

An End-to-End Steel Surface Defect Detection Approach via Fusing Multiple Hierarchical Features

Yu He, Kechen Song^{ID}, Qinggang Meng^{ID}, and Yunhui Yan

Abstract—A complete defect detection task aims to achieve the specific class and precise location of each defect in an image, which makes it still challenging for applying this task in practice. The defect detection is a composite task of classification and location, leading to related methods is often hard to take into account the accuracy of both. The implementation of defect detection depends on a special detection data set that contains expensive manual annotations. In this paper, we proposed a novel defect detection system based on deep learning and focused on a practical industrial application: steel plate defect inspection. In order to achieve strong classification ability, this system employs a baseline convolution neural network (CNN) to generate feature maps at each stage, and then the proposed multilevel feature fusion network (MFN) combines multiple hierarchical features into one feature, which can include more location details of defects. Based on these multilevel features, a region proposal network (RPN) is adopted to generate regions of interest (ROIs). For each ROI, a detector, consisting of a classifier and a bounding box regressor, produces the final detection results. Finally, we set up a defect detection data set NEU-DET for training and evaluating our method. On the NEU-DET, our method achieves 74.8/82.3 mAP with baseline networks ResNet34/50 by using 300 proposals. In addition, by using only 50 proposals, our method can detect at 20 ft/s on a single GPU and reach 92% of the above performance, hence the potential for real-time detection.

Index Terms—Automated defect inspection (ADI), defect detection dataset (NEU-DET), defect detection network (DDN), multilevel-feature fusion network (MFN).

I. INTRODUCTION

DEFFECT inspection is a crucial step to guarantee the quality of industrial production, especially for steel plates. However, this process is usually performed manually

Manuscript received March 21, 2019; revised April 28, 2019; accepted April 29, 2019. Date of publication May 8, 2019; date of current version March 10, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 51805078 and Grant 51374063, in part by the National Key Research and Development Program of China under Grant 2017YFB0304200, in part by the Fundamental Research Funds for the Central Universities under Grant N170304014 and Grant N150308001, and in part by the China Scholarship Council under Grant 201806085007. The Associate Editor coordinating the review process was Emanuele Zappa. (Corresponding authors: Kechen Song; Yunhui Yan.)

Y. He, K. Song, and Y. Yan are with the School of Mechanical Engineering and Automation, Northeastern University, Shenyang 110819, China, and also with the Key Laboratory of Vibration and Control of Aero-Propulsion Systems, Ministry of Education of China, Northeastern University, Shenyang 110819, China (e-mail: heyu142616@gmail.com; songkq@me.neu.edu.cn; yanyh@mail.neu.edu.cn).

Q. Meng is with the Department of Computer Science, Loughborough University, Loughborough LE11 3TU, U.K. (e-mail: q.meng@lboro.ac.uk).

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIM.2019.2915404

0018-9456 © 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See <https://www.ieee.org/publications/rights/index.html> for more information.

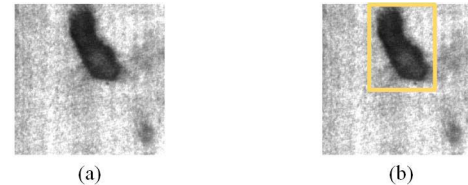


Fig. 1. Defect classification and defect detection task. (a) Defect classification task aims to “What,” only outputting a defect class score. (b) Defect detection task aims to “What” and “Where,” outputting a bounding box with a defect class score.

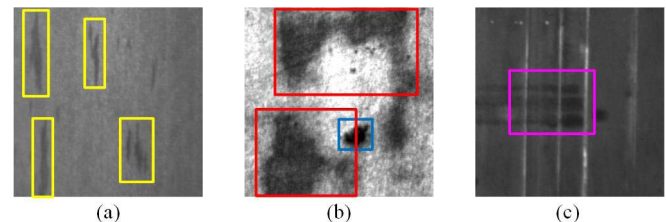


Fig. 2. Complicated defects. (a) Multiple defects. The yellow boxes indicate the defects belong to an identical class. (b) Multiclass defects. The red and blue boxes indicate the defects of different classes. (c) Overlapping defects. The pink box surrounds an overlapping region of defects of different classes.

in industry, which is unreliable and time-consuming. In order to replace the manual work, it is desirable to allow a machine to automatically inspect surface defects from steel plates with the use of computer vision technologies.

