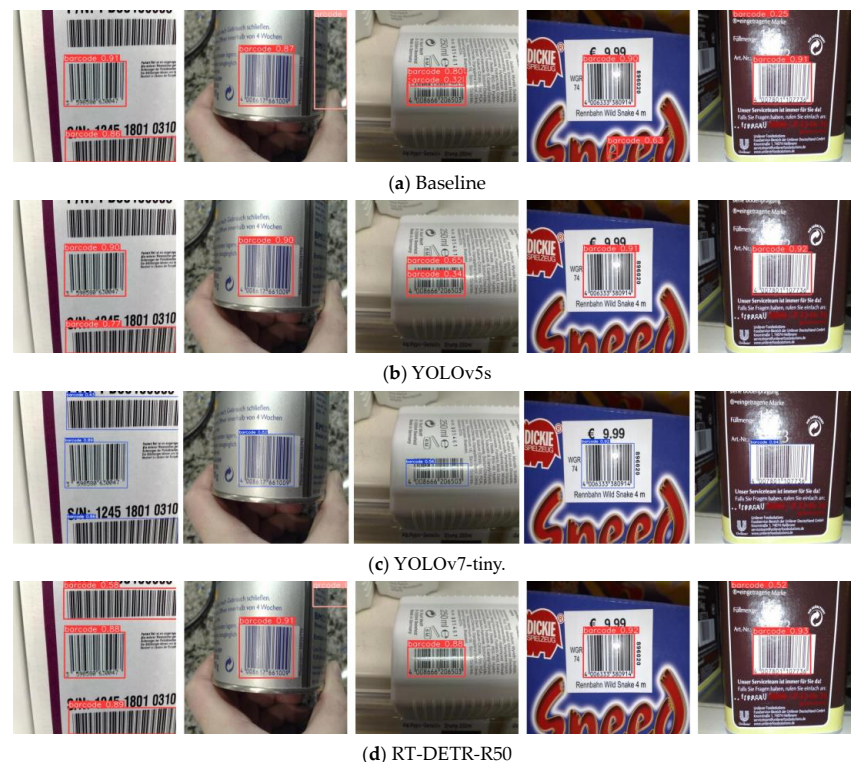


Figure 10. Cont.

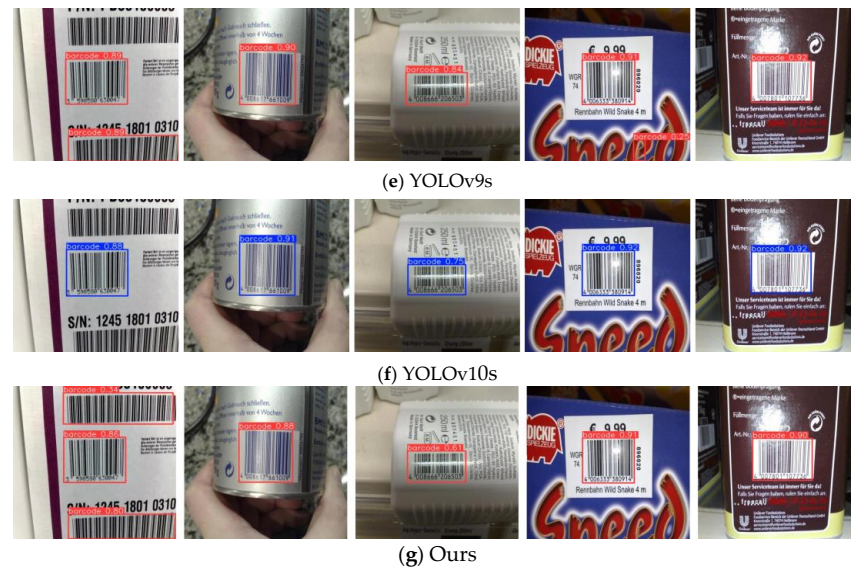


Figure 10. Comparison of detection results for barcodes in real-world scenarios using different models. The bar code detection model designed in this paper has the best performance.

Furthermore, this paper randomly crops and stitches images in the test set, compiling a total of 300 barcode images as the total sample for a barcode recognition comparison experiment. The experimental results are shown in Table 6. According to Table 6, the proposed model achieves the highest barcode reading success rate of 95.5% among the images and has the lowest number of missed detections. Based on all experimental results, the proposed model demonstrates the best overall performance in barcode detection tasks and outperforms other models in real-world scenarios.

