electronics

Article

Deep Learning Neural Network-Based Detection of Wafer Marking Character Recognition in Complex Backgrounds

Yufan Zhao, Jun Xie * and Peiyu He

School of Mechanical Engineering, Jiangsu University, Zhenjiang 212013, China
* Correspondence: xiejun@ujs.edu.cn

Abstract: Wafer characters are used to record the transfer of important information in industrial production and inspection. Wafer character recognition is usually used in the traditional template matching method. However, the accuracy and robustness of the template matching method for detecting complex images are low, which affects production efficiency. An improved model based on YOLO v7-Tiny is proposed for wafer character recognition in complex backgrounds to enhance detection accuracy. In order to improve the robustness of the detection system, the images required for model training and testing are augmented by brightness, rotation, blurring, and cropping. Several improvements were adopted in the improved YOLO model, including an optimized spatial channel attention model (CBAM-L) for better feature extraction capability, improved neck structure based on BiFPN to enhance the feature fusion capability, and the addition of angle parameter to adapt to tilted character detection. The experimental results showed that the model had a value of 99.44% for *mAP*@0.5 and an *F*1 score of 0.97. In addition, the proposed model with very few parameters was suitable for embedded industrial devices with small memory, which was crucial for reducing the hardware cost. The results showed that the comprehensive performance of the improved model was better than several existing state-of-the-art detection models.

Keywords: YOLO v7-Tiny; wafer character recognition; attention mechanism; BiFPN; rotation detection

Citation: Zhao, Y.; Xie, J.; He, P. Deep Learning Neural Network-Based Detection of Wafer Marking Character Recognition in Complex Backgrounds. *Electronics* **2023**, *12*, 4293. https://doi.org/10.3390/electronics12204293

Academic Editors: Junhua Ding, Haihua Chen, Yunhe Feng and Tozammel Hossain

Received: 31 August 2023
Revised: 15 September 2023
Accepted: 18 September 2023
Published: 17 October 2023

1. Introduction

Wafer characters are codes comprising numbers, letters, and symbols, and contain production information for each wafer. If error of a wafer character recognition occurs during production, the information cannot be matched, significantly reducing production efficiency. Therefore, improvement of the accuracy of the wafer character recognition method is significant for improving the production efficiency of the semiconductor industry.

Wafer character recognition belongs to the category of optical character recognition [1]. Before the advent of deep learning-based techniques [2], optical character recognition was based on the template matching method [3]. The matching results were obtained by this method through extracting characters from the image matrix and determining the similarity between the characters and the template. Tian et al. [4] proposed the segmentation of Chinese license plate characters by using multiscale template matching. Chen et al. [5] used image segmentation, normalization, and template matching techniques for license plate recognition. Ryan et al. [6] proposed character recognition on ID cards using template matching. Zhang et al. [7] proposed a license plate character segmentation method based on character contour and temple matching. Jung et al. [8] used template matching for 7-segment optical character recognition. However, before template matching, it is usually necessary to extract the target region to be recognized. With less interference contained in the extracted target region, the larger the area of the true target region, the better the results obtained after matching with the template. When the background in the image is more complex, then there are more intersections between the target region and the interference.

It would be difficult to extract the target accurately due to these interferences. Matching results are greatly affected by the large differences between the extracted regions and the template. Similarly, when template matching is performed using gradient and other methods, the matching results will be greatly affected by the overlapping background interference. After the emergence of deep learning-based technology, this technology had been widely used in optical character recognition due to its advantages of higher detection accuracy, great robustness, and time efficiency. Weng et al. [9] proposed a new deep learning-based handwritten character recognition system on mobile computing devices. Kim et al. [10] proposed a multi-task convolutional neural network system for license plate recognition. Yang et al. [11] used a combination of CNN and ELM for learning deep features related to Chinese characters. Rakhshani et al. [12] proposed a deep learning network for license plate recognition. Cao et al. [13] utilized registered 3D models as well as classical convolutional networks for wafer character recognition based on a priori information.

Wafer character recognition was often previously performed using the traditional method with a small model size and simple structure in practical industrial applications [14–16]. However, the accuracy and robustness of recognition were weak. Therefore, combining the excellent detection performance of the deep learning network and the characteristics of wafer characters, this paper proposes an improved YOLO v7-Tiny network for wafer character recognition in complex backgrounds. The main objective was to develop a robust, accurate, computationally inexpensive detection system that was applicable to industrial environments. The main contributions of this work are as follows: First, the wafer dataset was established, which was acquired from an industrial camera and image collection. Then, the network structure of YOLO v7-Tiny was improved in terms of the attention mechanism, the feature fusion network structure, and the rotation detection frame. The improved YOLO model achieved better detection performance while ensuring the model was lightweight.

2. Description of the Theories

2.1. Feature Extraction Module Based on Attention Mechanism

Processing efficiency and accuracy were improved under limited computational conditions by the attention module via focusing on the target information that was more critical to the current task among the many inputs, effectively obtaining more details related to the target and filtering out irrelevant information [17]. Combining the deep learning network with the attention mechanism module, the loss of important information in an image could be reduced in passing information between network layers [18]. The attention module has been widely used in practical detection [19].

2.2. Feature Fusion Network

The features in lower layers had a higher resolution and contained more location-based and detailed information during the feature extraction process. Higher-level features had stronger semantics but lower resolution and less location information and detail. A top-down architecture with horizontal connectivity was structured by FPN (Feature Pyramid Network for Object Detection) for constructing high-level semantic feature maps at different scales [20]. The high-level features could be transferred by this structure to complement the semantics of the lower levels while fusing high-level features with low-level features. In this way, high-resolution and strong semantic features could be obtained, facilitating multi-scale target detection. With the development of technology and research, the new enhancement paths were created by PANet and BiFPN structures to strengthen the tightness of information flow transfer in the network and enhance the network's ability to fuse feature information with good results [21,22].

2.3. Detection of Rotating Targets

A rectangular detection frame with a tilt angle of zero degrees has been previously commonly used to contain the detection object in many detecting works. However, many detection objects are unsuitable for such rectangular frames in practical application situ-

ations. This detection frame is not adaptable to detection targets with tilt angles and the inclusion of the object is not accurate enough. Meanwhile, problems such as decreasing the area of the overlapping part of the detection frame and the real object will occur. Therefore, a detection model with angular parameters has been used in many practical applications, such as ship position detection [23,24], unconstrained license plate position detection [25], insulator orientation detection [26], and remote sensing image detection [27].

3. Materials and Methods

The flow of the wafer character detection method proposed in this paper is shown in Figure 1. In the first step, a batch of wafers was placed on a mobile device, and the images of characters on the wafers were captured by an industrial camera during the moving process. Afterward, data enhancement was performed on the images. The processed images were collected to construct a wafer character dataset, and individual characters in the image were labeled. Then, the YOLO network was trained on the constructed dataset. Furthermore, the evaluation index was calculated to evaluate the detection results. Finally, the improved model was used in industrial production for actual detection.