The founder of computer vision, British neurophysiologist Marr, considers that a vision task can be defined as “What is Where” that is the process of discovering what presents in an image and where is it [1]. Therefore, the object classification and detection are the most fundamental problems in the field of computer vision research [2]. Similarly, the automated defect inspection (ADI) can also be divided into two types: defect classification and defect detection. Given a defect image, the defect classification task is to solve if this image contains some class of defect [Fig. 1(a)], and the defect detection task is to solve where a defect exists in this image, represented by a bounding box with a class score [Fig. 1(b)]. Therefore, a complete defect detection task consists of two parts: defect classification, determining specific categories of defects, and defect localization, obtaining detailed regions of defects. For defect inspection on steel plates, the detection task has superior advantages to complicated defects, e.g., multiple defects [Fig. 2(a)], multiclass defects [Fig. 2(b)], and overlapping defects [Fig. 2(c)]. The classification task can only find

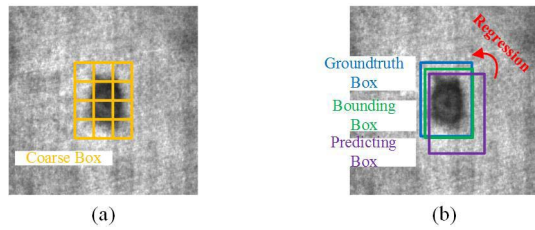


Fig. 3. Different styles of obtaining a defect region. (a) Many previous detectors based on hand-craft features directly combine related spatial cells into a block through various special approaches. The block is regarded as a detection region, which is a coarse box without refining. (b) Detectors based on DL mainly use regression methods to refine a predicting box. Through a large amount of iterative learning, the predicting box is gradually close to the groundtruth box. Finally, the refined box is regarded as the bounding box of the defect, which can represent the precise location information of the defect.

the defect with the highest category confidence in an image and not know the number of defects shown in Fig. 2(a), classes of defects shown in Fig. 2(b), and emerge of an overlapping defect shown in Fig. 2(c). However, for the follow-up quality assessment system, the quantity, category, and complexity of defects would be served as the chief indicators to evaluate the quality of a steel plate. It is apparent that defect detection can achieve a more comprehensive information reflection of a steel plate surface.

The previous ADI methods have two common problems: the one is the unclear usage of hand-craft features [3]–[5]. The determination of features is too subjective, and thereby human experience usually plays a decisive role in it. The other problem is imprecise defect localization [Fig. 3(a)]. Most methods only perform defect classification [6]–[8] or an incomplete defect detection. For example, some methods perform binary classification to find the regions of defects [9], [10] or only provide a coarse region of a defect [11], [12]. The recent developed deep learning (DL) technology can overcome the drawbacks of traditional ADI methods and have achieved significant results on many vision tasks. The DL can extract discriminative representations through a deep network [e.g., a convolution neural network (CNN)]. These representations can reach a high level of abstract and therefore have strong representation ability. The hand-craft features, by contrast, are merely the combination of low-level features [16]. Moreover, DL can train on location-annotated samples to obtain precise location information.

At present, some studies have already applied DL for ADI. However, most methods can only perform defect classification due to the lack of special data sets [18]–[21]. The defect classification seems to be oversimplify and unable to provide location information. Other methods use a combination of DL and traditional image processing to perform defect detection or segmentation [17]. These methods always use a DL classifier in parallel with a detector or a segmenter that based on traditional image processing. This way can eliminate the need for special training data sets but damage the end-to-end characteristic of DL system and lose the intelligence and generalization to some extent. Unlike the above-mentioned methods, we attempt to establish an end-to-end defect detection system for ADI, which can provide

a bounding box with a class score for precisely classifying and locating a defect [Fig. 3(b)]. A DL-based segmenter like Mask R-CNN [13] seems to be better for showing the shape of a defect. However, this kind of segmenter will consume huge amounts of computation source, which cannot meet the real-time demand of industrial inspection. Furthermore, it is highly impracticable for the industry to build a large instance-level defect segmentation data set, and thereby this kind of segmenter is almost impossible to apply. Therefore, it is the best tradeoff to perform defect detection for ADI at present.

