Segmentation-based Decision Networks for Steel Surface Defect Detection

Zhongqin Bi[1], Qiancong Wu[1*], Meijing Shan[2], Wei Zhong[3]

[1] College of Computer Science and Technology, Shanghai University of Electric Power, China
[2] Institute of Information Science and Technology, East China University of Political Science and Law, China
[3] No. 34 Research Institute, China Electronics Technology Group Corporation, China
zqbi@shiep.edu.cn, wu_qiancong@163.com, shanmeijing@ecupl.edu.cn, zhongwei518@163.com

Abstract

With the advent of the Industrial 4.0 era, deep learning has been continuously applied to the task of surface defect detection, and effective progress has been made. However, the limited number of training samples and high labelling costs are considerable obstacles to the vigorous development of this task. Thus, we explore the use of different numbers of labels with various accuracies during training to achieve the maximum detection accuracy with the lowest cost. Our proposed method includes improved segmentation and decision networks. An attention mechanism is integrated into the segmentation subnetwork. Moreover, atrous convolutions are used in the segmentation and decision subnetworks. In addition, the original loss function is improved. Several experiments are carried out on the Severstal Steel Defect dataset collected in Germany, and the results show that each component improves the detection accuracy by 1% to 2%. Finally, when we add an appropriate number of pixel-level labels in the weakly supervised learning mode, the detection accuracy reaches that of the fully supervised mode with a significantly reduced annotation cost.

Keywords: Quality control, Deep-learning Industrial 4.0, Surface defect detection

1 Introduction

In the industrial product quality control process, the most intuitive judgement is whether visible surface defects, such as spots, scratches, and leaks, can be found on the surface of the finished product [1]. These surface defects often occur when drill bits, cutting tools or other parts in the production machinery have been damaged and need to be reviewed and replaced to prevent greater losses. However, the efficiency of traditional manual testing does not satisfy industrial production requirements [2]. With the development of computer technology, machine vision has gradually been applied to surface defect detection [3]. By combining nondestructive testing, automation and intelligence, machine vision can not only meet the safety and high-efficiency requirements of industrial production but also achieve high testing accuracy. However, machine vision applications encounter a substantial problem: machine vision equipment needs different image processing algorithms for various defect detection tasks, limiting the versatility of the equipment. Gao et al. focused on trust node management in VANETs, solving the problems of mutual collaboration and data communication [4]. However, high production and maintenance costs increase the difficulty of developing new machine vision applications.

In recent years, deep learning methods have been applied to surface defect detection and anomaly detection in industrial quality control [5-7]. However, in the actual industrial production process, defective samples are often difficult to generate, resulting in a limited number of positive samples in defect datasets and an incomplete collection of defect types. This critical small sample problem must be addressed for deep learning methods in surface defect detection applications in the industrial field. During testing, the number of samples is insufficient; the proportion of positive and negative samples is uneven, with the number of negative samples far exceeding the number of positive samples, as shown in Table 1. In general, normal samples are readily available in actual production processes, while few defect samples, especially those with particular defects, are available. Thus, it is impossible to learn the specific characteristics of different defect types through a large number of carefully labelled samples in supervised learning.

Defect samples face another issue during the annotation process. The defect area of the sample must be annotated very finely, and pixel-level annotations can be used to distinguish defect and defect-free areas. However, pixel-level labels are often difficult to generate. Therefore, we shift the focus of our work to ensuring the detection accuracy while minimizing the need for annotations, thus reducing the labelling costs. Some training samples and their labels in the KolektorSDD and DAGM surface defect datasets are shown in Figure 1.

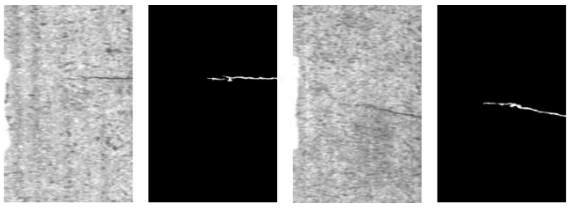
Table 1. Details of some surface defect datasets

Datasets	Positive Samples	Negative Samples	Ratio
KSDD	52	347	1: 6.7
DAGM (1-6)	450	3000	1: 6.7
DAGM (7-10)	600	4000	1: 6.7
Severstal Steel	4759	6666	1: 1.4

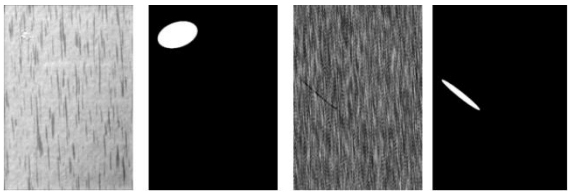
The surface defect detection task has achieved good results in full supervision mode; however, in this mode, a large amount of data need to be learned and matched with pixel-level labels through high-precision annotation. As a result, many industrial problems cannot be solved easily, or, because of the need for high-precision annotations, these problems are very expensive to solve. In unsupervised mode [8-10] and weakly supervised mode [11-13], although the cost of high-

*Corresponding Author: Qiancong Wu; E-mail: wu_qiancong@163.com
DOI: 10.53106/160792642022112306022

precision annotations can be effectively reduced, certain gaps remain between the final detection result and the full supervision result. In many practical industrial problems, a small number of fully labelled samples can be used during the training process to improve the detection effect. This approach is known as a mixed supervision mode, and the main issue is how many samples require pixel-level labels to satisfy the accuracy requirements of the actual detection task.



(a) KolektorSDD surface defect datasets



(b) DAGM surface defect datasets

Figure 1. Some training samples and their labels

In our proposed method, an improved segmentation and decision network is developed. It is worth noting that hybrid supervision has been used in some applications in the field of image segmentation [14-15]. In this work, we carried out 3 types of experiments on the Severstal Steel Defect dataset [16]: weakly supervised mode, mixed supervised mode and fully supervised mode. In the mixed supervised mode, the segmentation subnetwork uses pixel-level labels, and both subnetworks use image-level labels. In addition, we integrate an attention mechanism into the segmentation subnetwork, use atrous convolutions in both subnetworks, and improve the loss function.

The contributions of this paper can be summarized as follows:

- To explore the use of labels with different annotation precision, we propose an improved segmentation and decision network that implements surface defect detection in four supervised modes.
- To address the small number of targets in the defect region in the defective sample, we integrate an attention mechanism into the segmentation subnetwork to ensure that the training process addresses the defect region in the sample.
- Since the two subnetworks require sufficient receptive fields during the training process, to ensure that the size of the feature graph remains essentially constant, we use atrous convolutions in the networks.
- Since the proposed network has an end-to-end structure with two subnetworks, weight decay is added to the loss function to ensure that computing resources are applied appropriately to the two subnetworks.

