Digital Signal Processing 130 (2022) 103718

Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

A wafer surface defect detection method built on generic object detection network

Xinyu Wang [a], Xiaoli Jia [b], Chuyi Jiang [c], Sanxin Jiang [a,*]

[a] Department of Information and Communication Engineering, Shanghai University of Electric Power, Shanghai, China
[b] School of Electronics Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China
[c] Faculty of engineering, University of Bristol, Bristol, BS8 1UH, UK

ARTICLE INFO

Article history:
Available online 7 September 2022

Keywords:
Surface defect detection
Object detection
NMS
Machine vision
Deep neural network

ABSTRACT

Precise locating and correct classification of each defect in the image is the primary goal of surface defect detection (SDD). This is very similar to object detection, which seeks to predict a set of bounding boxes (Bboxes) and category labels for each object of interest. However, due to the fact that the category dependencies among defects are obviously different from those among general objects, the methods developed for object detection cannot be directly utilized for detecting defects. To address this issue, we proposed a category-related non-maximum suppression (CR-NMS) method. Different from most NMS methods, CR-NMS use Cover Percent (CoP), instead of Intersection over Union (IoU), to guide the Bbox regression. Moreover, a two-stage Bbox regression algorithm is proposed to remove the duplicate Bboxes. This algorithm works in a course-to-fine manner and takes the correlation between Bboxes into account. By this means, the CR-NMS can remove the duplicate Bboxes more effectively and is easy to embed in existing neural networks. A wafer surface defect dataset including 6,000 images and 11 defect categories was set up for training and evaluating our method. Experiments demonstrated that in the detectors, which were built on Faster R-CNN and RetinaNet with backbone ResNet-101, and YOLOX with backbone CSPDarknet, compared with STD NMS or Soft NMS, CR-NMS increased by 3.4%, 9.3%, and 5.5% on mean average accuracy (mAA), and their values were as high as 92.3%, 91.8%, and 90.8%, respectively. The same tests were also performed on dataset NEU-DET. The results showed that compared with STD-NMS and Soft NMS, the CR-NMS can also obviously improve the performance of all the detectors, and their mAA was improved by at least 1.2%.

1. Introduction

Surface defect detection (SDD) [31,8], which uses machine vision equipment to obtain images and then determines whether there are defects in the collected images, is a very important research area in the field of machine vision. At present, surface defect detection equipment based on machine vision has widely replaced manual visual inspection in various industrial fields, including automobile, machinery manufacturing, chemical industry, medicine, aerospace and other industries, especially in semiconductor and electronics. As a result, more and more SDD methods are developed to meet these needs.

Classical SDD methods, such as [20,37,7,22,30], usually use a well-designed imaging system to obtain uniformly illuminated images, which helps to show the surface defects clearly. For the sake of simplicity, one way is to select an optimal light source according to the color of the inspected surface. For example, Jing et al. [20] utilized a composite white light source to image the color cloth surface defects. Another common way is to select the best imaging scheme according to the reflection property of the detected surface, such as bright field imaging, dark field imaging, and hybrid imaging. Tao et al. [37] used dark-field imaging to detect faint scratches on the surface of large-aperture optical elements. To detect defects and deformations across the imaged can-end surface, Chen et al. [7] designed two concentric conical ring bright field light sources to illuminate the central and peripheral areas of the bottom of cans at the same time. Although a well-designed imaging system can greatly reduce the design difficulty of the classical SDD algorithm, it also increases the application cost. Additionally, in order to identify defects, classical SDD methods usually use hand-craft features as its basis. Since the determination of features is too subjective, the human experience usually plays a decisive role in it, which results in the features are hard to grasp. In prac-

* Corresponding author.
E-mail addresses: xinyuwang@mail.shiep.edu.cn (X. Wang), jiaxlm8601@126.com (X. Jia), ir19714@bdistil.ac.uk (C. Jiang), samjoe_2018@shiep.edu.cn (S. Jiang).

https://doi.org/10.1016/j.dsp.2022.103718

tice, classical SDD may face more challenges, such as slight difference between defects and background, low contrast, large change of defect scale and various types, and even a large amount of noise. At this time, the classical methods are often helpless and difficult to achieve good detection results.

In recent years, with the successful application of deep learning (DL) technique, especially convolution neural networks (CNNs) [34,13,18], in many computer vision fields, lots of CNN-based SDD methods have been developed. These methods can generally be divided into two categories: classification and localization. Due to the strong feature extraction ability of CNN, the classification network based on CNN has become the most commonly used model in surface defect classification. They use the classical CNN as their backbone, such as Krizhevsky et al. [21] used AlexNet, Simonyan and Zisserman [34] used VGG, Szegedy et al. [35] used GoolLeNet, He et al. [13] used ResNet, Huang et al. [18] used DenseNet, Hu et al. [17] used SENet, Howard et al. [16] used MobileNet, Zhang et al. [39] used ShuffleNet and so on. In some other tasks, only knowing the category is not enough, we also want to know the location of them, which is very similar to object detection. Therefore, many networks developed for object detection, such as Faster R-CNN in [33], SSD in [28] and YOLO in [32], are adapted to SDD. Cha et al. [3] first applied Faster R-CNN directly to bridge surface defect location, and its backbone network was replaced by ZF net. Li et al. [23] optimized the backbone structure of SSD through MobileNet, and proposed a MobileNet-SSD-based method for the detection of container sealing surface defects. Liu et al. [29] used MobileNet-SSD network to locate the high-speed rail catenary support group. He et al. [14] proposed a novel defect detection system based on DL for steel plate defect inspection. They set up a hot-rolled steel surface defect dataset NEU-DET for network training and evaluating. Cui et al. [9] constructed a SDDNet for textured surface defect detection. Compared with classical SDD methods, which use a combination of low-level hand-craft features, CNN-based methods utilize deep neural networks to extract high-level abstract representations, which have a strong representation ability, as a result, they achieve significant performance improvements. However, these methods focus primarily on the existence of a certain defect and the localization of it, while little consideration of the causes behind defects. In fact, the factors that cause surface defects are correlated each other, and taking advantage of these relations could help to defects detection.

To address this issue, we proposed a category-related non-maximum suppression (CR-NMS) method. Generally, the appearance of a defect is not random, more over, there is a strong correlation between defects, especially in mass production. Within all possible defects, some of them coexist side by side, some of them are mutually exclusive, and some others may appear in groups. This is very different from a general object detection. Taking advantage of these correlations will obviously improve the performance of SDD. To this end, our method uses parameter, Cover Percent (CoP), to denote the degree of overlap of two boxes. Furthermore, the category of each bounding box is also taken into account when removing the duplicated ones. In CR-NMS, the bounding boxes are regressed in a coarse-to-fine manner, where the confidence and types of each bounding box are taken into account. In addition, CR-NMS does not require additional supervision and is easy to embed in existing networks. We constructed a wafer surface defect detector on CR-NMS, which has been shown to be effective in improving defect recognition and duplicate removal steps.

In brief, the contributions of the proposed method are as follows.

1. Parameter CoP is proposed, which is used to indicate the extent to which one bounding box is covered by another. The CoP describes the geometric relation of two candidate bounding boxes, and is used to guide the bounding box regression.
2. A coarse-to-fine regression algorithm is proposed to remove the duplicated bounding boxes. In the process of defect detection, the neural network would generate lots of bounding boxes. To preserve the optimal bounding boxes, the algorithm needs to consider not only the geometric constrains between any two bounding boxes, that is CoP, but also the category constraints between them.
3. To evaluate the performance of defect detection on multi-category dataset, we introduced four metrics, mAA, mAE, mAU, and mAO. This is based on the fact that for each defect, there may be four possible judgments provided by the detector: correctness, error, uncertainty, and omission.
4. A dataset is built to evaluate our wafer defects detector. In semiconductor chip manufacturing, there are many factors that can cause wafer defects. Thus, the defects can be classified into multiple categories according to their causes. In this paper, we build a wafer surface defect dataset including 6,000 images (with 512×512) labeled for 11 types of defects.

2. Related works

Just like generic object detection, SDD is also a multi-task learning problem and usually consists of defect classification and localization. Moreover, many current state-of-the-art defect detectors also rely on bounding box regression to localize defects.

2.1. Defect detector

At present, defect detectors are usually built on CNN, which is capable of extracting the high-level features of the defects. In fact, defect detection is very similar to generic object detection, they both seek to predict a set of bounding boxes and category labels for each object of interest. Consequently, the frameworks devised for generic object detection can usually be adapted for defect detection. For example, Cha et al. [3] modified faster R-CNN to provide quasi real-time detection of multiple types of surface structural damages. Subsequently, Faster R-CNN was applied to more SDD tasks in various industrial fields, such as [41,36,14,38], either by improving its convolution network or by introducing new feature fusion method. Chen et al. [4] cascaded two detectors (SSD and YOLO) and a classifier to form a three-stage deep CNN-based detector, which is able to sequentially localize the cantilever joints and their fasteners and to diagnose the fasteners' defects in a coarse-to-fine manner. Liu et al. [29] proposed a framework, which cascades the coarse positioning network and the fine positioning network, to reduce multi-scale differences between different catenary support components.

Some other CNN-based methods are also developed in recent years. To handle large texture variation and small size of defects simultaneously, Cui et al. [9] introduced two modules, feature retaining block module and skip densely connected module to build a SDD network. He et al. [11] proposed a novel defect detection system based on DL for steel plate defect inspection. However, these methods are used to detect single type defect, and their structures are relatively simple.