This paper mainly addresses three challenges. First, the detection system needs strong classification ability. The common classification problems such as interclass similarity, intraclass difference, and background interference are also present in ADI [9], [11]. Therefore, we equip a deep network ResNet into the system as the backbone [23]. As current research in transfer learning [15], the key to drive large networks is pretraining on ImageNet [22]. The detection system can gain strong classification power by training ResNet on enough data.

Second, the challenge of performing defect localization using CNN features in DL-based methods remains. As we known, the convolutional layers of CNN can be regarded as filters, which results in some location details will be gradually lost when an image flows in the CNN. Usually, DL-based methods perform localization based on the last convolutional feature map [14], [28], [34]. Our method is to fuse multiple feature maps. Because the feature maps exhibit diverse characteristics at each stage of CNNs: the shallow features have rich information but not discriminative enough, and the deep features are semantic robustly but lose too many details. In other fields [34], the HyperNet also uses more features but they are mainly selected from the latter part of the network. The proposed multilevel-feature fusion network (MFN) combines the multiple features covering all stages. We address the detection from the industrial perspective. Since gray images have less information than color images, the MFN must include lower level features that are discarded by HyperNet. Furthermore, the MFN unifies the size of multiple features before fusion, which can not only save more details of images but also use less parameters of models.

Third, in defect detection, data annotation is expensive, because one has to draw a defect's bounding box and assign a class label to it. Recent progress in this field can be attributed to two factors: 1) ImageNet pretrained models and 2) large baseline CNNs, which made great progress in DL-based defect classification [18]–[20]. However, the limited data and expensive annotation still limit the development of defect detection. In this paper, we open a defect detection data set NEU-DET for fine-tuning models. When the DL models have finished training on a special data set, they can be used to perform the defect detection task.

This paper establishes an end-to-end ADI system, called defect detection network (DDN), in an attempt to overcome the above-mentioned challenges. The DDN 1) adopts a strong ResNet in defect classification; 2) proposes the MFN to assemble more location details; and 3) sets up a defect detection data set for fine-tuning and reports improvements on it. In more detail, first, we pretrain the ResNet on the ImageNet and

fine-tune all the models on the NEU-DET. The MFN can fuse the selected features into a multilevel feature, which has characteristics covering all the stages of the ResNet. Next, a region proposal network (RPN) is adopted in proposals generation based on the multilevel features and then the DDN can output the class scores and the coordinates of bounding box. Finally, we evaluate the proposed method on NEU-DET and the results can demonstrate a clear superior to other ADI methods.

To summarize, the main contributions of this paper are as follows.

- 1) The introduction of the end-to-end defect detection pipeline DDN that integrates the ResNet and the RPN for precise defect classification and localization.
- 2) The proposed MFN for fusing multilevel features. Compared with other fusing methods, MFN can combine the lower level and higher level features, which makes multilevel features to have more comprehensive characteristics.
- 3) A defect detection data set NEU-DET for fine-tuning networks and a demonstration that the proposed DDN has a very competitive performance on this data set.

II. RELATED WORK

A. Defect Inspection

Generally, a defect classification method includes two parts: a feature extractor and a classifier. The classic feature extractor is to obtain hand-craft features such as HOG and LBP, and they are always followed by a classifier, e.g., SVM. Therefore, the combination of different feature extractors and classifiers produces a variety of defect classification methods. For instance, Song and Yan [3] improve the LBP to against noise and adopt NNC and SVM to classify defects. Ghori *et al.* [9] is based on a small set of wavelet features and use SVM to perform defect classification. Different from above-mentioned two methods, Chu *et al.* [8] employ a general feature extractor and enhance SVM. From the perspective of computer vision, the defect classification task is essentially defect image classification, which is struggled in complicated defect images. To solve it, the simple and direct way is to perform defect localization before defect classification making the inspection task classify on regions of defects instead of a whole defect image, which is the defect detection task. For example, the defect detectors in [11] and [12] first perform a 0–1 classification to judge features whether belong to a defect class or a nondefect class, and then finds defect regions based on the boundary of defect-class features, finally perform different classification methods to determine the specific class of a defect. In addition, there is another simplified detector for the requirement of quick detection, which only focuses on regions of defects but regardless of the defects are in different categories [10].