The remainder of the article is organized as follows. Section 2 reviews related work. In Section 3, steel surface defect detection using the improved segmentation and decision network is introduced in detail. Section 4 presents our experimental results and an analysis of our method on the Severstal Steel Defect dataset, as well as a comparison with other methods. Finally, our conclusions are discussed in Section 5.

2 Related Work

Advances in deep learning have led to the development of a large number of excellent defect detection algorithms that can be roughly divided into two categories: supervised and unsupervised. Due to the lack of positive samples for defect detection tasks in actual industrial settings, effective supervision information is not available. In addition, sample labels are divided into pixel-level labels and image-level labels. Image-level labels address only whether an image contains defects, while pixel-level labels need to accurately annotate specific defect areas. As a result, the weak supervision mode, which uses only image-level labels, and the mixed supervision mode, which uses both kinds of labels for training, are developed to reduce the number of pixel-level labels.

Full supervision

In fully supervised mode, pixel-level labels are used for network training, and many scholars have studied deep learning applications in the field of defect detection and defect classification in fully supervised mode [17-21]. Masci et al. proposed a max-pooling convolutional neural network approach for supervised steel defect classification that performs considerably better than the commonly used support vector machine (SVM) classifier [22]. Weimer et al. proposed a deep convolutional neural network that achieves excellent results by learning the design and configuration of the network and investigating the influence of different hyperparameter settings on the defect detection accuracy [23]. In some recent defect detection tasks, Kim et al. showed that transfer learning can be successfully applied using image data obtained in a different domain by a pretrained VGG16 network [21]. Gao et al. proposed a mutually supervised few-shot segmentation network that requires a small number of annotated samples to generalize to new categories [20]. Rački et al. designed a unified convolutional neural network (CNN)-based framework for surface anomaly segmentation and detection and applied deep learning techniques for automated visual surface inspection [19]. Ronneberger et al. presented an efficient network and training strategy that rely on data augmentation to better use the available annotated samples, known as U-Net [24]. The architecture consists of a contracting path that captures the context and a symmetric expanding path that enables precise localization. Chen et al. proposed DeepLabv3, which includes an enhanced atrous spatial pyramid pooling module for detecting convolution features and image-level features at multiple scales, further improving performance [25]. In addition, an end-to-end learning methods was demonstrated; however, this approach is limited to full supervision mode. Dong et al. proposed a method for locating and classifying abnormalities using U-Net [26], combining this network with a support vector machine for defect classification and detection. Lin et al. used the small multiscale convolutional neural network MobileNet-v2 for surface defect detection [17]. Huang et al. proposed a lighter network that includes atrous spatial pyramid pooling (ASPP) and deep separable convolution [18].

Unsupervised learning

In unsupervised mode, annotations are not needed, even if they exist. The features are usually learned from the reconstruction objective [27-28], adversarial loss [29] or similar self-supervised objective [30-32]. In the training process, the model typically considers only defect-free images and is trained by using abnormal distribution detection as significant deviations in the features. Various methods have been proposed based on this principle, such as AnoGan [33] and f-AnoGan [34], which learn features from normal samples through generative adversarial networks (GANs).

Weak supervision

Most weakly supervised methods are developed based on semantic segmentation and object detection. Early applications of convolutional neural networks included multiple instance learning (MIL) [35] and constrained CNNs [36]. Saleh et al. proposed extracting a more accurate mask through a pretrained network, activating the mask using a higher-level convolution layer, and smoothing the mask through dense conditional random fields (CRFs) [37]. Bearman et al. proposed a semantic segmentation method that included point supervision, supervision at the combination point of the loss function of the neural network model and novel object potential [38]. Ge et al. explored automated industrial visual inspection and proposed a segmentation-aggregation framework to learn object detectors in weakly annotated visual data [39]. Most other methods use class activation maps (CAMs) [40]. Zhu et al. used CAMs for case segmentation [13], and Diba et al. proposed a new weakly supervised convolutional neural network with a cascaded network structure and introduced structures with either two cascade stages or three end-to-end training stages [41]. CAMs have also been used in the task of defect detection. Lin et al. applied a convolutional neural network for defect detection in LED chips and proposed a class activation mapping technique to locate defect regions without using manual pixel-level annotations [11]. Zhang et al. proposed a weakly supervised learning method known as the category-aware object detection network (CADN) that uses only image-level labels for training and achieves image classification and determines defect locations by extracting category-aware spatial information in the classification pipeline [12]. In the above methods, pixel-level labels are not considered.

Mixed supervision

The method of combining annotation labels with different precisions has been considered in a recent study. Souly et al. proposed a GAN-based semisupervised framework that uses a generator network to provide additional training examples for multiclass classification and acts as a discriminator in the GAN framework. On the one hand, this model uses a large amount of available untagged or weakly tagged data; on the other hand, it uses false images created by the generating countermeasure network for semantic segmentation [14]. Mlynarski et al. solved the problem of segmenting brain tumours in magnetic resonance images [15] and proposed a method that combines completely segmented images with weakly annotated image-level information. These methods focus on segmenting brain tumour images, while we focus on related problems in the field of industrial surface defect detection.

3 Steel Surface-defect Detection Using Improved Segmentation and Decision Networks

3.1 Segmentation and Decision Networks

The basic structure of the segmentation and decision network is shown in Figure 2, and the details of each subnetwork are shown in Table 2. Surface defect detection can be viewed as a binary image classification problem. For surface quality control, compared with defect location and classification, it is more important to quickly and accurately determine whether an image contains defects. Most previous deep learning methods used a large number of samples in their training sets, with the networks extracting features from these datasets for effective learning.

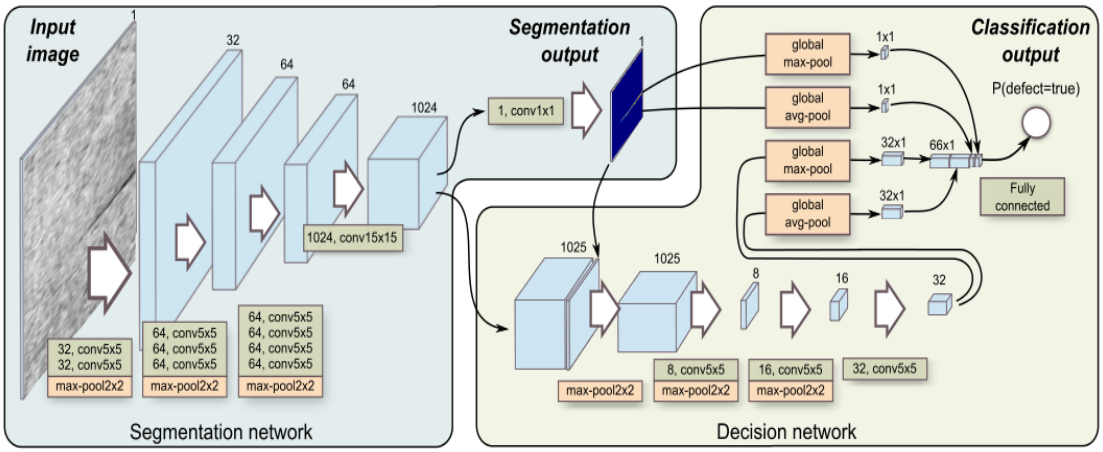


Figure 2. Architecture of the segmentation and decision networks [6]