2.2. Bounding box regression

In many object detectors, bounding box regression is implemented by NMS. The standard NMS (STD-NMS) was first proposed by Ren et al. [33], in which intersection-over-union (IoU) is used to evaluate the overlap relationship between bounding boxes. And the duplicated bounding boxes with an IoU greater than the threshold are removed based on the assumption that the bounding box with

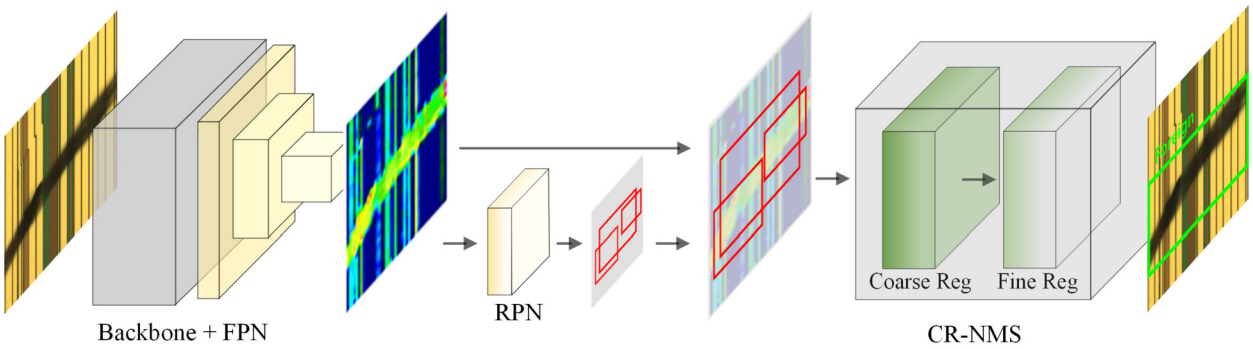


Fig. 1. The framework of our wafer surface defect detector, where the CR-NMS is utilized to remove the duplicated bounding boxes in a coarse-to-fine manner.

a higher score should also correspond to a higher localization accuracy.

As the key parameter of NMS, IoU has received considerable attentions. In STD-NMS, the bounding box M with the maximum score is selected and all other bounding boxes with a significant overlap (higher than a pre-defined IOU threshold) with M are suppressed. As a result, the object, which has a significant overlap with others, is very likely to be missed. To address this issue, Bodla et al. [1] proposed a Soft NMS method, which decays the detection scores of all other objects as a continuous function of their overlap with bounding box M. Moreover, Jiang et al. [19] found that classification confidence is not always strongly related to location. So they proposed an IoU-Guided NMS, in which a predicted localization confidence replaces the classification confidence as the ranking keyword for bounding boxes. Due to standard IoU loss only works when there is overlap between bounding boxes, and would not provide any moving gradient for non-overlapping cases. To solve this problem, the generalized IoU loss (GIoU) is proposed by adding a penalty term to standard IoU. In addition, bounding box regression usually suffered from slow convergence and inaccurate regression. To address this issue, Zheng et al. [40] proposed a Distance-IoU loss by incorporating the normalized distance between the predicted box and the target box, which converges much faster in training than IoU and GIoU losses. He et al. [15] introduced the idea of coordinate weighted average steadily to guide the regression. Liu et al. [27] proposed Adaptive NMS in which different IoU thresholds were set adaptively according to the density of the detection boxes.

All these methods aim to improve IoU, however they are still not suitable for defect detection. Thus, we proposed CoP, which will be discussed in detail in the following section, so as to better guide the bounding box regression.

3. Method

Wafer defects have their unique characteristics. First of all, wafer defects, even of the same category, vary considerably in their forms and sizes, as demonstrated in Fig. 9. Second, there is little overlap between any two defects, even if they are of the same category. Third, some defects have a higher weight than others, and can only appear at the same time with certain types of defects. Considering these characteristics, our detector is based on a two-stage, proposal-driven mechanism, and traditional two-stage detection networks can all be taken as the framework, such as Faster R-CNN, Mask R-CNN, and Cascade R-CNN.

Within our two-stage detector, a deep neural network is utilized to generate cluttered defect proposals, which results in a large number of duplicate bounding boxes. And then, a bounding box regression module, which is acted by the CR-NMS, is used to remove the duplicate bounding boxes so as to obtain the real localization of the defect. The architecture of our network is shown in Fig. 1.

3.1. Correlation between bounding boxes

As mentioned before, wafer defects have their unique characteristics, which leads to the correlation between the bounding boxes. As any two wafer surface defects are hardly intersected, this means that their corresponding bounding boxes should be separate and independent. As a result, if the overlap between two bounding boxes exceeds a preset threshold, we could conclude that there must be a duplicate bounding box among them. Furthermore, among all the possible wafer defects, some of them may be more important than others. This is obviously different from the generic object detection, where all the objects should be treated equally. Faced with this situation, during bounding boxes regression, more attentions should be paid to the ones with high importance. In addition, the fact that defect categories appear in clusters is also helpful for the removal of duplicate bounding boxes.

To sum up, due to the unique nature of wafer defects, the bounding box regression methods designed for generic object detection can not be directly applied to surface defect detection, it requires a new approach. In fact, the straightforward use may lead to lots of incorrect detection results, as demonstrated in Fig. 7.

3.2. CoP

To indicate the coverage proportion of the overlap, we use CoP as an indicator. Suppose that there are two boxes, $Bbox_i$ and $Bbox_j$, then CoP can be calculated by Eq. (1).

$$CoP(Bbox_i, Bbox_j) = \frac{|Bbox_i \bigcap Bbox_j|}{X} \tag{1}$$

where $|Bbox_i \bigcap Bbox_j|$ denotes the area of overlap between $Bbox_i$ and $Bbox_j$. The variable X represents the area of $Bbox_i$ or $Bbox_j$, and which one to use is determined by the needs of bounding box regression.

In some cases, CoP can better indicate the relative position between two bounding boxes than IoU. In generic object detection, IoU is used to indicate the intersection-over-union of two bounding boxes. However, if the sizes of the two boxes are too different, it will be difficult for the IoU to judge whether they are separated, overlapped or one completely covers the other. As an example, there are two bounding boxes, $Bbox_1$ and $Bbox_2$, of different sizes, which is demonstrated in Fig. 2. They are separated, overlapped and one completely covers the other, as shown in Fig. 2a, Fig. 2b, Fig. 2c, respectively. If the size of the two boxes is too different, such as the size of $Bbox_1$ is much larger than that of $Bbox_2$, then in these three cases, the values of IoU will be equal to or very close to 0, and the IoU will fail. Correspondingly, the values of CoP are

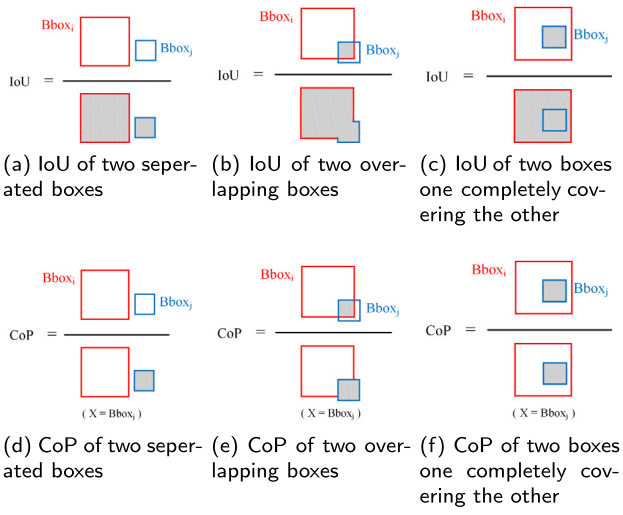


Fig. 2. Comparison of the calculation of IoU and CoP for the two boxes, $Bbox_1$ and $Bbox_2$, in three cases, where (a) (d), (b) (e) and (c) (f) correspond to the separation, overlap and coverage of the two boxes, respectively.

between 0 and 1, as shown in Fig. 2d, Fig. 2e, Fig. 2f, respectively, which can well overcome the shortage of IoU. As a result, we can distinguish them.

3.3. CR-NMS

CR-NMS utilizes CoP of any pair of bounding boxes and their corresponding category confidences (that is the score value) to guide the regression of bounding box. The whole procedure of CR-NMS is demonstrated in Algorithm 1.

Algorithm 1 CR-NMS.

Input: Bounding boxes;
Output: Bounding boxes with accurate locations and correct category labels;
1: Sort bounding boxes by score in descending order;
2: Calculate CoPs;
3: Remove duplicate bounding boxes;
4: Sort bounding boxes by area in ascending order;
5: Calculate CoPs;
6: Remove duplicate bounding boxes;

As shown in Algorithm 1, CR-NMS can be divided into two stages, coarse regression and fine regression. The coarse regression is guided by the score and CoP of the bounding boxes, so as to make a rough judgment on the location and category of the possible defect. At the same time, the bounding boxes, which with a low category confidence and high overlap, will be removed. In contrast, the fine regression is guided by the area and CoP of the remained bounding boxes, so as to make a further judgment on the uncertain defects.

3.3.1. Coarse regression

The procedure of coarse regression is described in Algorithm 2.

In this stage, all bounding boxes are sorted by their score values in descending order, firstly. Next, the bounding box with the maximum score is taken as $Bbox_i$ and the others serve as $Bbox_j$ and X in turn. And then, the CoPs are calculated according to Eq. (1). At last, the bounding boxes whose CoP values over a given threshold T_c will be removed. The whole process iterates several times until the last bounding box.