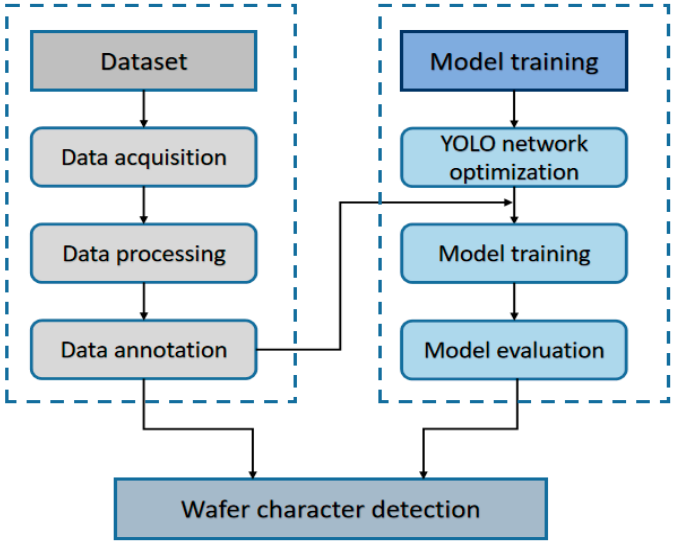

Figure 1. Overall architecture of the proposed character detection.

3.1. Dataset

3.1.1. Data Acquisition

The dataset image acquisition in this paper was divided into two parts. The first part involved using an industrial camera to capture 500 images of wafers in a real production line. The camera model used was MV-CA050-20GM, which is a 5-megapixel CMOS industrial camera with 4.8 μm pixel size and ML-MC 16HR lens. The camera shooting position was fixed. The wafer was transferred to the camera lens by a conveyor for each shot. Different wafer sizes and specifications were chosen for this test, so the size of the photolithographed characters, the type of font, the font format, and the background on which the characters are located would be different. A total of 100 images of characters from related studies that are difficult to recognize were collected in the second part by analyzing the related studies. The combined dataset was expanded to 1000 images by data processing. By randomly grouping the images through the algorithm in the program, 70% of the dataset was used as a training set, 10% as a validation set, and the other 20% was used to evaluate the performance of the proposed detection system.

The font type used was mainly Semi. The Semi character class contains 26 English letters, ten numbers, and two symbols. The Semi class is subdivided into traditional true type fonts (BcsSEMI), single-density dot matrix fonts (SEMI), and double-density dot matrix fonts (SEMIDouble). The three fonts are shown in Figure 2.

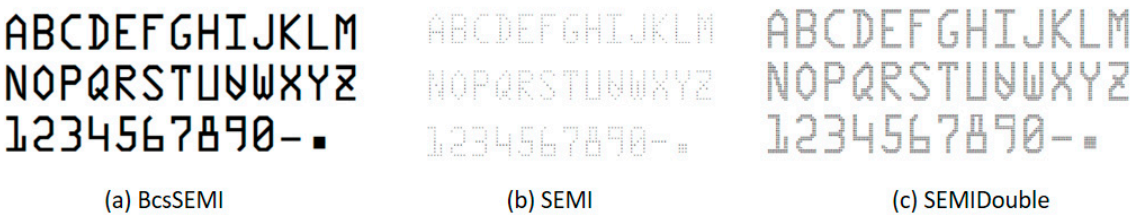

Figure 2. Schematic diagram of wafer character types.

3.1.2. Data Pre-Processing

Data augmentation is an effective method for comprehensively expanding datasets to enrich training data variations. The generic ability of the network can be improved by expanding the dataset in such a way that the overfitting problem of the network can be mitigated. In this paper, techniques such as brightness change, image rotation, and image scaling were used for dataset expansion. In order to simulate the actual situation of light interference in different factory environments, the robustness of the network to light interference was enhanced by using brightness adjustment on the image. When wafers were transported to the lens on the flow line, each wafer's relative position and angle under the lens would inevitably be different. Therefore, the actual angular deviation was simulated by rotating the original image at any angle. Moreover, the robustness of the training dataset was further improved by scaling the image sizes simultaneously. After pre-processing the captured photographs, the image characters were labelled using the RolabelImg tool. The process of labelling involves drawing the bounding boxes of different characters and distinguishing them by category.

3.2. The Proposed Improved YOLO Network Model

YOLO v7-Tiny is a lightweight improvement on YOLO v7. Compared to v7, the Tiny network is less complex, and the network parameters are less computationally intensive. Therefore, YOLO v7-Tiny is suitable for devices with small memory and low computation. At the same time, the lightweight model requires less training time, which satisfies the requirement of industrial production focusing on efficiency. However, due to the lightweight structure of YOLO v7-Tiny, its detection accuracy is lower than that of YOLO v7 and other algorithms, and the detection effect is weaker.

An improved YOLO model based on YOLO v7-Tiny was proposed to improve the detection performance while it was applied to small memory-embedded devices. The improved approach was based on an optimized attention mechanism module (CBAM-L), enhanced feature fusion structure (Bi-FPN), and a detection frame composed of adapted rotated characters.

3.2.1. Backbone Enhanced Feature Extraction Network

The backbone layer of the YOLO v7-Tiny network is located at the forefront of the overall structure and is used for feature extraction of the input image. The feature extraction process is composed of convolutional processing with sizes of 1×1 and 3×3, maximum pooling layer processing, and merging and activation function processing. After processing, the number of channels is expanded from 3 channels at the time of input image to 512 channels. When the feature extraction is in the low layer, more positional information is retained about the image. In contrast, more semantic information about the image is retained in the deep layer. The backbone network of YOLO v7-Tiny is not focused on feature extraction for a certain class of targets. Therefore, the detection accuracy is affected by a part of the useless information which is extracted. The targets of detection in this paper are wafer characters, including both words and symbols. With the long history of writing characters and people's reading habits, the characters are arranged in certain relative positional relationships to be convenient for people to read. Characters in the image are presented with certain spatial regularity in this positional relationship. Background

interference similar to character shapes can be eliminated using spatial regularity to reduce misjudgments and improve detection accuracy.

Therefore, in this paper, the CBAM (the spatial and channel feature attention mechanism) was introduced in the backbone network and improved [28]. The *LeakyRelu* function was introduced in CBAM to replace the original *Relu* function. When calculated, all values in the negative part of the axis are taken to be zero in the *Relu* function. In contrast, the negative values of the axes in the *LeakyRelu* function are multiplied by a very small constant factor called *Leak*. This way, when the input value is less than zero, the information is recorded fully, and the features are better preserved. The *LeakyRelu* function is calculated as shown in Equation (1). The principles of the improved attention module are shown in Figure 3.

$$LeakyRelu(x) = \begin{cases} x & , x > 0 \\ Leak * x & , x \leq 0 \end{cases} \quad (1)$$

where the coefficient *Leak* is usually set to 0.01.

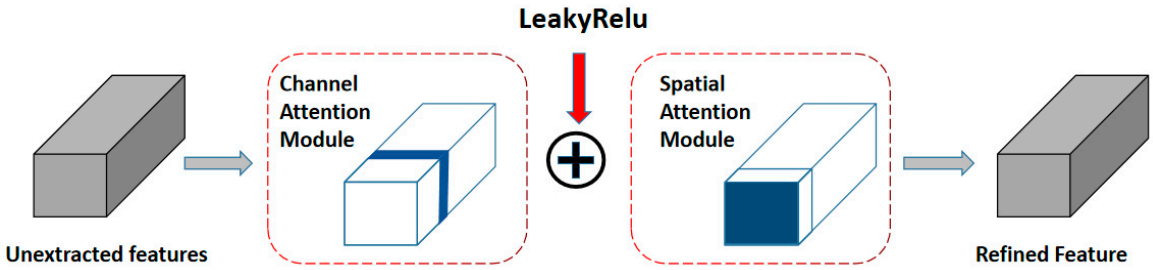

Figure 3. CBAM-L schematic.