Table 2. Architecture details for segmentation and decision subnetworks

Segmentation **Subnetwork**				Decision **Subnetwork**			
Layer	Kernel size	Dilation	Features	Layer	Kernel size	Dilation	Features
Input			3	Input			1025
Conv2D	3×3	2	32	Max-Pool	2×2		1025
Conv2D	3×3	2	32	Conv2D	3×3	2	8
Max-Pool	2×2		32	Max-Pool	2×2		8
Conv2D	3×3	2	64	Conv2D	3×3	2	16
Conv2D	3×3	2	64	Max-Pool	2×2		16
Conv2D	3×3	2	64	Conv2D	3×3	2	32
Max-Pool	2×2		64	Max-Pool	2×2		32
Conv2D	3×3	2	64	Avg-Pool	2×2		32
Conv2D	3×3	2	64	Segmentation Output			
Conv2D	3×3	2	64	Max-Pool	2×2		1
Conv2D	3×3	2	64	Avg-Pool	2×2		1
Max-Pool	2×2		64	Segmentation+Decision Outputs			
Conv2D	3×3	7	1024	1,1,32,32			66
Conv2D	1×1		1	FC			1

However, in surface defect detection tasks, the number of available positive samples is insufficient. As a result, networks must be reasonably designed and optimized, and defect detection networks must be trained with a limited number of samples. Our proposed end-to-end architecture includes two subnetworks; that is, our model has a two-stage design. The first subnetwork is the segmentation network, which locates surface defects at the pixel level to achieve defect segmentation. The second stage of the network is called the decision network; this subnetwork is based on the segmentation subnetwork and uses the output of the segmentation subnetwork as an additional feature. To achieve end-to-end simultaneous learning, the loss function combines the losses of two subnetworks into a single loss. In our improved method, an attention mechanism is introduced into the segmentation subnetwork, and atrous convolutions are used in both subnetworks. Considering the different contributions of the two subnetworks to the learning process, we increase the weight decay in the loss function, allowing the training network to gradually shift its focus from segmentation defects to decision making.

The segmentation network is composed of 11 convolution layers and 3 max-pooling layers, and each convolution layer is followed by feature normalization. The input values are standardized to ensure that the scales are in the same range. This process has several advantages. The convergence of the gradient is improved, and the training speed of the model is accelerated. Moreover, each layer can match the input value of each feature distribution as much as possible, which reduces the uncertainty caused by any changes. This process also reduces the impact on the back-layer network, and each network layer is relatively independent, alleviating the problem of gradient disappearance during training. The feature normalization is followed by an ReLU layer, which increases the convergence rate of the learning process. The details of the structure of the segmentation network are shown in the left half of Table 2.

The decision network uses the output of the segmentation network as input. The network uses the output of the final convolution layer in the segmentation network, which has 1024 channels and is connected in series with the single-channel output map of the penultimate convolution layer, resulting in an input with 1025 channels. The input is passed into 8 channels through a max-pooling layer and a convolution layer. The same operation is repeated twice, yielding 16-channel and 32-channel outputs. The details of the structure of the decision network are shown in the right half of Table 2. A global max-pooling operation is performed for the single-channel features obtained by the segmentation network. Then, a global average pooling operation is carried out to obtain two single-channel output features. By performing the same operation on the 32-channel output of the decision network in the final convolution layer, two 32-channel outputs are generated. The four outputs are connected in series, and the final output is obtained through the fully connected layer. This output represents the probability that the sample contains defects.

3.2 Attention Mechanism

When the square ratio of the surface defect area to the image area is less than 0.03, we define the object as a small object; thus, the defect detection task is a small target detection problem. Due to differences in the shapes and sizes of surface defects in the sample and the shooting angle when the sample is imaged, for surface defects that are relatively small or located far from the camera, the proportion of pixels in the image is very low, resulting in a low resolution. As a result, the feature expression ability is reduced, leading to a low detection accuracy or even missed detections. As a resource allocation scheme, the attention mechanism uses limited computing resources to address more important information, which is an effective method for addressing this problem.

However, not all of the content is useful in complex inputs. To reduce the computational burden of the neural network, only key information should be selected and processed by the subsequent network. The attention mechanism in deep learning is similar to the human selective visual attention mechanism, and its core goal is to select information that is critical to the current task goal from the considered information. In deep learning, the concept of attention was first proposed in computer vision for extracting image features. Gao et al. proposed a deep feature and attention mechanism-based method for dish health assessment that applies a hand-

deep local-global net (HDLGN) to dish image recognition [42]. At present, attention mechanisms have achieved good results in various tasks, such as speech recognition, text classification, and machine translation.

The core idea of the attention mechanism is shown in Figure 3. The constructor in the source can be regarded as a series of data pairs, namely, <Key, Value>. Given an element query in the target, the weight coefficient of each key corresponding to the value is obtained by calculating the similarity or correlation between the query and each key, and the final attention value is a weighted summation of the values. Therefore, the attention mechanism is essentially a weighted summation of the values in the source, with the query and key used to calculate the weight coefficients of the corresponding value.

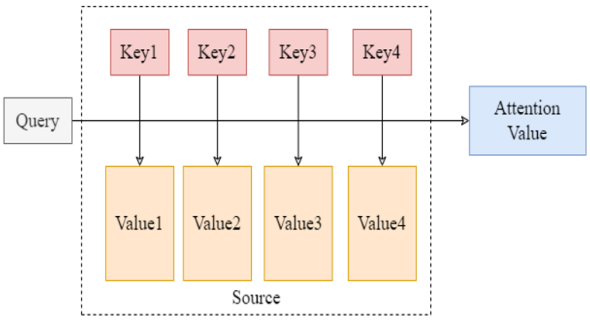


Figure 3. The core idea of the attention mechanism

The specific calculation of the attention mechanism can be divided into two processes: the first process calculates the weight coefficient according to the query and key, and the second process weights and adds the values according to the weight coefficients. The first process can be subdivided into two stages: the first stage calculates the similarity or correlation between the query and key, while the second stage normalizes the original score of the first stage. Thus, the attention calculation process can be abstracted into three stages, as shown in Figure 4.