3.3.2. Fine regression

As described in Algorithm 3, this stage is used to further remove the duplicate bounding boxes derived from the coarse regression. Unlike coarse regression, all bounding boxes in this stage

Algorithm 2 Stage 1: Coarse regression.

Input: $B = \{b_1, \ldots, b_N\}$, $S = \{s_1, \ldots, s_N\}$, T_c;
B: a set of predicted detection boxes;
S: the set of box category confidences;
T_c: the CoP threshold;
Output: The remained bounding boxes;
1: $D = \phi$;
2: Sort B by score in descending order;
3: **while** $B = \phi$ **do**
4: Move the top box b_{top} from B to D;
5: **for** $b_i \in B$ **do**
6: **if** $CoP(b_{top}, b_i) \geq T_c$ **then**
7: Delete b_i from B;
8: **end if**
9: **end for**
10: **end while**
11: **return D**

are sorted in ascending order by area rather than score. And then, the bounding box with the smallest area is taken as $Bbox_i$, and the others are taken as $Bbox_j$ in turn. Thus, CoPs can be calculated according to Eq. (1). At last, guided by the value of CoP and the category information of the bounding boxes, the duplicate ones could be identified and removed.

Algorithm 3 Stage 2: Fine regression.

Input: $D = \{d_1, \ldots, d_N\}$, $S = \{s_1, \ldots, s_N\}$, $A = \{a_1, \ldots, a_N\}$, $L = \{l_1, \ldots, l_N\}$, T_f, T_e;
D: a set of predicted detection boxes;
S: the set of box category confidences;
A: the set of box areas;
L: the set of box category labels;
T_f: the CoP threshold;
T_e: the threshold of function f;
Output: The predicted detection boxes;
1: $N = \phi$;
2: Sort D by area in ascending order;
3: **while** $D = \phi$ **do**
4: $M = \phi$;
5: Take the top box as d_{top};
6: **for** $d_i \in D$ **do**
7: **if** $CoP(d_{top}, d_i) \geq T_f$ **then**
8: Copy d_i to M;
9: **end if**
10: **end for**
11: **if** $M = \phi$ **then**
12: Move d_{top} from D to N;
13: **else**
14: Take the smallest box in M as d_{sml};
15: **if** $f(s_{top}, s_{sml}) > T_e$ **then**
16: Delete the box with lower score between d_{top} and d_{sml} from D;
17: **else if** $l_{top} = l_{sml}$ **then**
18: Fuse d_{top} and d_{sml} into a new box d_{new};
19: Add d_{new} to D;
20: Delete the box with lower score between d_{top} and d_{sml} from D;
21: **else**
22: Move d_{top} and d_{sml} from D to N;
23: **end if**
24: **end if**
25: **end while**
26: **return N**

It's important to note that parameter X of CoP denotes the bounding box with smaller area among $Bbox_i$ and $Bbox_j$, and the CoP here indicates the extent to which the smaller bounding box is covered by the larger one. Thus, if the CoP is higher than the given threshold T_f, the smaller bounding box would be considered to be completely covered by the larger one. In this case, both the score values and category labels of the two bounding boxes should be taken into account. If the two bounding boxes have the same category label and similar category confidence, they should be merged to form a new bounding box. Otherwise, for the two bounding boxes with different category label but similar category

Table 1
Distribution of each category in WDD.

Category	Foreign	Gold	Incomp	Lump	Res	Scratch	UBM	Raw I	Raw II	Raw III	Raw IV	Total
Training Set	550	550	550	550	500	380	505	380	505	590	340	5400
Testing Set	61	61	61	61	56	42	56	42	56	66	38	600

Table 2
Distribution of each category in NEU-DET.

Category	Crazing	Inclusion	Patch	Pitted surface	Rolled in scale	Scratch	Total
Training Set	218	213	209	211	197	212	1260
Testing Set	82	87	91	89	103	88	540

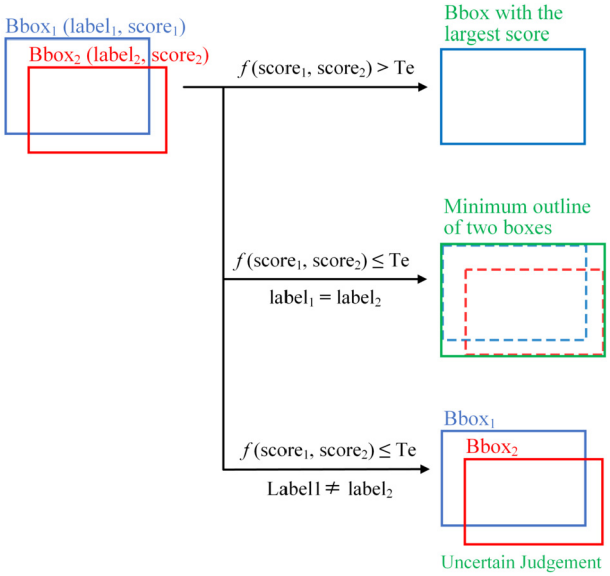


Fig. 3. The remove of duplicated bounding boxes. If the gap between the category confidence is too large, keep the box with the highest confidence without considering the category of the defect. If the two boxes have similar category confidence, and the category labels are same, merge their outlines into a new one (green box), otherwise it is difficult to judge (uncertainty). (For interpretation of the colors in the figures, the reader is referred to the web version of this article.)

confidence, they will be kept as uncertainty. In practice, the similarity of two bounding boxes could be measured by Eq. (2),

$$f(s_i, s_j) = e^{|s_i - s_j|} \tag{2}$$

where S_i and S_j are the category confidence of $Bbox_i$ and $Bbox_j$. And then, a given threshold T_e is used to guide the fine regression. The fusion process of two bounding boxes is demonstrated in Fig. 3.

In the other case, the category label will be ignored when the category confidence gaps of the two bounding boxes are larger than threshold T_e, and the one with lower category confidence will be removed directly.

4. Experiments

To evaluate the performance of our method, we constructed a surface defect detector on the generic object detection frameworks, Faster R-CNN [33], RetinaNet [25], Cascade R-CNN [2], YOLOF [10], YOLOX [6], and Mask R-CNN [12] respectively. In addition, within the detector STD-NMS [33], Soft NMS [1] and CR-NMS are employed to regress the bounding boxes, respectively. We implemented our defect detectors using MMDetection [5] framework on a workstation with a Nvidia 2080Ti GPU. Resnet50 and Resnet101, both with FPN [24], are used as the backbones, which are pretrained on the MS-COCO dataset [26].

4.1. Surface defect datasets

Two datasets, wafer defect dataset (WDD) and steel strip defect dataset (named as NEU-DET) [14], were used for training and evaluating our detector. WDD includes 6,000 images (with 512×512) labeled for 11 types of defects. These types are foreign material, gold particle, incomplete bump, scratch on bump, lump or nodule, residue, under bump metallization, and four types of raw material. And for simplicity, during the rest of this article they are denoted by Foreign, Gold, Incomp, Scratch, Lump, Res, UBM, Raw I, Raw II, Raw III, and Raw IV accordingly. In WDD, the number of samples used for training and testing are 5400 and 600, respectively, and the distributions of samples over the 11 defect categories are shown in Table 1. In each category, the typical forms and shapes of the defects are demonstrated in Fig. 9. As for dataset NEU-DET, it includes 1,800 gray images (with 200×200), labeled for 6 types of hot-rolled steel strips surface defects. They are rolled in scale, patch, cracking, pitted surface, inclusion and scratch, respectively. Similarly, in NEU-DET the samples were divided into two parts, 1500 and 300, used for training and testing, respectively. The distributions of samples over the 6 defect categories are shown in Table 2. And in each category, the typical forms and shapes of the defects are demonstrated in Fig. 10.

4.2. Evaluation metrics

Unlike generic object detectors, which mainly focus on the generated bounding boxes, that is, the accuracy of localization and the precision of classification, the main concern of a defect detector is the defects, that is, have they been found and correctly identified. Moreover, as mentioned before defects cannot be treated equally, as traditional object detectors do. Among all the possible defects, we tend to pay more attention to only a few of them. Therefore, the evaluation method developed for generic object detection will be unsuitable for defect detection, and new methods are needed.

To evaluate the performance of a wafer defect detector, we proposed four indicators. For any defect, there are four possibilities for the judgment made by the detector, correctness, wrong, uncertainty, and omission. A correct or wrong judgment means that there is only one bounding box, which correctly covered the ground truth (CoP value above the threshold), but its type label is correct or wrong. And a uncertain judgment denotes that detector cannot make an accurate judgment on the type of defects, therefore, it makes multiple judgments, including the correct judgment. In other words, there are multiple bounding boxes, which correctly covered the ground truth, and at least one of their corresponding labels is correct. In addition, omission denotes that detector omits the defect and makes no judgment on it. With regard to the bounding boxes, which fail to locate defects correctly (CoP value below the threshold), they are considered to be caused by the detector's wrong judgment of the background.

Table 3
Detection results on dataset WDD.