However, the DL-based methods differ radically from the above methods. Hand-craft feature extractor locally analyses a single image and extract features. However, CNN is to construct the representation of all the input data through a large amount of learning. CNN has fine generalization

and transferability so that there are some defect inspection methods based on CNN. For example, Chen and Ho [21] demonstrate that an object detector like Overfeat [24] can be transferred to be a defect detector by some means. Similar to [18] and [19], they demonstrate that using a sequential CNN to extract features can improve classification accuracy on defect inspection. Similarly, based on a sequential CNN, Ren *et al.* [17] perform an extra defect segmentation task on classification results to define the boundary of a defect. Moreover, Natarajan *et al.* [20] employ a deeper neural network VGG19 for defect classification. With the depth of CNN, the defect classification accuracy has been further improved.

B. Baseline Networks

There are three popular CNN architectures at present, which are used as baseline networks for pretraining. The early successful networks are based on the sequential pipeline architecture [25], which establish the basic structure of CNN and prove the importance of depth of networks. Subsequently, the inception networks employed modular units, which increase both the depth and width of a network without the increment of computational cost [26]. The third type is ResNet using residual blocks to make networks deeper without overfitting [23]. ResNet is widely applied in various vision tasks, achieving competitive results with a few parameters.

Choosing a proper baseline network is the key to gain good results for DL methods. A large network has strong represent-ability for input data hence the extracted features at high-abstract level, but there is a great demand for training data.

C. CNN Detectors

The CNN detectors aim to classify and locate each target with a bounding box. They are mainly divided into two methods: one is the region-based method and another is the direct regression method. The most famous region-based detectors are the “R-CNN family” [27], [28], [14]. In this framework, thousands of class-independent region proposals are employed for detection. Region-based methods are superior in precision but require slightly more computation. The representative direct regression methods are YOLO [29] and SSD [30]. They directly divide an image into small grids and then for each grid predict bounding boxes, which then regressed to the groundtruth boxes. The direct regression method is fast to detect but struggles in small instances.

III. DEFECT DETECTION NETWORK

In this section, the DDN is described in detail (see Fig. 4). A single-scale image of an arbitrary size is processed by a CNN, and the convolutional feature maps at each stage of the ConvNet are produced (ConvNet represents the convolutional part of a CNN). We extract multiple feature maps and then aggregate them in the same dimension by using a lightweight MFN. In this way, MFN features have the characteristics from several hierarchical levels of ConvNet. Next, RPN [14] is employed to generate region proposals