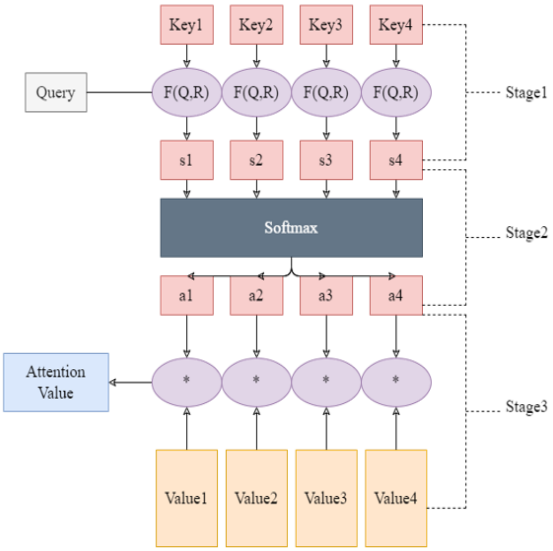


Figure 4. The process for calculating the attention value

We integrate an attention mechanism into the segmentation subnetwork to allow the network to automatically learn locations that require attention in the sample, that is, defect areas in defect samples. To integrate the attention mechanism into the segmentation subnetwork, we designed an SE module. The principle of the SE module is shown in Figure 5. The module generates a mask through the neural network and learns correlations between the channels to focus attention on specific channels. Our segmentation subnetwork includes 11 convolutional layers, with the output results of the 3rd to 9th convolutional layers containing 64 channels and the output results of the 10th convolutional layer containing 1024 channels. We add the SE module after these 7 outputs.

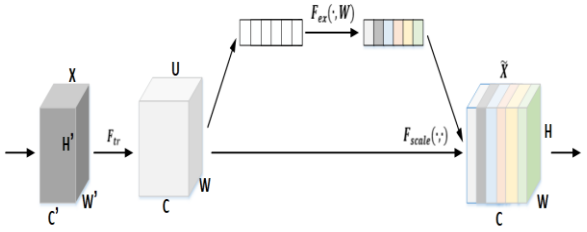


Figure 5. Principles of the SE module

3.3 Atrous Convolution

Most convolution kernels in the original network are 5×5, while some are 15×15. Although this increases the receptive field, the large convolution kernels lead to sharp increases in the computational cost, which is not conducive as the depth of the model increases, reducing the computational performance. Experiments on the VGG and Inception networks have shown that the combination of two 3×3 convolution kernels is better than the use of one 5×5 convolution kernel, reducing the number of parameters; thus, 3×3 convolution kernels are widely used in various models.

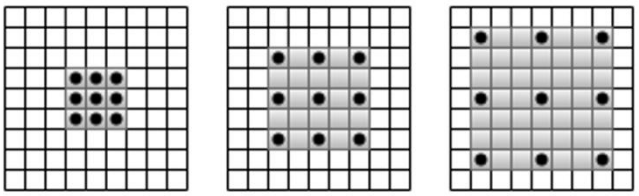
Atrous convolution, also known as dilated convolution, is a method that increases the receptive field of the output unit while not increasing the number of parameters [43-44]. Atrous convolution was originally proposed to address image segmentation issues. The typical method is to use pooling and convolutional layers to increase the receptive field and reduce the size of the feature image and then use upsampling to restore the image size. However, the process of reducing and magnifying the feature image reduces the accuracy. Therefore, an operation that maintains the size of the feature image while increasing the receptive field is needed to replace the downsampling and upsampling operations, resulting in atrous convolution.

In contrast to ordinary convolutions, atrous convolutions introduce a super parameter known as the dilation rate, which defines the spacing between values when the convolution kernel processes data. Taking a 3×3 convolution kernel as an example, an atrous convolution is shown in Figure 6. Atrous convolutions can increase the receptive field, achieving effects similar to larger convolution kernels. The effective size of the convolution kernel can be calculated as:

$$K' = K + (K - 1) \times (D - 1), \quad \textbf{(1)}$$

where K represents the size of the convolution kernel, D represents the dilation rate, and K' represents the effective size of the convolution kernel. For a 3×3 convolution kernel, if D is set to 7, the effective size of the convolution kernel is 15.

Some details on the use of atrous convolutions in segmentation and decision networks are shown in Table 2.



(a) dilation rate = 0 (b) dilation rate = 2 (c) dilation rate = 3

Figure 6. Atrous convolution (3×3)

In the 11 convolutional layers of the segmentation subnet, a 5×5 convolutional kernel is required for the first nine convolutional layers to ensure sufficient receptive fields, increasing the number of parameters. Therefore, we introduce atrous convolutions, using a 3×3 convolution kernel and setting the dilation rate to 2. Similarly, for the tenth convolution layer, a 15×15 convolution kernel is needed, and the dilation rate is set to 7 when a 3×3 convolution kernel is used. In the three convolutional layers in the decision subnetwork, no 5×5 convolutional kernels are used; instead, 3×3 convolutional kernels are applied, and the dilation rate is set to 2.

3.4 Weight Decay

As mentioned above, a loss function is designed to combine the losses of the two subnetworks into a single loss. According to the different tasks of the two subnetworks, the mean squared error loss (MSE) is used for the segmentation network, and the cross-entropy loss is used for the decision network. A description of the symbols used below is shown in Table 3. The loss function is defined as:

$$\mathcal{L}_{total} = \lambda \cdot \gamma \cdot \mathcal{L}_{seg} + (1 - \lambda) \cdot \delta \cdot \mathcal{L}_{dec}, \quad \textbf{(2)}$$

where λ is a balance factor that is defined as:

$$\lambda = 1 - {}^{n}/_{n_{ep}}, \quad \textbf{(3)}$$

where n represents the current training epoch and n_{ep} represents the total number of training epochs. According to this equation, halfway through the training process, the training centre shifts towards the decision network. However, according to the experimental results, at this point, the segmentation network has not yet achieved its best results. Therefore, weight decay is introduced to alleviate the excessive tilt of the training centre towards the decision network. This weight decay is defined as:

$$\theta_t \leftarrow (1 - \beta) \cdot \theta_{t-1} - \alpha \cdot g_t. \quad \textbf{(4)}$$

Thus, the final loss function is defined as:

$$\mathcal{L}_{total} = \theta_t \cdot \gamma \cdot \mathcal{L}_{seg} + (1 - \theta_t) \cdot \delta \cdot \mathcal{L}_{dec}. \quad \textbf{(5)}$$

Table 3. Symbols used in the equations

Symbol	Description
$\mathcal{L}_{total}$	Total loss function
λ	Balance factor
γ	Whether a positive sample exists
$\mathcal{L}_{seg}$	Segmentation network loss
δ	Weight delta for decision loss
$\mathcal{L}_{dec}$	Decision network loss
n	Current epoch
$\mathbf{n_{ep}}$	Total number of epochs
θ_t	Decay coefficient
β	Weight decay coefficient
α	Learning rate
g_t	Current gradient

4 Experiments and Analysis

4.1 Implementation Details

The proposed network was trained and tested using the PyTorch framework and a single NVIDIA RTX3080 GPU. Our experiments were carried out on the Severstal Steel Defect dataset. The dataset contains 12,568 greyscale images divided into 4 classes. According to the original method, we changed the total number of positive samples used during each round of the training process. Thus, different numbers of epochs were used for each experiment. We performed several types of experiments in three modes: weak supervision, mixed supervision and full supervision. The total number of positive samples used during training is recorded as N_{all}, the number of pixel-level labels is recorded as L_p, and the total number of training epochs is recorded as N_{ep}. The specific parameter settings of the experiments are shown in Table 4.