The improved spatial channel attention mechanism module, CBAM-L, was added to the backbone feature extraction network. At the same time, it was combined with the C5 module in the backbone network to form the C5-CL module. Through the improved network, the attention to the target spatial feature information was strengthened in the process of information extraction to improve detection accuracy and reduce the interference of similar backgrounds on the target characters. The detailed structure of the improved backbone feature extraction network is shown in Figure 4.

Type	Input Channal	Size/Layer	Output Channal	Layer Position	
CBAM-L	3	3×3	32	0	
CBAM-L	32	3×3	64	1	
CBAM-L	64	1×1	32	2	
CBAM-L	32	1×1	32	3	
CBAM-L	32	3×3	32	4	
CBAM-L	32	3×3	32	5	← C5-CL
Concat		[−1, −2, −3, −4]		6	
CBAM-L	128	1×1	64	7	
MP	64		64	8	
C5-CL	64		128	9–14	← Stage One
MP	128		128	15	
C5	128		256	16–21	← Stage Two
MP	256		256	22	
C5	256		512	23–28	← Stage Three

Figure 4. The detailed structure of the improved backbone feature extraction network.

3.2.2. Feature Fusion Enhancement Network

There are top-down and bottom-up paths in the feature fusion part of YOLO v7-Tiny. Compared to the FPN network structure with only top-down feature fusion, this structure makes it easier to pass the bottom information to the top. However, after two paths, top-down and bottom-up, the path of information flow is too long. The connection between the feature layer of the backbone output and the bottom-up path is not tight enough, and some important information plays a minor role in the fusion.

Therefore, in this paper, the structure of the feature fusion part of the YOLO v7-Tiny network was improved by combining the BiFPN network structure. In the bottom-up path, a path was added to connect the information output from the backbone network directly to the fusion node, thus increasing the tightness of the information flow. Moreoever, the fusion nodes were combined with the band-weighted fusion method. In contrast to simply overlaying or adding feature maps, different input feature maps are set with different contribution weight values by the weighted fusion method. The importance of different input features can be understood by setting the weights so that different input features are fused in a differentiated manner. There are three common ways of weighted feature fusion:

a. Unbounded fusion: This method, although simple, may be unstable during training because the weights are unconstrained. The formula is shown in Equation (2).

$$O = \sum_i W_i \times I_i \tag{2}$$

b. Softmax-based fusion: The weight range is limited to [0, 1] by this method, and the training effect is stable but slow. The formula is shown in Equation (3).

$$O = \sum_i \frac{e^{W_i} \times I_i}{\varepsilon + \sum_j W_j} \tag{3}$$

c. Fast normalized fusion: Not only is the weight range limited to [0, 1] by this method, but the training is faster and more efficient. The formula is shown in Equation (4).

$$O = \sum_i \frac{W_i \times I_i}{\varepsilon + \sum_j W_j} \tag{4}$$

Therefore, the Fast normalized fusion method was adopted in this paper. The multi-branch fusion module Concat was replaced by the multi-branch fusion module Bi_Concat with entitled values. At the same time, the feature fusion network structure of YOLO v7-Tiny was improved by combining the BiFPN network structure. At the 40th layer position, the feature layer output from the 21st layer was processed with the 1×1 Conv, BN, and *LeakyRelu* activation functions. The processed feature layer was connected to Bi_Concat in layer 59 to strengthen the tightness of information flow between layers, making the information fusion richer and more valuable.

3.2.3. Character Rotation Detection

When calculating the loss function, *IoU* is one of the important calculation indexes. It is the ratio of the intersection and concatenation of the ground truth and the prediction of the detected target. The closer the calculated value is to 1, the closer the prediction is to the real target, as shown in Figure 5.

R1 is the ground truth region in the figure, and *R*2 is the prediction region. The *IoU* equation is shown in Equation (5).

$$IoU = \frac{|R1 \cap R2|}{|R1 \cup R2|} \tag{5}$$

The prediction frames are rectangular frames of different sizes with horizontal bottom edges in most of the detection. However, for targets with tilted angles, a rectangular frame

with a horizontal bottom edge will have a weak fit to the target. In a factory production operation, each time a wafer was transferred under the camera, as shown in Figure 6, the tilt of the characters was likely to change between shots.

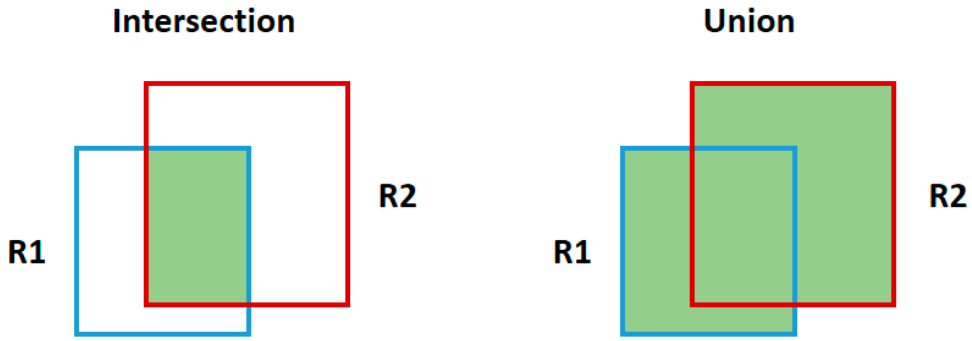

Figure 5. Schematic diagram of intersection and concatenation.

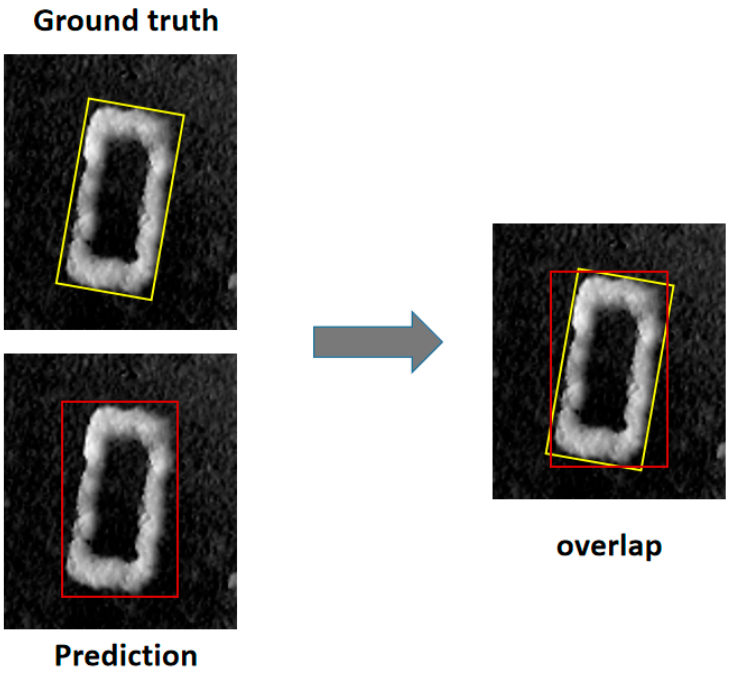

Figure 6. Comparison of ground truth and prediction regions.