Method	Backbone	STD-NMS	Soft NMS	CR-NMS	*mAA*	*mAE*	*mAU*	*mAO*	*AP*	*AR*	*Bbox*	FPS	Time (μs)
Faster R-CNN	ResNet-50	√			88.9	3.7	3.2	4.2	93.1	93.9	1200	22.7	937.2
	ResNet-50		√		89.1	3.7	3.2	4.0	93.1	94.1	4001	24.0	816.2
	ResNet-50			√	92.2	3.8	0.1	3.8	96.1	92.3	847	21.4	1016.1
	ResNet-101	√			88.9	3.3	3.5	4.3	93.6	94.0	1092	20.5	963.9
	ResNet-101		√		89.0	3.3	3.5	4.2	93.6	94.0	3628	21.3	827.8
	ResNet-101			√	**92.3**	**3.3**	**0.1**	**4.2**	**96.7**	**92.3**	**811**	19.6	1065.7
RetinaNet	ResNet-50	√			76.1	5.0	15.2	3.7	78.7	94.5	6810	32.9	907.4
	ResNet-50		√		76.1	5.0	15.2	3.7	78.7	94.5	18859	33.3	861.6
	ResNet-50			√	90.8	4.9	0.1	4.2	95.1	90.9	1481	32.6	994.2
	ResNet-101	√			82.5	4.1	9.5	4.0	85.5	94.2	4799	26.0	939.5
	ResNet-101		√		82.5	4.1	9.5	4.0	85.5	94.2	13753	26.1	919.7
	ResNet-101			√	**91.8**	**4.1**	**0.1**	**4.0**	**95.8**	**91.9**	**1101**	25.8	986.4
Cascade R-CNN	ResNet-50	√			88.9	3.3	2.6	5.2	93.5	92.3	1055	24.6	865.4
	ResNet-50		√		89.0	3.3	2.6	5.1	93.5	92.5	2415	26.8	788.5
	ResNet-50			√	91.6	3.3	0.1	5.0	96.5	91.7	820	17.2	973.7
	ResNet-101	√			89.7	3.7	2.4	4.2	94.0	93.9	1030	20.9	890.2
	ResNet-101		√		89.7	3.7	2.4	4.2	94.0	93.9	2326	21.7	794.8
	ResNet-101			√	**91.9**	**3.7**	**0.1**	**4.3**	**96.5**	**92.2**	**816**	16.0	961.2
YOLOF	ResNet-50	√			89.1	3.8	2.6	4.5	93.9	93.6	2984	47.6	684.9
	ResNet-50		√		89.1	3.8	2.6	4.5	93.9	93.6	8713	47.8	634.1
	ResNet-50			√	**91.3**	**4.0**	**0.1**	**4.6**	**95.9**	**91.4**	**1106**	47.0	761.1
YOLOX	CSPDarknet	√			85.3	6.6	6.4	1.7	88.4	94.9	5826	31.8	666.2
	CSPDarknet		√		85.3	6.6	6.4	1.7	88.4	94.9	13286	32.4	619.1
	CSPDarknet			√	**90.8**	**6.5**	**0.1**	**2.6**	**93.9**	**90.9**	**2499**	31.0	831.0

In our work, there are four metrics used to evaluate the performance of the detector for a certain type of defects. They are average accuracy (AA), average error (AE), average uncertainty (AU), and average omission (AO), respectively, and can be calculated as Eq. (3).

$$AA = \frac{N_c}{N_{all}}, \quad AE = \frac{N_w}{N_{all}}, \quad AU = \frac{N_u}{N_{all}}, \quad AO = \frac{N_o}{N_{all}}$$
$$N_{all} = N_c + N_w + N_u + N_o \tag{3}$$

where N_c, N_w, N_u, and N_o represent the number of samples of a certain type which are judged correctly, wrongly, uncertainly, and are omitted, respectively. And N_{all} represents the total number of samples of this type.

Furthermore, in order to evaluate the comprehensive performance of the detector for all types, we used mean AA (mAA), mean AE (mAE), mean AU (mAU), and mean AO (mAO), which respectively denotes the mean average of accuracy, error, uncertainty, and omission of all types, and can be calculated by Eq. (4).

$$mAA = \sum_{i=1}^{C} \alpha_i AA_i, \qquad mAE = \sum_{i=1}^{C} \alpha_i AE_i,$$
$$mAU = \sum_{i=1}^{C} \alpha_i AU_i, \qquad mAO = \sum_{i=1}^{C} \alpha_i AO_i \tag{4}$$

where C denotes the total number of types and α_i denotes the weight of type i, that is, the proportion of type i samples in all types of samples.

4.3. Defect detection

To evaluate the performance of our method, we carried out the defect detection experiments on both WDD and NEU-DET. Conventionally, the two datasets were all divided into two parts, training set and testing set, and the number of samples for each category were shown in Table 1 and Table 2. We implemented five defect detectors, which used Faster R-CNN [33], RetinaNet [25], Cascade R-CNN [2], YOLOX [6], and YOLOF [10] to generate bounding boxes, respectively, and used CR-NMS for the bounding box regression. For comparison, we replaced CR-NMS with STD-NMS and Soft-NMS respectively for bounding box regression. The results of the experiments on WDD and NEU-DET were shown in Table 3 and Table 4, respectively.

4.3.1. Overall accuracy

The overall accuracy of the defect detectors were evaluated by the parameters mAA, mAE, mAU and mAO, and the results were shown in Table 3, from which we can see at least three points. First of all, among the three NMS methods, CR-NMS helped the detector achieve the highest detection accuracy. With backbone ResNet-50, compared with STD-NMS (or Soft-NMS), CR-NMS helped Faster R-CNN, RetinaNet, and Cascade R-CNN improve their mAA by 3.4%, 9.3%, and 2.2%, respectively. Second, there is a significant drop in the value of mAU when the detector uses the CR-NMS, instead of STD-NMS (or Soft-NMS). For Faster R-CNN, RetinaNet, and Cascade R-CNN, the gaps of mAU between STD-NMS and CR-NMS are 3.4% (from 3.5% to 0.1%), 12.8% (from 9.5% to 0.1%), and 2.3% (from 2.4% to 0.1%), respectively. Similar results appeared between Soft-NMS and CR-NMS. These imply that by using the correlation between categories, CR-NMS effectively reduces the proportion of uncertain judgments. Furthermore, it also means that CR-NMS makes correct judgments on most of the previously uncertain defects (mAA increased obviously), and very few judgments are wrong (mAE changes very little). At last, under the same architecture, the detector using ResNet-101 has higher detection accuracy than the detector using ResNet-50. When the backbone changes from ResNet-50 to ResNet-101, the mAA of Faster R-CNN, RetinaNet, and Cascade R-CNN increase by 0.1%, 1.0%, and 0.3%, respectively. The reason lies in the fact that with a more deep neural network, ResNet-101 can better extract the characteristics of the defects.

For further performance analysis, we compared the detection results of CR-NMS with STD-NMS and Soft NMS for each type of defect. The detector is based on Faster R-CNN with backbone ResNet-101, and the results are shown in Fig. 4. Fig. 4a shows the difference in AA between CR-NMS and STD-NMS and Soft NMS for each defect category. We can see that CR-NMS has improved AA in all other categories other than category Foreign. The principal reason we think is that compared with other categories, category Foreign is more complicated (this can be seen from its obvious

Table 4
Detection results on dataset NEU-DET.

Method	Backbone	STD-NMS	Soft NMS	CR-NMS	*mAA*	*mAE*	*mAU*	*mAO*	*AP*	*AR*	*Bbox*	FPS	Time (μs)
Faster R-CNN	ResNet-50	√			94.3	1.7	1.7	2.3	96.7	97.0	5623	45.7	1321.9
	ResNet-50		√		94.2	1.7	1.9	2.2	96.5	97.2	20556	45.8	1272.7
	ResNet-50			√	**95.3**	**1.6**	**0.4**	**2.7**	**98.0**	**96.0**	**1762**	44.6	2368.5
RetinaNet	ResNet-50	√			70.3	4.0	12.1	13.6	86.2	85.4	3879	45.0	1075.4
	ResNet-50		√		70.4	4.0	12.1	13.5	86.2	85.4	43024	45.0	1035.7
	ResNet-50			√	**88.7**	**4.5**	**0.7**	**6.1**	**94.9**	**91.2**	**2415**	43.2	1303.5
Cascade R-CNN	ResNet-50	√			94.0	1.2	1.4	3.4	97.7	96.1	4028	35.8	830.2
	ResNet-50		√		94.0	1.2	1.4	3.4	97.7	96.2	13559	36.0	815.8
	ResNet-50			√	**95.2**	**0.9**	**0.2**	**3.7**	**98.8**	**95.4**	**1250**	34.0	1139.6
YOLOF	ResNet-50	√			67.8	7.0	15.7	9.5	78.8	86.2	48322	44.6	1773.1
	ResNet-50		√		67.9	7.0	15.6	9.5	78.9	86.2	53725	45.1	1646.7
	ResNet-50			√	**84.8**	**7.5**	**5.2**	**2.5**	**87.7**	**90.8**	**2893**	43.4	3167.4
YOLOX	CSPDarknet	√			82.1	3.0	1.5	13.4	95.4	84.4	7475	62.7	666.2
	CSPDarknet		√		82.2	3.0	1.4	13.4	95.5	84.4	19037	63.3	619.1
	CSPDarknet			√	**83.3**	**3.1**	**0.5**	**13.1**	**96.0**	**84.0**	**2378**	61.7	1271.0