Table 4. Experimental parameter setting

N_{all}	L_p	N_{ep}	β	γ	δ	Learning rate
300	0	90	0.01	1	0.1	0.1
	10	90				
	50	90				
	150	90				
	300	90				
750	750	80	0. 01			
1500	1500	60	0. 015			
3000	3000	40	0. 02			

Table 5. Experimental results

(a) WS

Mode	N_{all}	L_p	N_{ep}	AUC	AP	f-m	FP	FN
WS	300	0	90	0. 918	**0. 932**	0. 843	174	198
	750	0	80	0. 918	**0. 933**	0. 847	171	196
	1500	0	60	0. 920	**0. 937**	0. 849	168	192
	3000	0	40	0. 958	**0. 962**	0. 914	133	102

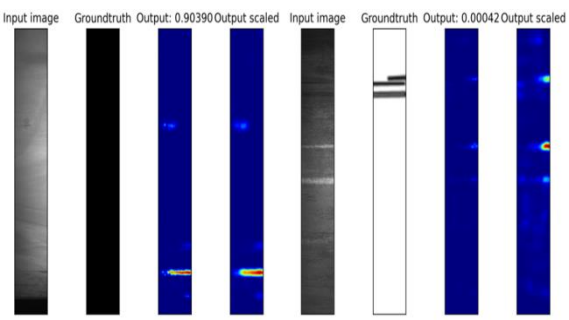
(b) MS

Mode	N_{all}	L_p	N_{ep}	AUC	AP	f-m	FP	FN
MS	300	10	90	0. 918	**0. 935**	0. 845	170	195
		50		0. 952	**0. 959**	0. 907	117	83
		150		0. 963	**0. 972**	0. 927	96	72
	750	10	80	0. 918	**0. 934**	0. 843	171	196
		50		0. 918	**0. 934**	0. 842	169	198
		150		0. 957	**0. 962**	0. 913	130	104
		300		0. 979	**0. 983**	0. 945	47	51
	1500	10	60	0. 923	**0. 942**	0. 870	129	97
		50		0. 954	**0. 969**	0. 921	93	69
		150		0. 980	**0. 983**	0. 947	44	53
		300		0. 979	**0. 983**	0. 949	46	52
		750		0. 983	**0. 985**	0. 952	42	51
	3000	10	40	0. 949	**0. 961**	0. 910	103	74
		50		0. 953	**0. 965**	0. 918	82	63
		150		0. 969	**0. 982**	0. 939	53	50
		300		0. 978	**0. 985**	0. 947	46	50
		750		0. 981	**0. 987**	0. 952	43	49
		1500		0. 985	**0. 988**	0. 957	42	48

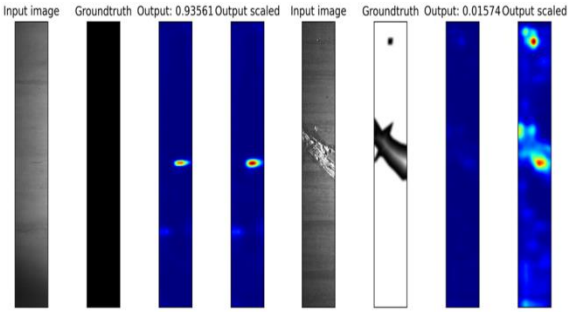
(c) FS

Mode	N_{all}	L_p	N_{ep}	AUC	AP	f-m	FP	FN
FS	300	300	90	0. 978	**0. 982**	0. 944	46	54
	750	750	80	0. 985	**0. 990**	0. 966	41	47
	1500	1500	60	0. 989	**0. 993**	0. 971	38	36
	3000	3000	40	0. 993	**0. 996**	0. 979	35	32

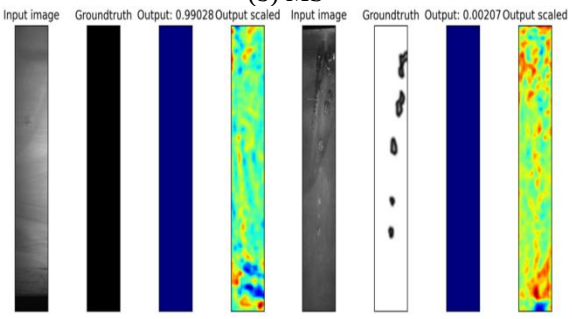
We evaluate the experimental results with five super parameters: AUC is a performance index that measures the advantages and disadvantages of the learners, AP is the average precision, F-M represents the f measure, which is the harmonic average of the accuracy and recall, FP represents the number of false positive samples and FN represents the number of false negative samples. Some examples of FPs and FNs generated in the three modes are shown in Figure 7. The left half of Figure 7(a) shows an example of a false positive sample, namely, a negative sample that was incorrectly identified as a positive sample during the detection process. The right half shows an example of a false negative sample, namely, a positive sample that was incorrectly identified as a negative sample. Figure 7(b) and Figure 7(c) are organized similarly.



(a) WS



(b) MS



(c) FS

Figure 7. False positive and false negative sample

First, we used only image-level labels for training in weak supervision mode. In this mode, pixel-level labels, even if they

can be obtained, are not used; thus, the values of the pixel-level labels are always set to 0. Then, we conducted experiments in mixed supervision mode. To study the number of pixel-level labels required to ensure that mixed supervision mode reaches an accuracy similar to that of full supervision mode, the number of pixel-level labels is gradually increased, namely, $0 < < N_{all}$. The experimental results are shown in Table 5.

4.2 Comparison Results

Tabernik et al. proposed a segmentation-based deep learning structure for surface defect detection and segmentation [6]. This method uses only a small number of defective training samples and achieves ideal results on the KolektorSDD dataset. On this basis, Jakob et al. explored the use of annotations with different levels of precision and carried out experiments in weak supervision, mixed supervision and full supervision modes [45]. This method was comprehensively evaluated on several industrial quality inspection datasets, including KolektorSDD, DAGM and Severstal Steel Defect. In all three supervision modes, this method was superior to all related methods. We performed a large number of comparative experiments with our improved method, and the results are shown in Figure 8.