Therefore, to better cope with the weak fit between the detection frame and the tilted font, the rotation angle *theta* was added to the Head of YOLO v7-Tiny in this paper. The position of the prediction box was adjusted by five parameters (*xc*, *yc*, *w*, *h*, *theta*). Inclined targets could be better included, reducing the area of the connecting region between ground truth and prediction, increasing the area of the intersecting region, and improving the overlap, thus increasing the *IoU* value. The overall structure of the improved model is shown in Figure 7.

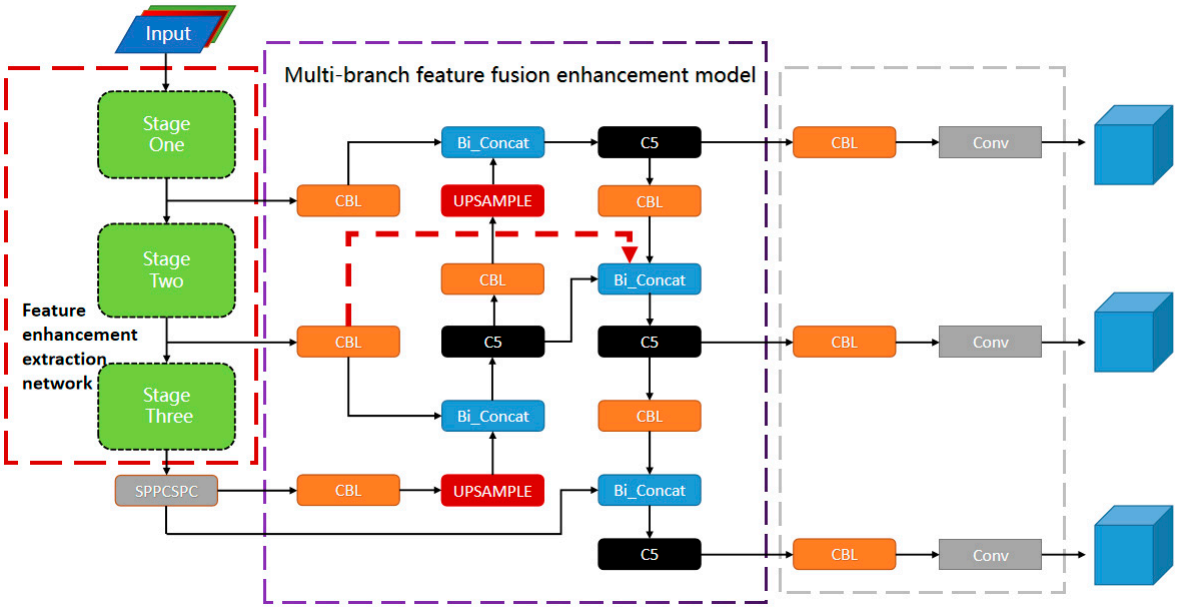

Figure 7. Schematic diagram of the overall structure of the improved network. The dotted line in the red group in the figure shows the newly added paths.

4. Experiments and Discussion

Several experiments were conducted to verify the reliability and enhancement of the improved model proposed in this paper for wafer character detection. The experimental setup was first described. Then, the evaluation metrics were explained. Finally, the experimental results were analyzed and discussed. The experiments in this paper were conducted on a dataset composed of self-taken and collected images, and the data results of the improved model proposed were compared with those of several YOLO models.

4.1. Experimental Setup

All experiments were performed on the Windows 11 64-bit operating system platform. The AMD Ryzen-7-5800H 3.20 GHz processor, 16 GB RAM, and NVIDIA GeForce RTX 3050Ti graphics card were used. The YOLO-based models were trained in the Pytorch framework. Hyperparameters were normalized for all YOLO-based models. The image size was set to 640 × 640 pixels and the number of iterations was set to 250. The dataset images were captured by a 5-megapixel CMOS industrial camera and contained 1000 character-images after data processing. The deep learning model was run using Python 3.7.16.

4.2. Evaluation Metrics

Several key data metrics were referenced to evaluate the reliability and validity of the recognition model in terms of detection and computation [29].

Pr (precision) and *Rc* (recall) are important metrics to show the performance of the model. *Pr* is the proportion of samples judged to be "true" in all systems that are actually true. When the ground truth boxes are matched by the predicted bounding boxes, *Pr* measures the correct prediction. *Rc* is the proportion of samples that are judged to be true out of the total number of samples that are indeed true. It also measures the probability of correct detection of ground truth objects. *Pr* and *Rc* are calculated as shown in Equations (6) and (7).

$$Pr = \frac{TP}{TP + FP} \tag{6}$$

$$Rc = \frac{TP}{TP + FN} \tag{7}$$

TP (True Positive) indicates that the positive category is predicted as positive, i.e., correct prediction; *FN* (False Negative) indicates that the positive category is predicted as negative, i.e., incorrect prediction; *FP* (False Positive) indicates that the negative category is predicted as positive, i.e., incorrect prediction; and *TN* (True Negative) indicates that the negative category is predicted as negative, i.e., correct prediction. The details are shown in Table 1. The positive category is indicated as 1, and the negative category as 0.

Table 1. Classification of sample prediction results.

Categories	Real Value	Predicted Value
TP	1	1
FN	1	0
FP	0	1
TN	0	0

The area enclosed by *Pr* and *Rc* at different thresholds is *AP* (average precision). A P–R curve can be formed by *Pr* and *Rc*. If the P–R curve of one model is completely enclosed by the curve of the other model, it can be asserted that the latter model outperforms the former. However, sometimes the two worse curves are difficult to compare, so the *F*1 scores are introduced.

An *F*1 score is defined as the reconciled average of precision and recall, often used as the final measure in some multiclassification problems. The value of *F*1 scores ranges from 0 to 1, with 1 being the best and 0 the worst. *AP* is calculated as shown in Equation (8), and *F*1 score is calculated as shown in Equation (9).

$$AP = \int_0^1 Pr(Rc)dRc \tag{8}$$

$$F1 = 2 \times \frac{Pr \times Rc}{Pr + Rc} = \frac{2 \times TP}{2 \times TP + FN + FP} \tag{9}$$

In addition, the detection time required by the model to detect the images, the number of model parameters, and the model loss function curve were calculated to evaluate the detection speed as well as the detection performance of the model.

4.3. Experimental Results

The proposed improved model was measured in this section. The performance of different models obtained at different locations with the improved attention module was compared. Meanwhile, the improved YOLO model proposed in this paper was compared with several other state-of-the-art detection models (including YOLO v7, YOLO v7-Tiny, and YOLO v5s) in terms of detection speed and detection accuracy to validate the performance of the improved model.