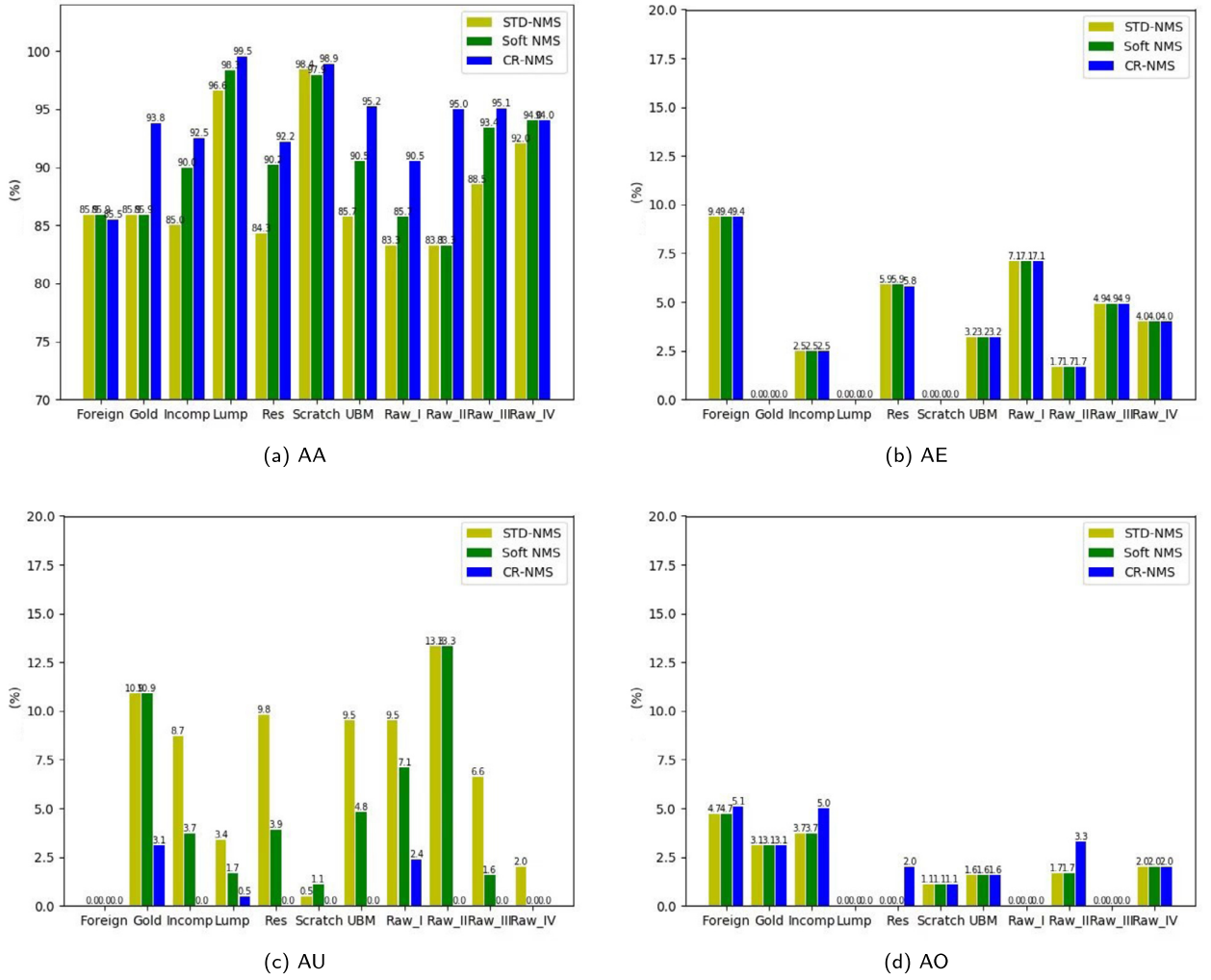


Fig. 4. Comparison of four indicators of 11 categories in WDD based on framework Faster R-CNN with backbone ResNet-101. (a) Results of average accuracy (AA). (b) Results of average error (AE). (c) Results of average uncertainty (AU). (d) Results of average omission (AO).

AE value, demonstrated in Fig. 4b), and CR-NMS fails to grasp the correlation between the categories, resulting in a slight decrease in *AA*. Fig. 4c shows that although Soft NMS handles some uncertainties, it is not satisfactory. It further verifies the fact that compared with STD-NMS and Soft NMS, CR-NMS greatly reduces the proportion of uncertain judgments in each category. From Fig. 4d, it can be seen that CR-NMS has a slight impact on omission judgment, and the reason is discussed in Section 4.5.

4.3.2. Precision and coverage

In order to evaluate the accuracy and coverage of the judgments made by the detectors, we count the AP (average precision) and

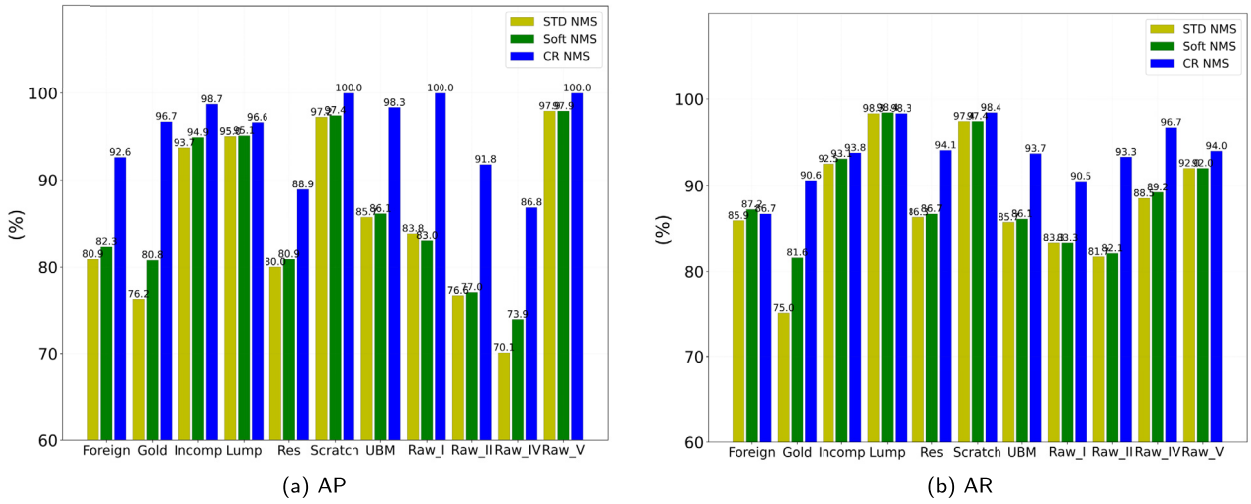


Fig. 5. Comparison of parameters AP and AR of 11 categories in WDD based on framework Faster R-CNN with backbone ResNet-101. (a) Results of average precision (AP). (b) Results of average recall (AR).

AR (average recall) of each detector, and the results are shown in Table 3. From these results, we can see that compared with STD-NMS and Soft NMS, CR-NMS can obviously improve the AP and AR of the detectors. With backbone ResNet-101, compared with STD-NMS, CR-NMS helped Faster R-CNN, RetinaNet, and Cascade R-CNN improve their AP by 3.1%, 10.3%, and 2.5%, respectively. Correspondingly, the reduced values of AR are −1.7%, −2.3%, and −1.7%, respectively. Similar results can also be seen from YOLOX and YOLOF. This implies that in the contradiction between AP and AR, CR-NMS pays more attention to AP, which trades a slight decrease in AR for a significant increase in AP. It is reasonable for defect detection to improve the accuracy of judgment as much as possible while ensuring a certain coverage.

To evaluate the effect of STD-NMS, Soft NMS and CR-NMS on AP and AR for each type of defect, we built a detector based on Faster R-CNN with backbone ResNet-101, and the results are shown in Fig. 5. Fig. 5a shows that CR-NMS can significantly improve the AP of the detector. For some categories, such as Raw_V, Raw_I and Scratch, the detector can even identify them correctly (the value of AP is 100%). As can be seen from Fig. 5b, except the AR of Foreign dropped obviously, the AR of other categories remained unchanged or increased slightly. The reason lies in the fact that Foreign represents foreign materials, and their features are relatively complex, which make it difficult for the detector to correctly identify them.

4.3.3. Complexity

Generally, under the same operating conditions the computational complexity of an algorithm is proportional to its running time, and the higher the complexity, the longer its running time. Therefore, we take the running times of STD-NMS, Soft NMS and CR-NMS as their complexity indicators, respectively. The detailed results are shown in Table 3 and Table 4. From the two tables, it can be seen that the running time of CR-NMS is significantly increased compared to that of STD-NMS and Soft NMS, which means that the CR-NMS has much higher computational complexity than the other two. The main reason lies in the fact that CR-NMS removed much more redundant boxes than the other two methods, in other words, it retained the least number of Bboxes. When STD-NMS is replaced by CR-NMS, the time consumption of Faster R-CNN, RetinaNet, and Cascade R-CNN for bounding boxes regression is increased by 78.9 μs (from 937.2 μs to 1016.1 μs), 86.8 μs (from 907.4 μs to 994.2 μs), and 108.3 μs (from 865.4 μs to 973.7 μs), accounting for 8.4%, 9.6%, and 12.5%, respectively. As for YOLOF and YOLOX, when CR-NMS replaces STD-NMS, their time consumption increases by 76.2 μs (from 684.9 μs to 761.1 μs) and 164.9 μs (from 666.1 μs to 831.0 μs), accounting for 11.1% and 24.8%, respectively. This implies that the CR-NMS is more efficient than STD-NMS and Soft NMS in bounding box regression and leads to more time consumption (computation complexity). However, the increase in complexity of CR-NMS is relatively small and has little effect on the overall complexity of the detector, and its FPS only slightly drops. When CR-NMS replaces Soft NMS, we can draw similar conclusions. However, Soft NMS retains more Bboxes, and the complexity of CR-NMS will increase much more.