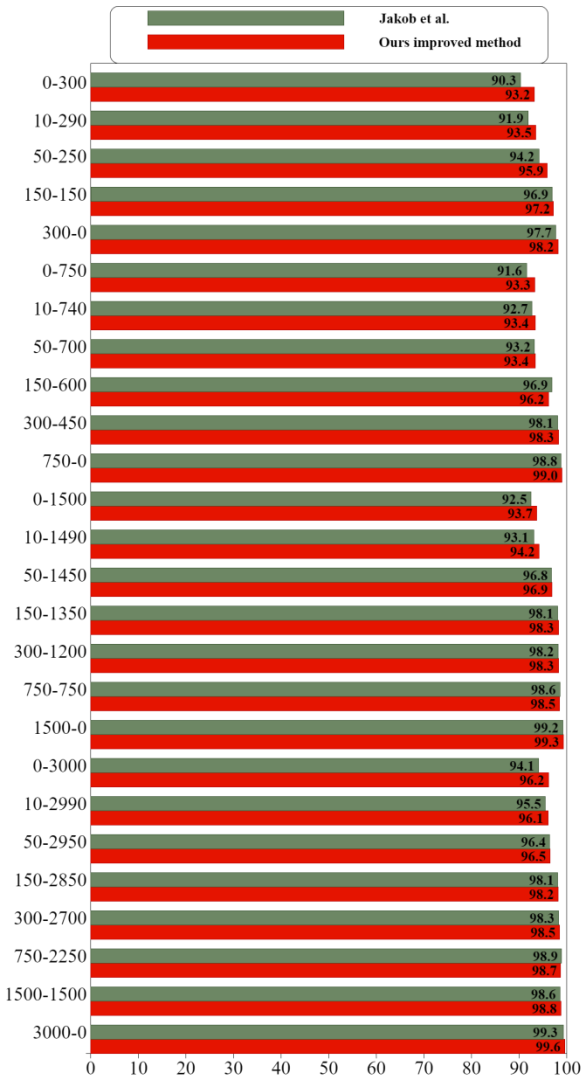


Figure 8. Comparison of the experimental results

The numbers at the bottom of the chart represent the AP, the two numbers on the left represent the number of pixel-level labels and image-level labels used in a single training session, and the sum of the two numbers represents the total number of positive samples used during each training session. For example, 0-300 means that a total of 300 positive samples were used during training, including 0 pixel-level labels and 300 image-level labels; thus, this training was a type of weakly supervised learning.

The figure shows that our proposed method performs well in weakly supervised mode, with an average increase of 2%, due to our improved feature extraction approach. However, in fully supervised mode, the features extracted from the network are sufficient due to the use of pixel-level labels; thus, our method does not significantly improve performance. In mixed supervised mode, approximately 45% of pixel-level labels can be used to achieve a detection accuracy similar to that of the fully supervised mode.

4.3 Ablation Study

Finally, we evaluated the effects of the attention mechanism, atrous convolutions and weight decay on the overall experimental results on the DAGM, KolektorSDD and Severstal Steel Defect datasets. To reduce the training time, the training samples in each group of experiments are not meticulously divided. The number of positive samples used on the Severstal Steel Defect dataset is set to 1000 ($\mathbf{N_{all}}$ = 1000), the number of training rounds is set to 50 ($\mathbf{N_{ep}}$ = 50), and 25% of the pixel-level labels are used in mixed supervision mode ($\mathbf{L_p}$ = 250).

We evaluate our proposed method by using a single improvement in the proposed method, a combination of two improvements and the simultaneous application of all three improvements. The results are shown in Table 6. The results show that the worst performance occurs when no options are used, while the best performance occurs when all three options are used. An in-depth study of the overall improvement method shows that the key to the performance improvement lies in the success of the defect segmentation task. The attention mechanism enables the segmentation network to focus on defect areas in the defect samples, and the atrous convolutions notice slight features in the feature map while increasing the receptive field. Moreover, the weight decay ensures that the computing resources are inclined towards the segmentation network and do not tilt towards the decision network too quickly. The segmentation effect of some samples in the experiment is shown in Figure 9. When no options are applied, the detection results of the two positive sample images in Figure 9 show that the probability that the image contains defects is 0. When we use any combination of two options, the probability of detecting defects in the sample reaches up to 81.2%. When all three options are used, the probability of detecting defects in the sample reaches 97.7%.

When only the attention mechanism is introduced into the segmentation network, the performance of the three modes improves only slightly. However, this does not mean that this component is redundant; because of the large convolution kernel in the subsequent network, subtle defect features identified by the attention mechanism are ignored. After atrous convolutions are added, Table 6 shows that the performance improves. The AP in weak supervision mode increased by 2.26% (from 90.27% to 92.53%); in mixed supervision mode,

the AP increased by 0.4% (from 97.1% to 97.5%); and in full supervision mode, the AP increased by 0.2% (from 98.7% to 98.9%). Weight decay can be introduced only after the introduction of the attention mechanism or atrous convolutions; otherwise, since the segmentation network does not capture new features, it is meaningless to allocate more computing resources. After weight decay is added to the loss function, the best effect is achieved. The AP in weak supervision mode increased by 2.89% (from 90.27% to 93.16%); in mixed supervision mode, the AP increased by 0.8% (from 97.1% to 97.9%); and in full supervision mode, the AP increased by 0.5% (from 98.7% to 99.2%).

Table 6. AP results of the ablation experiment

Mode	DAGM	KSDD	Severstal steel	Attention mechanism	Atrous convolution	Weight decay
WS	74. 82	92. 65	90. 27			
	75. 06	92. 91	91. 31	☑		
	76. 12	93. 12	92. 25		☑	
	75. 49	93. 01	92. 53	☑	☑	
	76. 77	93. 05	91. 08	☑		☑
	76. 98	93. 27	92. 82		☑	☑
	77. 21	**93. 41**	**93. 16**	☑	☑	☑
MS	68. 43	98. 92	97. 1			
	68. 85	97. 63	97. 2	☑		
	71. 24	99. 00	97. 4		☑	
	71. 49	98. 68	97. 5	☑	☑	
	69. 18	97. 81	97. 3	☑		☑
	71. 35	99. 24	97. 6		☑	☑
	72. 18	**99. 36**	**97. 9**	☑	☑	☑
FS	88. 74	99. 46	98. 7			
	89. 10	98. 26	98. 7	☑		
	90. 18	99. 51	98. 8		☑	
	89. 56	99. 24	98. 9	☑	☑	
	88. 97	99. 43	98. 7	☑		☑
	90. 98	99. 87	99. 0		☑	☑
	91. 03	**100. 00**	**99. 2**	☑	☑	☑