Table 6. Comparison of detection and decoding results for barcodes using different detection models.

Methods	Decoding Success Rate (%)	Number of Missed Tests
Baseline	93.7	48
YOLOv5s	92.3	58
YOLOv7-tiny	93.4	50
YOLOv9s	93.2	52
YOLOv10s	92.8	55
RT-DETR	91.4	65
Ours	95.5	34

5. Conclusions

For barcode detection tasks in real-world scenarios, this paper proposes a barcode detection algorithm based on an improved YOLOv8. In the model's backbone, the EfficientViT Block is utilized to enhance the accuracy of feature extraction. In the model's neck, improved C2f modules and ADown modules are employed, optimizing the efficiency of feature fusion at the convolutional level. Additionally, the model's detection head is reconstructed, significantly improving training quality and detection performance with lower parameter and computational costs. Experimental results indicate that, compared to the original algorithm, the improved model achieves a 1.8-percentage-point increase in recall rate, a 1.9-percentage-point improvement in mAP50:95, a 20 frames-per-second (FPS) increase, a 74.2% reduction in model parameters, and a 79.6% decrease in FLOPs. The algorithm strikes a good balance between recognition efficiency, recognition accuracy, algorithm versatility, and deployment costs.

Currently, there is still room for optimization of this model. Researchers will conduct more experiments to further enhance the model's feature extraction capabilities, improve

the efficiency of feature fusion, and refine the model's network structure, aiming to design a universal barcode detection model suitable for real-world scenarios.

Author Contributions: Conceptualization, J.C. and N.D.; methodology, X.H.; software, J.C.; validation, J.C. and N.D.; formal analysis, X.H. and J.C.; investigation, J.C. and N.D.; resources, X.H. and Y.Y.; data curation, X.H. and J.C.; writing—original draft preparation, Y.Y.; writing—review and editing, X.H., N.D. and J.C.; visualization, J.C.; supervision, X.H. and J.C.; project administration, X.H. and J.C.; funding acquisition, X.H. and N.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by The Science and Technology Program of Zhejiang Province, China (No. 2022C01202, No. 2022C01065), Zhejiang Provincial Department of Education Research Project (Y202455953) and the Zhejiang Sci-Tech University Research Start-up Fund, China (No. 23242083-Y).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data will be made available on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Yan, L.Y.; Tan, G.W.H.; Loh, X.M.; Hew, J.J.; Ooi, K.B. QR code and mobile payment: The disruptive forces in retail. *J. Retail. Consum. Serv.* **2021**, *58*, 102300. [[CrossRef](#)]
2. De Luna, I.R.; Liébana-Cabanillas, F.; Sánchez-Fernández, J.; Muñoz-Leiva, F. Mobile payment is not all the same: The adoption of mobile payment systems depending on the technology applied. *Technol. Forecast. Soc. Change* **2019**, *146*, 931–944. [[CrossRef](#)]
3. Elaskari, S.; Imran, M.; Elaskri, A.; Almasoudi, A. Using barcode to track student attendance and assets in higher education institutions. *Procedia Comput. Sci.* **2021**, *184*, 226–233. [[CrossRef](#)]
4. Tan, L.; Lu, Y.; Yan, X.; Liu, L.; Zhou, X. XOR-ed visual secret sharing scheme with robust and meaningful shadows based on QR codes. *Multimed. Tools Appl.* **2020**, *79*, 5719–5741. [[CrossRef](#)]
5. Nuhi, A.; Memeti, A.; Imeri, F.; Cico, B. Smart attendance system using qr code. In Proceedings of the 2020 9th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 8–11 June 2020; pp. 1–4.
6. Küng, K.; Aeschbacher, K.; Rüttsche, A.; Goette, J.; Zürcher, S.; Schmidli, J.; Schwendimann, R. Effect of barcode technology on medication preparation safety: A quasi-experimental study. *Int. J. Qual. Health Care* **2021**, *33*, mzab043. [[CrossRef](#)] [[PubMed](#)]
7. Ang, J.L.F.; Lee, W.K.; Ooi, B.Y.; Ooi, T.W.M. Location Sensing using QR codes via 2D camera for Automated Guided Vehicles. In Proceedings of the 2020 IEEE Sensors Applications Symposium (SAS), Kuala Lumpur, Malaysia, 9–11 March 2020; pp. 1–6.
8. Ohbuchi, E.; Hanaizumi, H.; Hock, L.A. Barcode readers using the camera device in mobile phones. In Proceedings of the 2004 International Conference on Cyberworlds, Tokyo, Japan, 18–20 November 2004; pp. 260–265.
9. Wachenfeld, S.; Terlunen, S.; Jiang, X. Robust recognition of 1-d barcodes using camera phones. In Proceedings of the 2008 19th International Conference on Pattern Recognition, Tampa, FL, USA, 8–11 December 2008; pp. 1–4.
10. Katona, M.; Nyúl, L.G. Efficient 1D and 2D barcode detection using mathematical morphology. In Proceedings of the Mathematical Morphology and Its Applications to Signal and Image Processing: 11th International Symposium, ISMM 2013, Uppsala, Sweden, 27–29 May 2013; pp. 464–475.
11. Aceto, G.; Ciunzo, D.; Montieri, A.; Pescapé, A. Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges. *IEEE Trans. Netw. Serv. Manag.* **2019**, *16*, 445–458. [[CrossRef](#)]
12. Li, J.; Sun, A.; Han, J.; Li, C. A survey on deep learning for named entity recognition. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 50–70. [[CrossRef](#)]
13. Chen, J.; Wen, Y.; Nanekaran, Y.A.; Zhang, D.; Zeb, A. Multiscale attention networks for pavement defect detection. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 2522012. [[CrossRef](#)]
14. Liu, M.; Jiang, J.; Zhu, C.; Yin, X.C. Vlpd: Context-aware pedestrian detection via vision-language semantic self-supervision. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 6662–6671.
15. Hansen, D.K.; Nasrollahi, K.; Rasmussen, C.B. Real-time barcode detection and classification using deep learning. In *International Joint Conference on Computational Intelligence*; SCITEPRESS Digital Library: Lisbon, Portugal, 2017; pp. 321–327.
16. Tian, Y.; Che, Z.; Zhai, G.; Gao, Z. BAN, a barcode accurate detection network. In Proceedings of the 2018 IEEE Visual Communications and Image Processing (VCIP), Taichung, Taiwan, 9–12 December 2018; pp. 1–5.
17. Jia, J.; Zhai, G.; Zhang, J. EMBDN: An efficient multiclass barcode detection network for complicated environments. *IEEE Internet Things J.* **2019**, *6*, 9919–9933. [[CrossRef](#)]