In addition, Table 4 demonstrated the results of the detectors on dataset NEU-DET. From this table, we can also see that compared with STD-NMS and Soft NMS, CR-NMS significantly reduces the number of Bboxes, but at the same time, its time consumption has increased obviously.

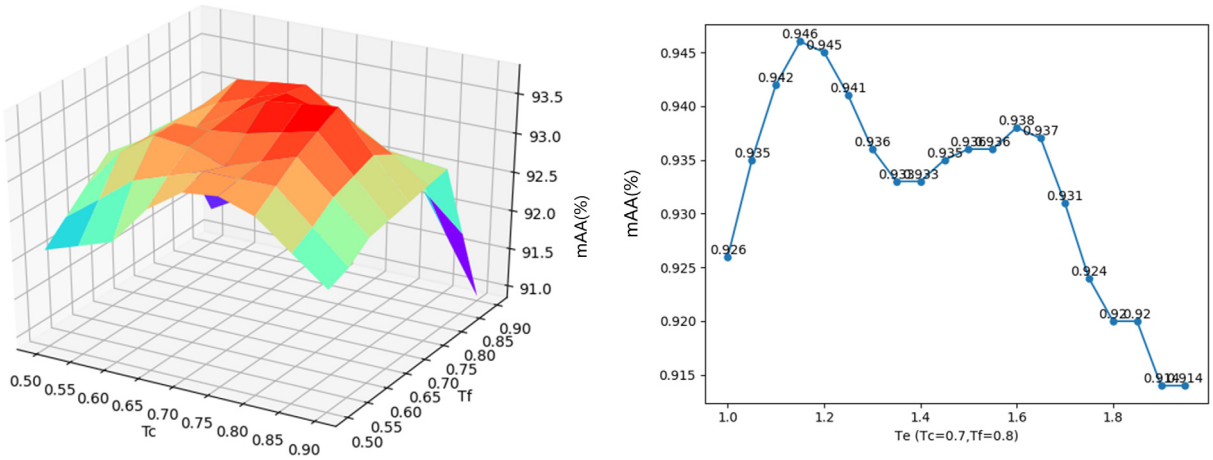
4.4. Thresholds of CR-NMS

In CR-NMS, there are three thresholds, T_e, T_c and T_f, where T_e is used to measure the difference of confidence between two bounding boxes, T_c and T_f are used to guide the coarse and fine bounding box regression, respectively. In our work, we evaluated the impacts of T_c, T_f and T_e on mAA employing detector Faster R-CNN with dataset WDD.

The curves of the mAA at different T_e, T_c and T_f values are shown in Fig. 6. Fig. 6a shows the surface of mAA at different T_c and T_f, where the T_e is fixed to 1.0, so as to ignore its influence on mAA. From this figure, we can see that T_c and T_f both have a significant impact on mAA. In addition, when T_f is fixed, as T_c gradually increases, mAA will reach its maximum value. Similarly, when T_c is fixed, mAA will also reach its maximum value with the gradual increase of T_c. And when T_c is equal to 0.7 and T_f is equal to 0.8, mAA will reach the global maximum. Fig. 6b shows the curve of mAA at different T_e, where T_c and T_f were fixed to 0.7 and 0.8. It can be seen that if T_e is between 1.15 and 1.2, the mAA reaches its maximum. This denotes that if the confidence difference between two bounding boxes is less than 0.18, calculated by Eq. (2), they are considered to be similar enough in confidence. In general, when $T_c = 0.7$, $T_f = 0.8$, and $T_e = 1.15$, CR-NMS reaches its optimal performance.

4.5. Ablation study

For a better understanding of CR-NMS, we investigate the effects of coarse regression and fine regression on detection accuracy. As in the previous section, WDD is selected as the test dataset



(a) Trend of the mAA with different T_c and T_f values. (b) Trend of the mAA with different T_e values.

Fig. 6. The mAA of CR-NMS at different T_c, T_f, and T_e, where the defect detector is built on framework Faster R-CNN with backbone ResNet-101. (a) shows the variation curve of mAA with T_c and T_f while the threshold T_e is set to 1.0 temporarily, and its influence is ignored. (b) shows the variation curve of mAA with different T_e values while the threshold T_c and T_f are set to 0.7 and 0.8 respectively, the optimal values of T_c and T_f can be seen from (a).

Table 5
The effects of each stage in CR-NMS.

NMS	Coarse	Fine	mAA	mAE	mAU	mAO
STD	-	-	88.9	3.3	3.5	4.3
Soft	–	–	89.0	3.3	3.5	4.2
CR	√		90.7	2.9	4.4	2.0
		√	91.2	2.8	2.7	3.3
	√	√	**92.3**	**3.3**	**0.1**	**4.2**

and the detector is also based on Faster R-CNN with backbone ResNet-101. And the coarse regression and fine regression contained in CR-NMS are activated sequentially.

Results are shown in Table 5. We can see that even if coarse regression or fine regression is used alone, it can well regress the bounding box (mAA is 90.7% and 91.2%, respectively) and achieve performance comparable to STD-NMS (mAA is 88.9%) and Soft-NMS (mAA is 89.0%). Besides, when coarse regression is used alone, the detector will have a considerable percentage of uncertain judgments (mAU is 4.4%). On the contrary, when fine regression is used alone, although the proportion of uncertain judgments is significantly reduced (mAU reduced from 4.4% to 2.7%), the proportion of omission has been obviously increased (mAO increased from 2.0% to 3.3%). This implies that Fine Regression can remove the uncertain judgments effectively, but at the same time it also made a small amount of misjudgments, resulting in a slight increase of omissions. Moreover, when both the coarse and fine regression are enabled, mAA reaches the maximum (92.3%), and mAU and mAO also reach a compromise value, 0.1% and 4.2%, respectively.

4.6. Comparative analysis of detection results

As mentioned in the previous section, CR-NMS improves the performance of the detector by using inter-category correlation to convert the uncertain judgment into a certain judgment. Fig. 7 shows seven typical examples. In this figure, the first row provides seven images of different types of defects. And the seven images in the second, third, and fourth rows correspond to the detection results of STD-NMS, Soft NMS, and CR-NMS, respectively.

From Fig. 7, we can see at least two points. First, for some complex defects, STD-NMS retains two or more bounding boxes, that is, it makes an uncertain judgment, such as the red boxes at the 4th, 5th, and 6th columns of the second row. And although Soft NMS handles part of some uncertainty cases, it does not sufficiently remove redundant boxes, such as the red boxes at the first to third columns of the third row. In contrast, CR-NMS removes those wrong or redundant bounding boxes and retains the only correct one, that is, make a correct judgment. Second, the appearance of redundant bounding boxes with similar confidence (images in the second row) indicates that the features extracted by the backbone are not enough to accurately distinguish each defect category, and further training or improvement of the neural network is required.

4.7. Defect segmentation

Besides the location and category of defects, we also often want to know their corresponding geometric properties, such as length, width, area, contour, center, etc. At this time, segmentation of defects is essential. Mask R-CNN is one of the most commonly used image instance segmentation methods at present, and can be regarded as a multi-task learning method based on the combination of detection and segmentation networks. In order to evaluate the effect of CR-NMS in defect segmentation, we tested Mask R-CNN with STD-NMS, Soft NMS, and CR-NMS on WDD. The results are demonstrated in Table 6.

From this table, we can see at least two points. First of all, it is further confirmed that under the same network architecture, CR-NMS has higher detection accuracy than STD-NMS and Soft NMS, and the improvement comes from CR-NMS's processing of uncertain judgments. Second, compared with bounding box, CR-NMS has higher accuracy in segmentation. When using ResNet-101 and CR-NMS, the mAA values of Mask R-CNN are 92.0% and 92.5% under the detection task and segmentation task, respectively. This means that when the pixel-to-pixel is adopted, CR-NMS can make more effective use of the correlation between categories. The results of instance segmentation are shown at the last row in Fig. 8.

5. Conclusions

In this work, we proposed a two-stage bounding box regression method, CR-NMS, by which we can transform a generic object detector into a defect detector. In order to evaluate the performance of CR-NMS, we constructed four defect detectors base on Faster R-CNN, RetinaNet, Cascade R-CNN and Mask R-CNN respectively, and trained and tested them on the WDD and public NEU-DET dataset.