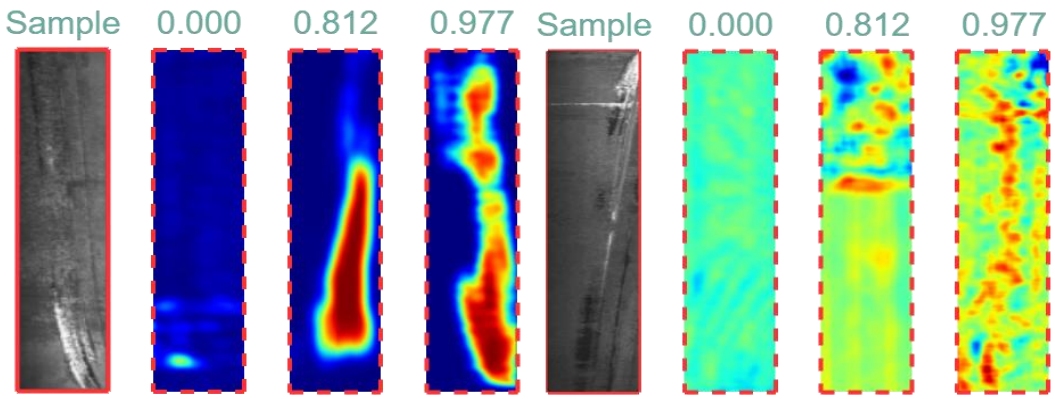


Figure 9. Different segmentation effects

Because we use only a small subset of the DAGM dataset in the ablation experiments, the overall detection accuracy is low. However, our goal was to study the impact of each component on network performance, not evaluate the final detection effect of our method on this dataset. The KSDD dataset is a very small dataset containing only 52 defect images and 347 defect-free images, which leads to underfitting of the network during the training process and a high detection accuracy.

5 Discussion

This paper proposes an improved surface defect detection method based on segmentation and decision networks. By incorporating an attention mechanism, atrous convolutions and an improved loss function, the segmentation results of the segmentation network for defect regions in defect samples are improved. Compared with existing advanced methods, the surface defect detection accuracy is improved.

We also performed a large number of experiments on the DAGM and KolektorSDD datasets, and the results show that our proposed method is robust. However, because these two

datasets are smaller than the Severstal Steel Defect dataset, we did not conduct more in-depth experiments.

Considering that our improvements were focused on the segmentation network, our future work will focus on improving the performance of the decision network. In addition, we intend to enhance the original dataset. Various GAN-based data enhancement methods have been developed. Gao et al. proposed a GAN-based automatic property generation (GAPG) approach for generating verification properties, supporting model checking [46]. We aim to expand our training dataset through GANs to improve the detection accuracy.

References

[1] S. Li, J. Yang, Z. Wang, S. Zhu, G. Yang, Review of development and application of defect detection technology, *Acta Automatica Sinica*, Vol. 46, No. 11, pp. 2319-2336, November, 2020.

[2] X. Wei, Z. Yang, Y. Liu, D. Wei, L. Jia, Y. Li, Railway track fastener defect detection based on image processing and deep learning techniques: A comparative study, *Engineering Applications of Artificial Intelligence*, Vol. 80, pp. 66-81, April, 2019.

[3] C. Jian, J. Gao, Y. Ao, Automatic surface defect detection for mobile phone screen glass based on machine vision, *Applied Soft Computing*, Vol. 52, pp. 348-358, March, 2017.

[4] H. Gao, C. Liu, Y. Yin, Y. Xu, Y. Li, A Hybrid Approach to Trust Node Assessment and Management for VANETs Cooperative Data Communication: Historical Interaction Perspective, *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, pp. 1-10, November, 2021.

[5] Y. Yang, R. Yang, L. Pan, J. Ma, Y. Zhu, T. Diao, L. Zhang, A lightweight deep learning algorithm for inspection of laser welding defects on safety vent of power battery, *Computers in Industry*, Vol. 123, Article No. 103306, December, 2020.

[6] D. Tabernik, S. Šela, J. Skvarč, D. Skočaj, Segmentation-based deep-learning approach for surface-defect detection, *Journal of Intelligent Manufacturing*, Vol. 31, No. 3, pp. 759-776, March, 2020.

[7] H. Gao, B. Qiu, R. J. D. Barroso, W. Hussain, Y. Xu, X. Wang, TSMAE: A Novel Anomaly Detection Approach for Internet of Things Time Series Data Using Memory-Augmented Autoencoder, *IEEE Transactions on Network Science and Engineering (TNSE)*, pp. 1-1, March, 2022.

[8] B. Staar, M. Lütjen, M. Freitag, Anomaly detection with convolutional neural networks for industrial surface inspection, *Procedia CIRP*, Vol. 79, pp. 484-489, 2019.

[9] P. Bergmann, M. Fauser, D. Sattlegger, C. Steger, Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings, 2020 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, Washington, USA, 2020, pp. 4182-4191.

[10] V. Zavrtanik, M. Kristan, D. Skočaj, Reconstruction by inpainting for visual anomaly detection, *Pattern Recognition*, Vol. 112, Article No. 107706, April, 2021.

[11] H. Lin, B. Li, X. Wang, Y. Shu, S. Niu, Automated defect inspection of LED chip using deep convolutional neural network, *Journal of Intelligent Manufacturing*, Vol. 30, No. 6, pp. 2525-2534, August, 2019.

[12] J. Zhang, H. Su, W. Zou, X. Gong, Z. Zhang, F. Shen, CADN: A weakly supervised learning-based category-aware object detection network for surface defect detection, *Pattern Recognition*, Vol. 109, Article No. 107571, January, 2021.

[13] Y. Zhu, Y. Zhou, H. Xu, Q. Ye, D. Doermann, J. Jiao, Learning Instance Activation Maps for Weakly Supervised Instance Segmentation, *IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019, pp. 3116-3125.

[14] N. Souly, C. Spampinato, M. Shah, Semi Supervised Semantic Segmentation Using Generative Adversarial Network, *International Conference on Computer Vision*, Venice, Italy, 2017, pp. 5688-5696.

[15] P. Mlynarski, H. Delingette, A. Criminisi, N. Ayache, Deep learning with mixed supervision for brain tumor segmentation, *Journal of Medical Imaging*, Vol. 6, No. 3, Article No. 034002, July, 2019.

[16] Kaggle, Severstal: Steel Defect Detection on Kaggle Challenge, https://www.kaggle.com/c/severstal-steel-defect-detection.

[17] Z. Lin, H. Ye, B. Zhan, X. Huang, An efficient network for surface defect detection, *Applied Sciences*, Vol. 10, No. 17, Article No. 6085, September, 2020.

[18] Y. Huang, C. Qiu, X. Wang, S. Wang, K. Yuan, A compact convolutional neural network for surface defect inspection, *Sensors*, Vol. 20, No. 7, Article No. 1974, April, 2020.

[19] D. Rački, D. Tomaževič, D. Skočaj, A compact convolutional neural network for textured surface anomaly detection, *IEEE Winter Conference on Applications of Computer Vision*, Lake Tahoe, NV, USA, 2018, pp. 1331-1339.