4.3.1. Comparison of Performance

Firstly, ablation experiments were performed on the improved part of the backbone network. The effect of adding the improved CBAM-L and C5-CL modules at different locations in the backbone network on model performance was compared. In each set of experiments, the feature fusion network and the detection part were kept the same, and the positions of CBAM-L and C5-CL in the backbone network were changed. The reasonability of the improvement scheme proposed in this paper was verified by comparing the loss function curves of the experimental output of each group. The specific added positions and loss function curves are shown in Table 2 and Figure 8.

Table 2. Comparison table of module-specific addition positions.

Experiment No.	CBAM-L Addition Layer Location	C5-CL Addition Layer Location	Loss
1	0, 1 Layer	None	0.01696
2	0, 1 Layer	2–7, 9–14 Layer	0.0162
3	0, 1 Layer	2–7, 9–14, 16–21 Layer	0.01736
4	None	2–7, 9–14, 16–21 Layer	0.01762

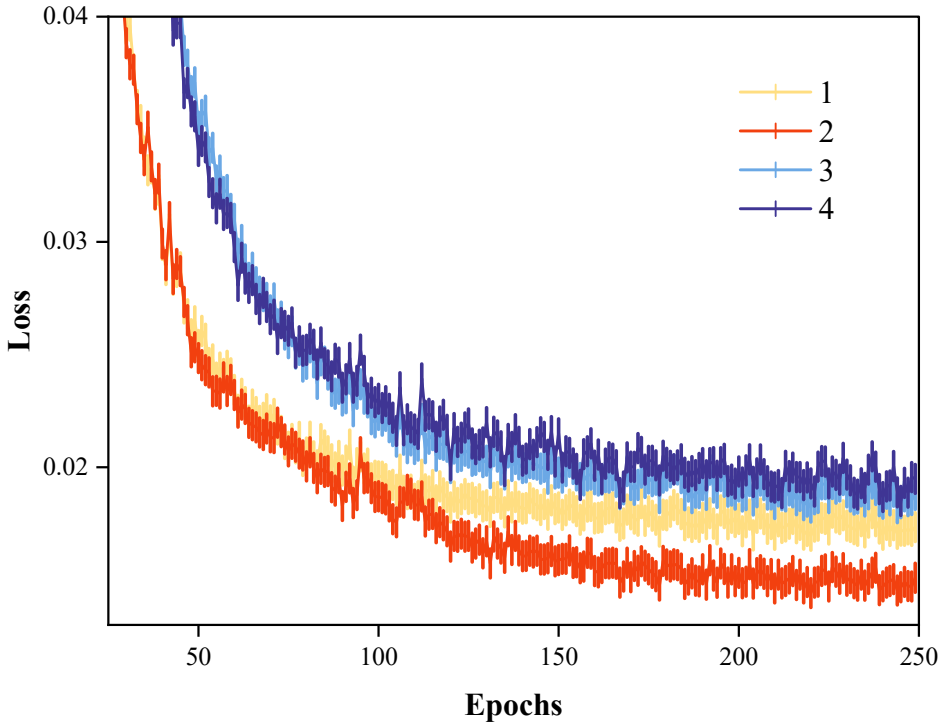

Figure 8. Loss curves of CBAM-L vs. C5-CL module ablation experiments.

It was shown that, after the convergence of the loss function curves, the convergence values from the first group to the fourth group were 0.01696, 0.0162, 0.01736, and 0.01762, respectively. From the loss function data, it was clear that the second group of ablation experiments had the smallest values. Therefore, the module addition location used in this paper was the best location to improve the performance of the whole network. It was also illustrated by the experimental data that the appropriate location of the improvement was important for the improvement of the performance of the model.

The improved YOLO model proposed in this paper was compared with YOLO v7, YOLO v7-Tiny, and YOLO v5s. The performance of the improved model in this paper was verified by comparing the final loss function curve and the *AP* curve when the training phase was stable. The resulting curves are shown in Figure 9. When the function converged, the lowest value of the loss function of each model were obtained as 0.0151, 0.0155, 0.0178, and 0.0226, in that order. From the data, it could be seen that the loss function of the improved model was significantly lower than that of YOLO v7-Tiny and YOLO v5s, and slightly lower than that of the YOLO v7 model with a complex structure. Therefore, the improvements in this paper could be verified to have a significant improvement on the model performance.

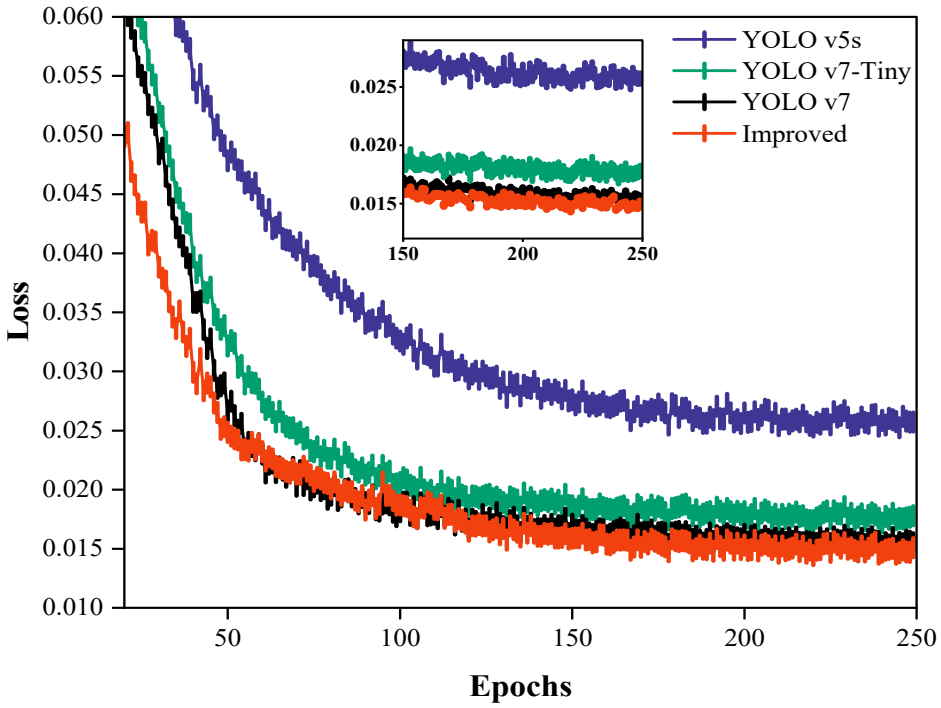

Figure 9. Plot of the training loss function for each model.

Then, the performance of the improved model in this paper was further verified from the perspective of the *AP* value. An *AP* comparison curve based on the YOLO model is shown in Figure 10. According to the curves in the figure, the *AP* values obtained from the training of the improved model were significantly higher than those of YOLO v7-Tiny and YOLO v5s. The specific values of each model's performance are shown in Table 3. It is worth noting that, among the improved models, YOLO v7-Tiny and YOLO v5s, the improved model obtained the highest *mAP* value at 0.5 *IoU*, with a value of 0.9944; the *mAP* value of YOLO v7-Tiny was 0.9796 and that of YOLO v5s was 0.9463. The improved model was 1.48% better than the original model. YOLO v5s is the lightweight model in the YOLO v5 series, with a backbone consisting only of convolutional layers and a C3 module with fewer layers. Hence, the *mAP* value obtained was lower than that of the YOLO v7-Tiny model with more layers in the backbone network. The attention module was added to the backbone network of the improved model. The structural complexity and information exchange density of the feature fusion part was also increased. As a result, the improved model was better at extracting and fusing target image features.