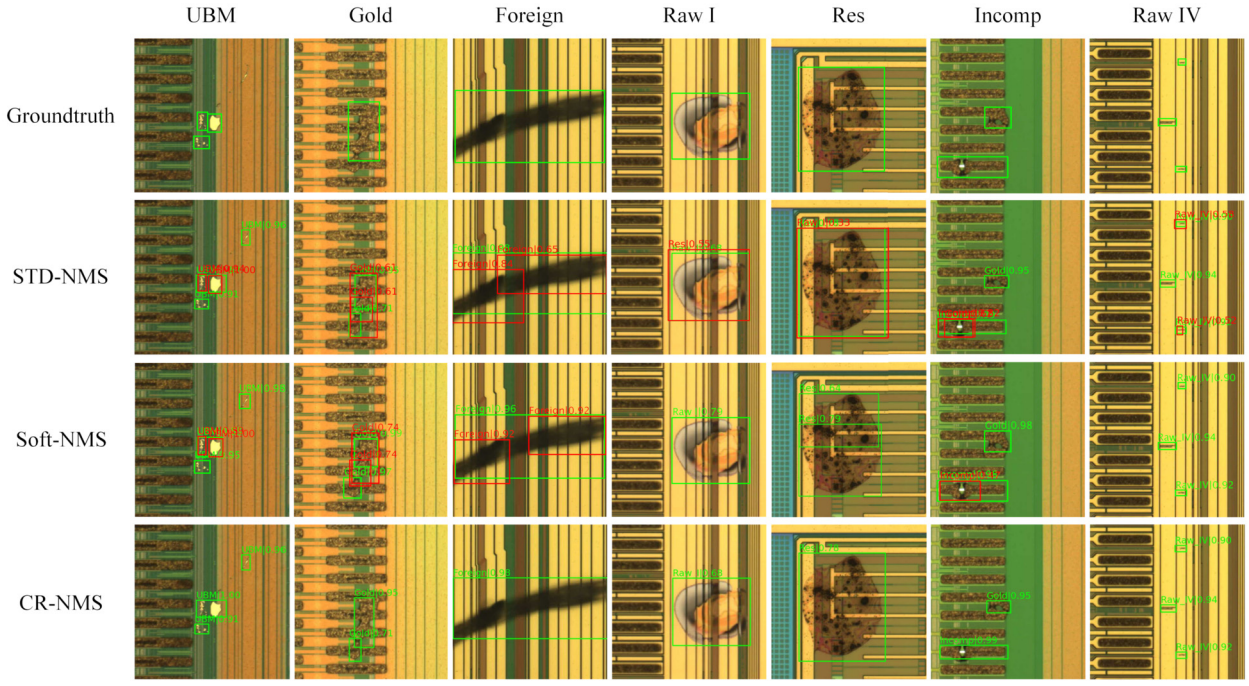


Fig. 7. Comparison of the detection results. The first row shows seven images in different categories with ground truth bounding boxes. The second to fourth rows show the detection results of the images in the first row under STD-NMS, Soft NMS, and CR-NMS, respectively.

Table 6
Detection results of Mask R-CNN on WDD.

Method	Backbone	STD-NMS	Soft NMS	CR-NMS	*mAA*		*mAE*		*mAU*		*mAO*		*Bbox*
					bbox	*segm*	*bbox*	*segm*	*bbox*	*segm*	*bbox*	*segm*	
Mask R-CNN [12]	ResNet-50	√			82.1	82.5	3.5	3.4	11.5	11.3	2.8	2.8	1552
	ResNet-50		√		85.1	85.0	3.8	3.6	8.0	8.0	3.1	3.4	4224
	ResNet-50			√	91.3	91.5	3.6	3.6	0.6	0.5	4.4	4.4	880
	ResNet-101	√			84.9	85.3	3.1	3.1	9.3	9.1	2.7	2.6	1408
	ResNet-101		√		86.5	86.5	3.6	3.6	7.1	7.1	2.7	2.7	3836
	ResNet-101			√	**92.0**	**92.5**	**3.6**	**3.6**	**0.8**	**0.5**	**3.6**	**3.4**	**851**

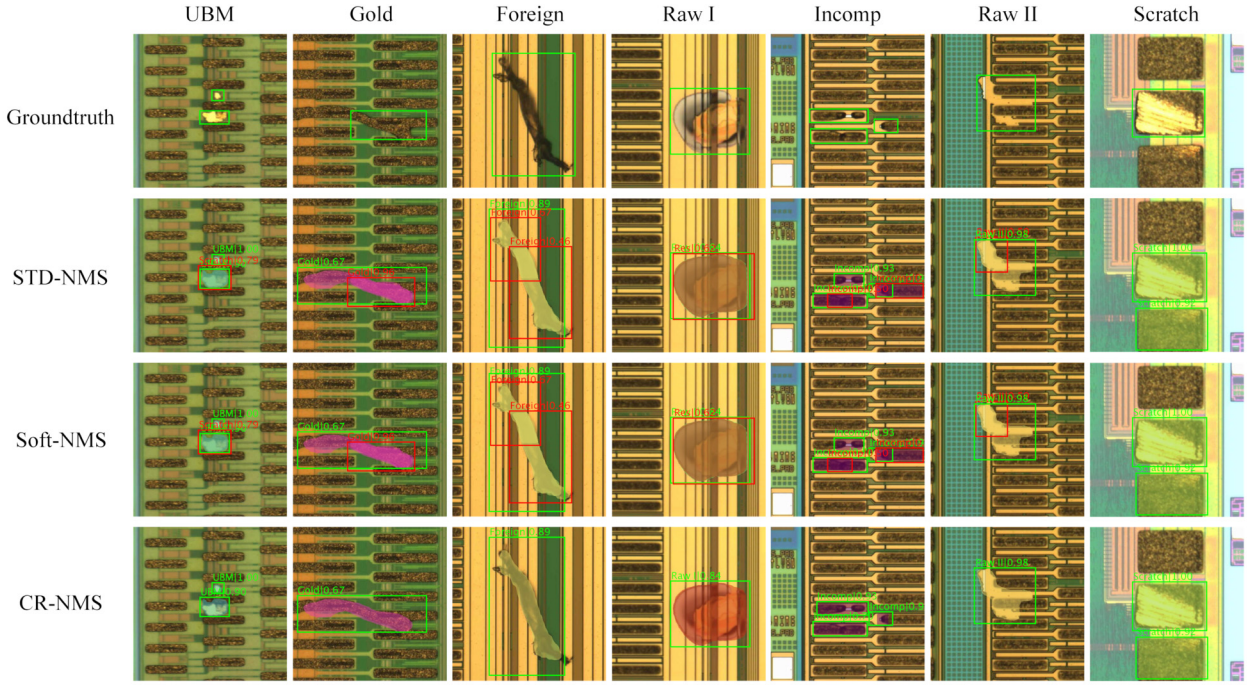


Fig. 8. Comparison of the segmentation results of the three methods in Mask R-CNN. The first row shows seven images in different categories with ground truth bounding boxes. The second to fourth rows show the segmentation results of the images in the first row under STD-NMS, Soft NMS, and CR-NMS, respectively.

The results show that by using the correlation between categories, CR-NMS can effectively suppress the redundant bounding boxes, reduce the proportion of uncertain judgments, and obtain a higher mAP than STD-NMS and Soft NMS. At the same time, a considerable proportion of wrong judgments and omission judgments in the results imply that we need to further optimize the architecture of neural network in our future work.

CRediT authorship contribution statement

Xinyu Wang: Data curation, Investigation, Methodology, Software, Visualization, Writing – original draft. **Xiaoli Jia:** Investigation, Writing – review & editing. **Chuyi Jiang:** Investigation, Writing – review & editing. **Sanxin Jiang:** Conceptualization, Investigation, Methodology, Software, Writing – review & editing.

Declaration of competing interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Data availability

The data used to support the findings of this study are available from the corresponding author upon request.

Acknowledgments

The authors would like to thank Jiangsu nepes Semiconductor Co., Ltd. for providing the experiment images and helping to label them. They would also like to thank the editors and anonymous reviewers for their help in improving this paper.

Appendix A. Typical shapes and sizes of various types of defects in the WDD and NEU-DET dataset

See Figs. 9 and 10 below.