18. Zhang, J.; Min, X.; Jia, J. Fine localization and distortion resistant detection of multi-class barcode in complex environments. *Multimed. Tools Appl.* **2020**, *80*, 16153–16172. [[CrossRef](#)]
19. Xu, X.; Xue, Z.; Zhao, Y. Research on an algorithm of express parcel sorting based on deeper learning and multi-information recognition. *Sensors* **2022**, *22*, 6705. [[CrossRef](#)] [[PubMed](#)]
20. Cai, H.; Li, J.; Hu, M. Efficientvit: Lightweight multi-scale attention for high-resolution dense prediction. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 4–6 October 2023; pp. 17256–17267.
21. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
22. Han, K.; Wang, Y.; Tian, Q. Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF conference on computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 1577–1586.
23. Wang, C.Y.; Yeh, I.H.; Liao, H.Y.M. Yolov9: Learning what you want to learn using programmable gradient information. *arXiv* **2024**, arXiv:2402.13616.
24. Zhang, H.; Zhang, S. Focaler-IoU: More Focused Intersection over Union Loss. *arXiv* **2024**, arXiv:2401.10525.
25. Li, X.; Wang, W.; Wu, L. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 21002–21012.
26. Bolya, D.; Foley, S.; Hays, J. Tide: A general toolbox for identifying object detection errors. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; pp. 558–573.
27. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence 2020, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12993–13000.
28. Zhang, Y.F.; Ren, W.; Zhang, Z. Focal and efficient IOU loss for accurate bounding box regression. *arXiv* **2021**, arXiv:2101.08158. [[CrossRef](#)]
29. Gevorgyan, Z. SIoU loss: More powerful learning for bounding box regression. *arXiv* **2022**, arXiv:2205.12740.
30. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 7464–7475.
31. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. Yolov10: Real-time end-to-end object detection. *arXiv* **2024**, arXiv:2405.14458.
32. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. DETRs beat YOLOs on real-time object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024; pp. 16965–16974.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.