The outperformance of the improved model proposed in this paper over the v7-Tiny model and the YOLO v5s model was also verified in terms of higher *mAP* values. At *IoU* = 0.5:0.95, the *mAP* value obtained by the model was 0.7711, which was higher than the 0.7459 and 0.7588 obtained by the YOLO v5s and YOLO v7-Tiny models, respectively. At high *IoU* thresholds, it could be proven that the detection frames of the model were closer to the tilted characters at different angles. In terms of *F*1 score, the score of the improved model in this paper was 0.97, which was close to the score of YOLO v7 and higher than the scores of the other two YOLO models. Therefore, the improved model was better in terms of overall precision and recall.

YOLO v7 and the improved model were compared using the data shown in Table 3. Although some values of the improved model were slightly lower than v7 in terms of detection performance, the difference in *mAP* values was 0.0013 at the *IoU* = 0.5 threshold, 0.0104 at *IoU* = 0.5:0.95, and 0.01 at *F1* score. However, in terms of the number of model parameters and structural complexity, it can be seen from Table 4 that the number of parameters in YOLO v7 was 5.3 times higher than that of the improved model, and the training time was 1.99 times higher. While the computational performance of the model

could be improved by more convolutional layers and complex structures, the size of the model was greatly increased.

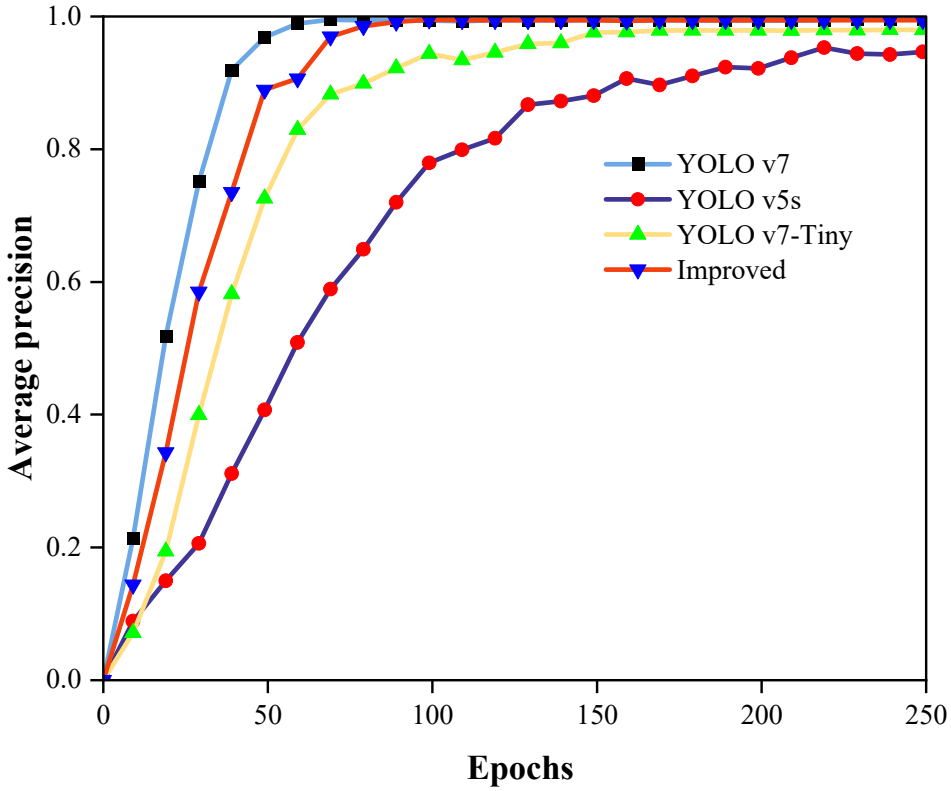

Figure 10. AP curves for each model.

Table 3. Comparison of detection performance for several detection models.

Method	Loss	*mAP*@0.5	*mAP*@0.5:0.95	*F*1 Score	*Pr*	*Rc*
YOLO v7	0.0155	0.9957	0.7815	0.98	0.9698	0.9905
YOLO v7-Tiny	00178	0.9796	0.7588	0.94	0.9533	0.941
YOLO v5s	0.0226	0.9463	0.7459	0.90	0.8947	0.9103
Improved YOLO	**0.0151**	**0.9944**	**0.7711**	**0.97**	**0.9551**	**0.9906**

Table 4. Computational performance of detection models.

Method	Training Time	Parameters (Millions)	Detection Time (s)
YOLO v7	6 h 22 min	32.42	7.307
YOLO v7-Tiny	2 h 54 min	6.10	4.395
YOLO v5s	3 h 7 min	7.11	4.282
Improved YOLO	**3 h 12 min**	**6.13**	**4.61**

Smaller size models are more suitable for small memory-embedded devices in industrial production. At the same time, the requirements for hardware configuration and CPU performance are lower for small-sized models. Therefore, it is very important to develop small-size models. The performance was improved for various detections while ensuring the appropriate model size. In Table 4, the improved model was shown to be at the same level as YOLO v7-Tiny and YOLO v5s in terms of the number of parameters. However, the improved model was better in terms of performance, verifying the effectiveness and usability of the improvement scheme proposed in this paper.

4.3.2. Detection on Factory Test Images

One hundred images were collected by an industrial camera in a practical factory environment as a dataset for the experimental detection of the model to further test the reliability of the improved model. Different lighting conditions, wafer specifications, character tilt angles, and complex character background interferences were contained in the images in the dataset. The detection robustness and accuracy of the improved model were tested through the dataset images. Some images of the detection results at different brightness levels on a simple background are given in Figure 11. The size and shape of the characters are different depending on the specification and type of the wafer. Some of the detected images for different font formats are given in Figure 12. The effectiveness of character detection was greatly affected by the complexity of the background. Most of the background interference was from the pattern of the lattice, the surrounding colloid, and scratches on the wafers during shipping and storage. Several images of the effect of wafer character detection on different complex backgrounds are given in Figure 13. Finally, the effect image adapted for character tilt detection is shown in Figure 14. With the detection result graphs, the improved model can be proven to have higher detection accuracy and better robustness under different lighting, complex background, and font shape conditions.

Figure 11. Plot of detection results for different light intensities on simple backgrounds.

Figure 12. Detection results of different kinds of characters on a simple background.