[20] H. Gao, J. Xiao, Y. Yin, T. Liu, J. Shi, A Mutually Supervised Graph Attention Network for Few-Shot Segmentation: The Perspective of Fully Utilizing Limited Samples, *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, pp. 1-13, March, 2022.

[21] S. Kim, W. Kim, Y.-K. Noh, F. C. Park, Transfer learning for automated optical inspection, *International Joint Conference on Neural Networks*, Anchorage, AK, USA, 2017, pp. 2517-2524.

[22] J. Masci, U. Meier, D. Ciresan, J. Schmidhuber, G. Fricout, Steel defect classification with Max-Pooling Convolutional Neural Networks, *The 2012 International Joint Conference on Neural Networks (IJCNN)*, Brisbane, QLD, Australia, 2012, pp. 1-6.

[23] D. Weimer, B. Scholz-Reiter, M. Shpitalni, Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection, *CIRP Annals-Manufacturing Technology*, Vol. 65, No. 1, pp. 417-420, 2016.

[24] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Munich, Germany, 2015, pp. 234-241.

[25] L.-C. Chen, G. Papandreou, F. Schroff, H. Adam, *Rethinking Atrous Convolution for Semantic Image Segmentation*, June, 2017, https://arxiv.org/abs/1706.05587.

[26] X. Dong, C. J. Taylor, T. F. Cootes, Defect Detection and Classification by Training a Generic Convolutional Neural Network Encoder, *IEEE Transactions on Signal Processing*, Vol. 68, pp. 6055-6069, October, 2020.

[27] D. P. Kingma, D. J. Rezende, S. Mohamed, M. Welling, Semi-Supervised Learning with Deep Generative Models, *Proceedings of the 27th International Conference on Neural Information Processing Systems*, Montreal, Canada, 2014, pp. 3581-3589.

[28] X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, l. Sutskever, P. Abbeel, Variational Lossy Autoencoder, *International Conference on Learning Representations*, Toulon, France, 2017, pp. 1-17.

[29] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative Adversarial Networks, *Communications of the ACM*, Vol. 63, No. 11, pp. 139-144, November, 2020.

[30] I. Croitoru, S.-V. Bogolin, M. Leordeanu, Unsupervised learning from video to detect foreground objects in single images, *International Conference on Computer Vision*, Venice, Italy, 2017, pp. 4335-4343.

[31] X. Wang, K. He, A. Gupta, Transitive Invariance for Self-supervised Visual Representation Learning, *International Conference on Computer Vision*, Venice, Italy, 2017, pp. 1338-1347.

[32] R. Zhang, P. Isola, A. A. Efros, Split-Brain Autoencoders: Unsupervised Learning by Cross-Channel Prediction, *2017 IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, Hawaii, USA, 2017, pp. 645-654.

[33] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, G. Langs, Unsupervised anomaly detection with generative adversarial networks to guide marker discovery, *International Conference on Information Processing in Medical Imaging*, Boone, USA, 2017, pp. 146-157.

[34] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, U. Schmidt-Erfurth, f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks, *Medical Image Analysis*, Vol. 54, pp. 30-44, May, 2019.

[35] D. Pathak, E. Shelhamer, J. Long, T. Darrell, Fully Convolutional Multi-Class Multiple Instance Learning, *International Conference on Learning Representations Workshop*, San Diego, CA, USA, 2015, pp. 1-4.

[36] D. Pathak, P. Krahenbuhl, T. Darrell, Constrained Convolutional Neural Networks for Weakly Supervised Segmentation, *International Conference on Computer Vision*, Santiago, Chile, 2015, pp. 1796-1804.

[37] F. Saleh, M. S. A. Akbarian, M. Salzmann, L. Petersson, S. Gould, J. M. Alvarez, Built-in foreground /background prior for weakly-supervised semantic segmentation, *European Conference on Computer Vision*, Amsterdam, The Netherlands, 2016, pp. 413-432.

[38] A. Bearman, O. Russakovsky, V. Ferrari, F.-F. Li, What's the point: Semantic segmentation with point supervision, *European Conference on Computer Vision*, Amsterdam, The Netherlands, 2016, pp. 549-565.

[39] C. Ge, J. Wang, J. Wang, Q. Qi, H. Sun, J. Liao, Towards automatic visual inspection: A weakly supervised learning method for industrial applicable object detection, *Computers in Industry*, Vol. 121, Article No. 103232, October, 2020.

[40] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning Deep Features for Discriminative Localization, *2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016, pp. 2921-2929.

[41] A. Diba, V. Sharma, A. Pazandeh, H. Pirsiavash, L. V. Gool, Weakly supervised cascaded convolutional networks, *2017 IEEE Conference on Computer Vision and Pattern Recognition*, Hawaii, HI, USA, 2017, pp. 5131- 5139.

[42] H. Gao, K. Xu, M. Cao, J. Xiao, Q. Xu, Y. Yin, The Deep Features and Attention Mechanism-Based Method to Dish Healthcare Under Social IoT Systems: An Empirical Study With a Hand-Deep Local-Global Net, *IEEE Transactions on Computational Social Systems (TCSS)*, Vol. 9, No. 1, pp. 336-347, February, 2022.

[43] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille, DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 40, No. 4, pp. 834-848, April, 2018.

[44] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, *International Conference on Learning Representations*, San Juan, Puerto Rico, 2016, pp. 1-4.

[45] J. Božič, D. Tabernik, D. Skočaj, Mixed supervision for surface-defect detection: from weakly to fully supervised learning, *Computers in Industry*, Vol. 129, Article No. 103459, August, 2021.

[46] H. Gao, B. Dai, H. Miao, X. Yang, R. J. D. Barroso, W. Hussain, A Novel GAPG Approach to Automatic Property Generation for Formal Verification: The GAN Perspective, *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, February, 2022, https://doi.org/10.1145/3517154.

Biographies

Zhongqin Bi received his Ph.D. degree in system analysis and integration from East China Normal University, China in 2009. He is currently a professor in College of Computer Science and Technology, Shanghai University of Electric Power. Dr. Bi has published more than 30 articles in refereed journals and conference proceedings. His main research interests are cloud computing, data process and quality control in smart grid.

Qiancong Wu is currently pursuing the M.S. degree in computer technology with the Shanghai University of Electric Power, Shanghai, China. His research interests include object detection and image generation.

Meijing Shan received her Ph.D. from East China Normal University, China in 2009. She is an Assistant Professor at East China University of Political Science and Law. She has published the results of her research in more than 20 papers in international journals, conference proceedings and book chapters. Her research areas include cybercrime and computer forensics.

Wei Zhong received his master degree in computer application technology from Wuhan University of Technology. He is currently senior engineer and expert of the group in No. 34 Research Institute, China Electronics Technology Group Corporation. He has published more than 10 articles, more than 10 invention patents and 10 software copyrights. His research includes AI, data processing and IOT control.