References

[1] N. Bodla, B. Singh, R. Chellappa, L.S. Davis, Soft-nms – improving object detection with one line of code, 2017.
[2] Z. Cai, N. Vasconcelos, Cascade r-cnn: delving into high quality object detection, 2017.
[3] Y.J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, O. Buyukozturk, Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types, Comput.-Aided Civ. Infrastruct. Eng. 33 (2018) 731–747.
[4] J. Chen, Z. Liu, H. Wang, A. Núñez, Z. Han, Automatic defect detection of fasteners on the catenary support device using deep convolutional neural network, IEEE Trans. Instrum. Meas. (2017).
[5] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Mmdetection: open mmlab detection toolbox and benchmark, 2019.
[6] Q. Chen, Y. Wang, T. Yang, X. Zhang, J. Cheng, J. Sun, You only look one-level feature, CoRR, arXiv:2103.09460, 2021.
[7] T. Chen, Y. Wang, C. Xiao, Q. Wu, A machine vision apparatus and method for can-end inspection, IEEE Trans. Instrum. Meas. 65 (2016) 1–12.
[8] Y. Chen, Y. Ding, F. Zhao, E. Zhang, L. Shao, Surface defect detection methods for industrial products: a review, Appl. Sci. 11 (2021) 7657.
[9] L. Cui, X. Jiang, M. Xu, W. Li, B. Zhou, Sddnet: a fast and accurate network for surface defect detection, IEEE Trans. Instrum. Meas. 70 (2021) 2505713.
[10] Z. Ge, S. Liu, F. Wang, Z. Li, J. Sun, YOLOX: exceeding YOLO series in 2021, CoRR, arXiv:2107.08430, 2021.
[11] D. He, K. Xu, P. Zhou, D. Zhou, Surface defect classification of steels with a new semi-supervised learning method, Opt. Lasers Eng. 117 (2019) 40–48.
[12] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961–2969.
[13] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
[14] Y. He, K. Song, Q. Meng, Y. Yan, An end-to-end steel surface defect detection approach via fusing multiple hierarchical features, IEEE Trans. Instrum. Meas. 69 (4) (2020) 1493–1504.
[15] Y. He, C. Zhu, J. Wang, M. Savvides, X. Zhang, Bounding box regression with uncertainty for accurate object detection, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
[16] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: efficient convolutional neural networks for mobile vision applications, 2017.
[17] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141.
[18] G. Huang, Z. Liu, V. Laurens, K.Q. Weinberger, Densely Connected Convolutional Networks, IEEE Computer Society, 2016.
[19] B. Jiang, R. Luo, J. Mao, T. Xiao, Y. Jiang, Acquisition of localization confidence for accurate object detection, 2018.
[20] J. Jing, S. Liu, P. Li, L. Zhang, The fabric defect detection based on cie l*a*b* color space using 2-d Gabor filter, J. Text. Inst., Proc. Abstr. 107 (2016) 1305–1313.
[21] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2012) 1097–1105.
[22] F.R. Leta, F.F. Feliciano, F. Martins, Computer vision system for printed circuit board inspection.
[23] Y. Li, H. Huang, Q. Xie, L. Yao, Q. Chen, Research on a surface defect detection algorithm based on mobilenet-ssd, Appl. Sci. 8 (2018).
[24] T.Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
[25] T.Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, IEEE Trans. Pattern Anal. Mach. Intell. (2017) 2999–3007.
[26] T.Y. Lin, M. Maire, S. Belongie, J. Hays, C.L. Zitnick, Microsoft coco: common objects in context, in: European Conference on Computer Vision, 2014.
[27] S. Liu, D. Huang, Y. Wang, Adaptive NMS: Refining Pedestrian Detection in a Crowd, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 6452–6461.
[28] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, A.C. Berg, Ssd: Single Shot Multibox Detector, Springer, Cham, 2016.
[29] Z. Liu, K. Liu, J. Zhong, Z. Han, W. Zhang, A high-precision positioning approach for catenary support components with multiscale difference, IEEE Trans. Instrum. Meas. (2019).
[30] S. Putera, Z. Ibrahim, Printed circuit board defect detection using mathematical morphology and matlab image processing tools, in: International Conference on Education Technology & Computer, 2010.
[31] A. Rasheed, B. Zafar, A. Rasheed, N. Ali, M.T. Mahmood, Fabric defect detection using computer vision techniques: a comprehensive review, Math. Probl. Eng. 2020 (2020).
[32] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You Only Look Once: Unified, Real-time Object Detection, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788.
[33] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (2017) 1137–1149.
[34] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, Comput. Sci. (2014).
[35] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, A. Rabinovich, Going Deeper with Convolutions, IEEE Computer Society, 2014.
[36] X. Tao, D. Zhang, Z. Wang, X. Liu, H. Zhang, D. Xu, Detection of power line insulator defects using aerial images analyzed with convolutional neural networks, IEEE Trans. Syst. Man Cybern. Syst. 50 (2020) 1486–1498.
[37] X. Tao, Z. Zhang, F. Zhang, D. Xu, A novel and effective surface flaw inspection instrument for large-aperture optical elements, Int. J. Autom. Comput. 14 (2017) 420–431.
[38] Y. Xue, Y. Li, A fast detection method via region-based fully convolutional neural networks for shield tunnel lining defects, Comput.-Aided Civ. Infrastruct. Eng. 33 (2018).
[39] X. Zhang, X. Zhou, M. Lin, J. Sun, Shufflenet: an extremely efficient convolutional neural network for mobile devices, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6848–6856.
[40] Z. Zheng, P. Wang, W. Liu, J. Li, D. Ren, Distance-iou loss: faster and better learning for bounding box regression, in: AAAI Conference on Artificial Intelligence, 2020.
[41] J. Zhong, Z. Liu, Z. Han, H. Ye, W. Zhang, A cnn-based defect inspection method for catenary split pins in high-speed railway, IEEE Trans. Instrum. Meas. (2018).

Xinyu Wang received B.S. degree form the Northeast Electric Power University, Jilin, China.

He is currently a postgraduate student of Shanghai University of Electric Power. His research interests include image processing, machine learning, and pattern recognition.

Xiaoli Jia received the B.S. and M.S. degrees from the University of Electronic Science and Technology of China, Chengdu, China, in 2009 and 2012, respectively.

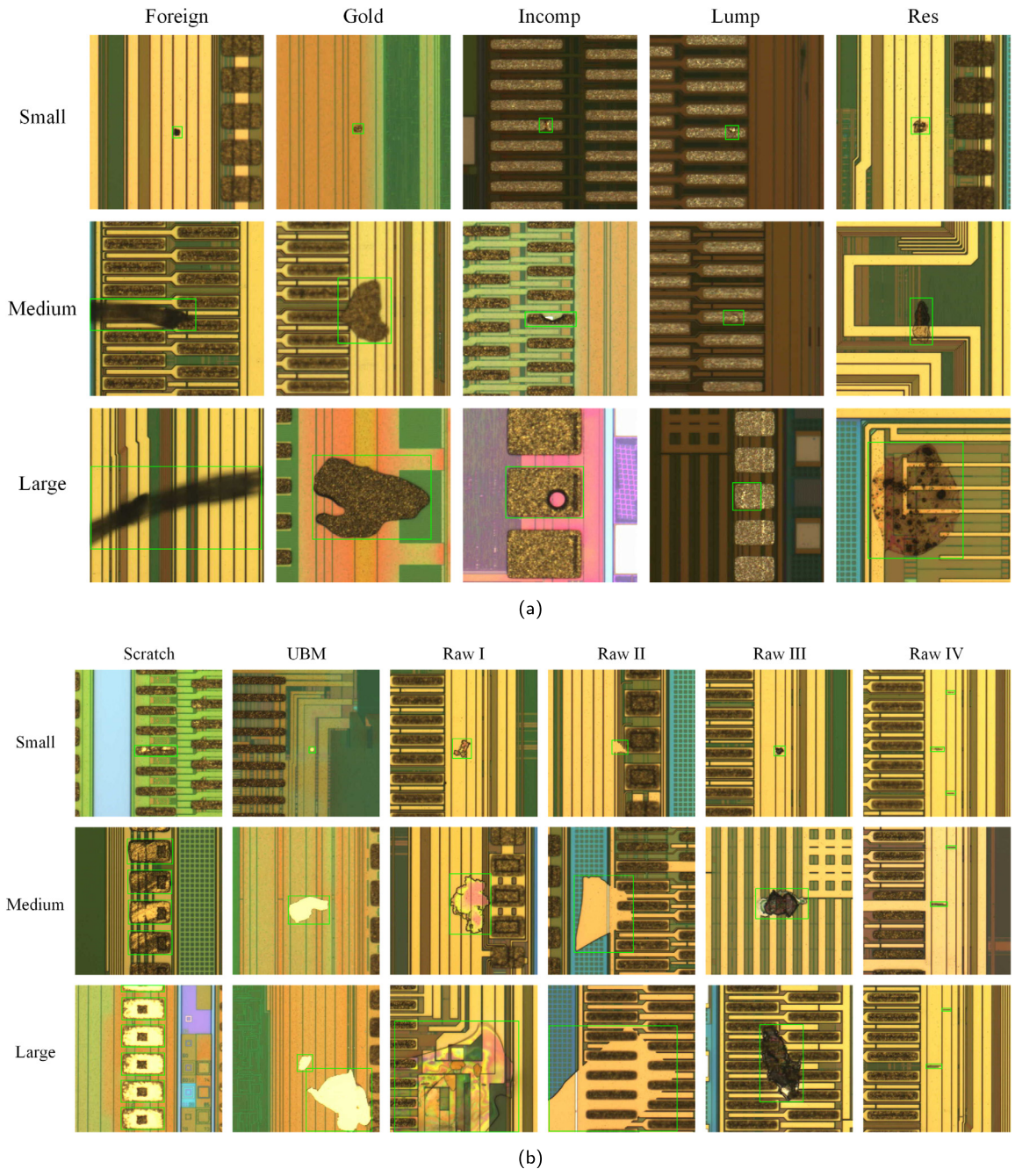


Fig. 9. Typical shapes and sizes of 11 types of defects in the WDD dataset. (a) five types of defects. (b) six types of defects.

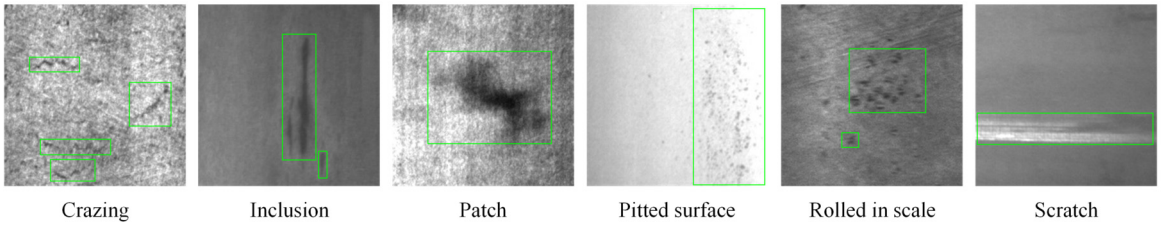


Fig. 10. Typical shapes and sizes of 6 types of defects in the NEU-DET dataset.

He is currently pursuing the Ph.D. degree in information and communication engineering at Shanghai Jiao Tong University. His research interests include compressive sensing, sound source localization, speech enhancement.

Chuyi Jiang received the B.S. degree from the University of Bristol, UK, in 2022. Now he starts his master program of Robotic and Computation at the University College London, UK.

His research interests include machine learning, computer vision, and robotic.

Sanxin Jiang received the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2015.

He is currently an Assistant Professor with the College of Electronics and Information Engineering, Shanghai University of Electric Power, Shanghai, China. His research interests include intelligent inspection, image processing, and pattern recognition.