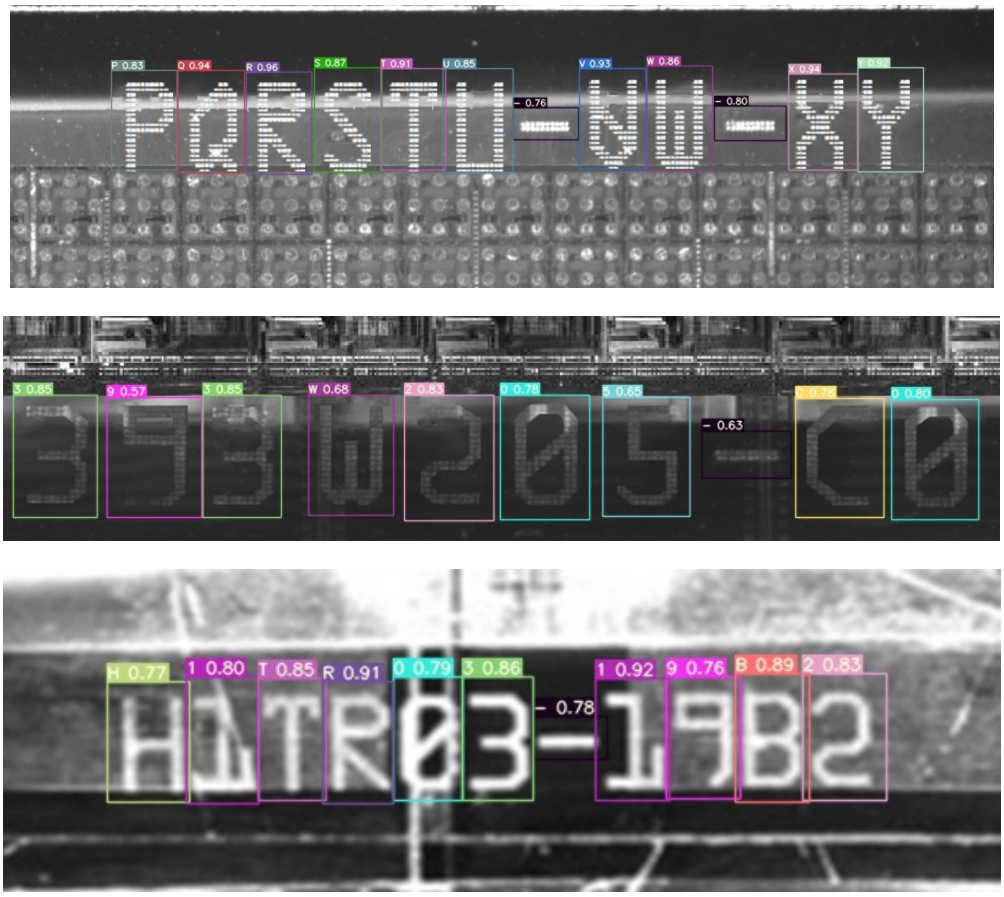

Figure 13. *Cont.*

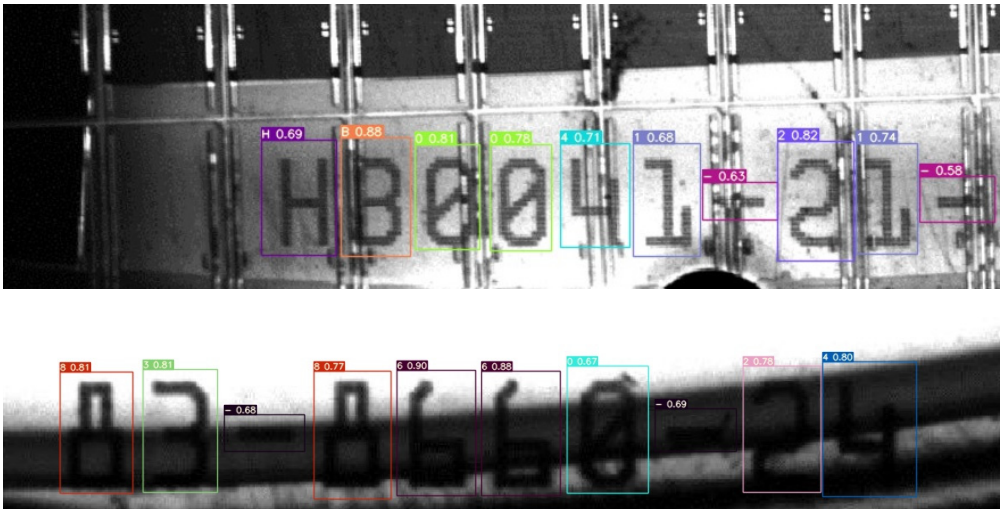

Figure 13. Complex background wafer character detection results.

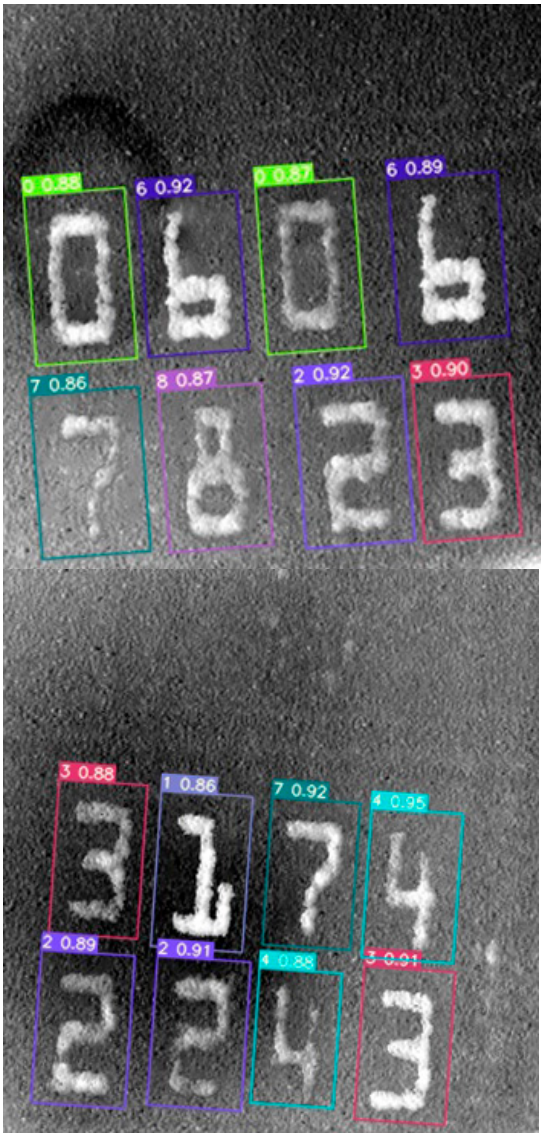

Figure 14. Tilt character detection results.

5. Conclusions

An improved model based on YOLO v7-Tiny was proposed in this paper for wafer character detection on complex backgrounds in industrial production environments. The CBAM-L module was improved from CBAM, where the *Relu* function was improved to the *LeakyRelu* function. Negative axis information could be effectively retained by the improved module during data computation. The CBAM-L module was combined with the C5 module in the backbone network to form the C5-CL module. The feature extraction capability was improved by adding the C5-CL module into the backbone network, thus enhancing the spatial attention to the character images. Then, the feature fusion network part was improved based on the BiFPN structure to enhance the feature fusion capability of the network. Finally, the angle parameter was added to the detection part to accommodate the character image detection with angular deviation and to improve the overlap between the prediction and the ground truth. It was shown that the performance of the improved YOLO model was better in terms of detection and computation. The convergence values for the loss function of the model were 0.0151, 0.9944 for *mAP*@0.5, 0.7711 for *mAP*@0.5:0.95, and 0.97 for *F*1 score. As a result of the overall analysis, it can be concluded that the developed improved model was suitable for small memory-embedded capture devices in industry and was more effective in detecting wafer characters. With this method, visual information could be provided for the development of a wafer character detection system under complex backgrounds. With the improved model, higher accuracy and better robustness were achieved in detecting fonts under different lighting, complex backgrounds, and font shapes.

Author Contributions: Conceptualization, J.X., Y.Z. and P.H.; methodology, J.X., Y.Z. and P.H.; software, J.X. and Y.Z.; validation, Y.Z. and P.H.; formal analysis, J.X., Y.Z. and P.H.; investigation, Y.Z.; data curation, Y.Z. and P.H.; writing—original draft preparation, Y.Z.; writing—review & editing, J.X., Y.Z. and P.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All data in this paper are standard datasets. The datasets taken in this paper are authentic and valid.

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

References

1. Fujisawa, H. Forty years of research in character and document recognition—An industrial perspective. *Pattern Recognit.* **2008**, *41*, 2435–2446. [CrossRef]
2. Wu, X.W.; Sahoo, D.; Hoi, S.C.H. Recent advances in deep learning for object detection. *Neurocomputing* **2020**, *396*, 39–64. [CrossRef]
3. Olszewska, J.I. Active contour based optical character recognition for automated scene understanding. *Neurocomputing* **2015**, *161*, 65–71. [CrossRef]
4. Tian, J.M.; Wang, G.Y.; Liu, J.G.; Xia, Y.C. Chinese license plate character segmentation using multiscale template matching. *J. Electron. Imaging* **2016**, *25*, 053005. [CrossRef]
5. Chen, H.X.; Ding, X.Y. Research on license plate recognition based on template matching method. *Appl. Mech. Mater.* **2014**, *668–669*, 1106–1109. [CrossRef]
6. Ryan, M.; Hanafiah, N. An examination of character recognition on ID card using template matching approach. *Procedia Comput. Sci.* **2015**, *59*, 520–529. [CrossRef]
7. Zhang, Y.; Zha, Z.Q.; Bai, L.F. A license plate character segmentation method based on character contour and template matching. *Appl. Mech. Mater.* **2013**, *333–335*, 974–979. [CrossRef]
8. Jung, M. 7-Segment Optical character recognition using template matching. *J. Electron. Disp. Technol.* **2020**, *19*, 130–134.
9. Weng, Y.; Xia, C.L. A new deep learning-based handwritten character recognition system on mobile computing devices. *Mob. Netw. Appl.* **2019**, *25*, 402–411. [CrossRef]
10. Kim, H.-H.; Park, J.-K.; Oh, J.-H.; Kang, D.-J. Multi-task convolutional neural network system for license plate recognition. *Int. J. Control Autom. Syst.* **2017**, *15*, 2942–2949. [CrossRef]
11. Yang, Y.; Li, D.H.; Duan, Z.T. Chinese vehicle license plate recognition using kernel-based extreme learning machine with deep convolutional features. *IET Intell. Transp. Syst.* **2017**, *12*, 213–219. [CrossRef]

12. Rakhshani, S.; Rashedi, E.; Nezamabadi-pour, H. Representation learning in a deep network for license plate recognition. *Multimed. Tools Appl.* **2020**, *79*, 13267–13289. [CrossRef]
13. Cao, F.; Tian, Z.G.; Jiang, B.Z.; Zhang, H.S.; Chen, H.; Zhu, X.G. 3D Model Registration-Based Batch Wafer-ID Recognition Algorithm. *IEEE Access* **2021**, *9*, 150283–150291. [CrossRef]
14. Hsu, W.C.; Yu, T.Y.; Chen, K.L. Wafer identification recognition by stroke analysis and template matching. *Sens. Lett.* **2012**, *10*, 1223–1229. [CrossRef]
15. Soora, N.R.; Deshpande, P.S. Review of feature extraction techniques for character recognition. *IETE J. Res.* **2017**, *64*, 280–295. [CrossRef]
16. Shao, J.H. Research on Wafer Identifier Recognition System in Complex Background. Master's Thesis, Harbin Institute of Technology, Harbin, China, 2021.
17. Xu, Z.H.; Yang, J.; Zhang, W.J.; Yang, Z.J. Regions of interest detection algorithm based on improved visual attention model. *Appl. Mech. Mater.* **2014**, *513–517*, 3368–3371. [CrossRef]
18. Jiang, T.Y.; Li, C.; Yang, M.; Wang, Z.L. An improved YOLOv5 algorithm for object detection with an attention mechanism. *Electronics* **2022**, *11*, 2494. [CrossRef]
19. Zang, D.; Chai, Z.L.; Zhang, J.Q.; Zhang, D.D.; Cheng, J.J. Vehicle license plate recognition using visual attention model and deep learning. *J. Electron. Imaging* **2015**, *24*, 033051. [CrossRef]
20. Deng, C.F.; Wang, M.M.; Liu, L.; Liu, Y.; Jiang, Y.L. Extended feature pyramid network for small object detection. *IEEE Trans. Multimed.* **2022**, *24*, 1968–1979. [CrossRef]
21. Wang, P.; Wang, Y.L.; Jiao, B.W.; Wang, H.C.; Yu, Y.X. Research on road target detection algorithm based on YOLOv5. *Comput. Eng. Appl.* **2023**, *59*, 117–125. [CrossRef]
22. Zhu, F.Z.; Wang, Y.Y.; Cui, J.Y.; Liu, G.X.; Li, H.L. Target detection for remote sensing based on the enhanced YOLOv4 with improved BiFPN. *Egypt. J. Remote Sens. Space Sci.* **2023**, *26*, 351–360. [CrossRef]
23. Yang, Y.; Pan, Z.X.; Hu, Y.X.; Ding, C.B. CPS-Det: An anchor-free based rotation detector for ship detection. *Remote Sens.* **2021**, *13*, 2208. [CrossRef]
24. Fu, K.; Li, Y.; Sun, H.; Yang, X.; Xu, G.L.; Li, Y.T.; Sun, X. A ship rotation detection model in remote sensing images based on feature fusion pyramid network and deep reinforcement learning. *Remote Sens.* **2019**, *10*, 1922. [CrossRef]
25. Hee, L.M.; Pang, Y.Y.; Ong, C.H.; Sim, H.M. Deep learning convolutional neural network for unconstrained license plate recognition. *MATEC Web Conf.* **2019**, *255*, 05002. [CrossRef]
26. Li, C.L.; Zhang, Q.H.; Chen, W.H.; Jiang, X.B.; Yuan, B.; Yang, C.L. Insulator orientataion detection based on deep learning. *J. Electr. Inform. Technol.* **2020**, *42*, 1033–1040. [CrossRef]
27. Shi, P.F.; Zhao, Z.X.; Fan, X.N.; Yan, X.J.; Yan, W.; Xin, Y.X. Remote sensing image object detection based on angle classification. *IEEE Access* **2021**, *9*, 118696–118707. [CrossRef]
28. Fu, G.D.; Huang, J.; Yang, T.; Zheng, S.Y. Improved lightweight attention model based on CBAM. *Comput. Eng. Appl.* **2021**, *57*, 150–156.
29. Junos, M.H.; Mohd Khairuddin, A.S.; Thannirmalai, S.; Dahari, M. Automatic detection of oil palm fruits from UAV images using an improved YOLO model. *Vis. Comput.* **2021**, *38*, 2341–2355. [CrossRef]