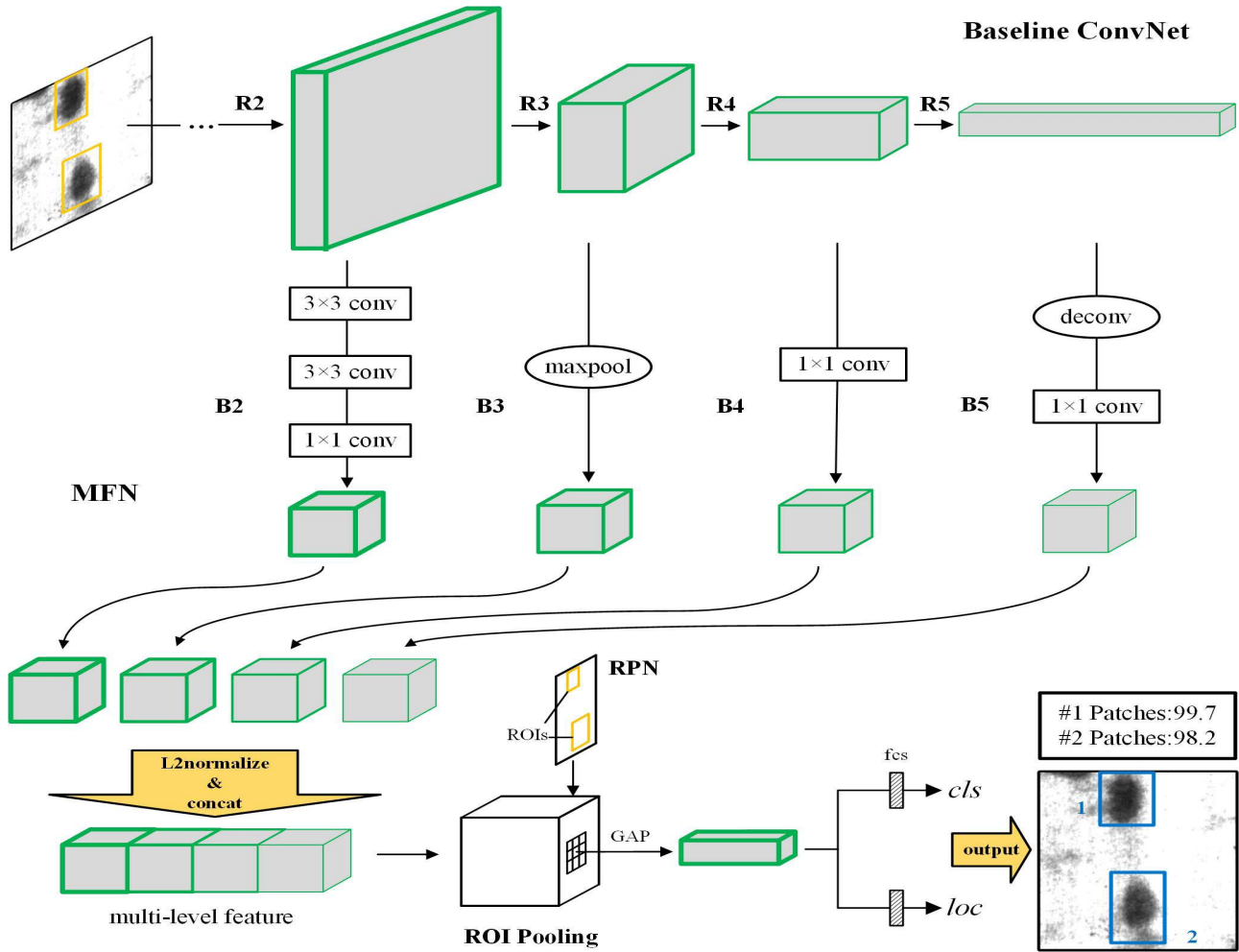


Fig. 4. DDN. In a single pass, we extract features from each stage of the Baseline ConvNet, which then fused into a multilevel feature by MFN. RPN is adopted to generate ROIs based on the multilevel feature. For each ROI, the corresponding multilevel feature is transformed into a fixed-length feature through the ROI pooling and the GAP layers. Two fc layers process each fixed-length feature and feed into output layers producing two results: a one-of-($C + 1$) defect class prediction (cls) and a refined bounding box coordinate (loc).

[regions of interest (ROIs)] over the MFN feature. Finally, the MFN feature corresponding to each ROI is transformed into a fixed-length feature through the ROI pooling [28] and the global average pooling (GAP) layers. The feature is fed into two fully connected (fc) layers. One is a one-of-($C + 1$) defect classification layer (“cls”) and the other is a bounding-box regression layer (“loc”).

The rest of this section introduces the details of DDN and motivates why we need to design MFN into the network for the defect detection task.

A. Baseline ConvNet Architecture

As we know that pretraining on the ImageNet data set is important to achieve competitive performance, and then this pretrained model can be fine-tuned on a relatively small defect data set. In this paper, we select the recent successful baseline network ResNet as the backbone. ResNet presents several attractive advantages as follows.

- 1) ResNet can achieve the state-of-the-art precision with extremely few parameters, in comparison with the CNN

of sequential pipeline architecture of the same magnitude (ResNet50 vs. VGG16, 0.85 M vs. 138 M parameters). It implies that ResNet has lower computational cost and less probability of overfitting.

- 2) ResNet uses GAP to process the final convolutional feature map instead of the dual stacked fc layers, which can be in a manner of preserving more comprehensive location information of defects in the image.
- 3) ResNet has a modularized ConvNet, which is easy to integrate.

In this paper, we select ResNet34 and ResNet50 as baseline networks. The detailed structures of both networks are shown in Table I, and residual blocks are denoted as {R2, R3, R4, R5}.

B. Produce Multilevel Features

Previous excellent approaches only utilize high-level features to extract region proposals (like the faster R-CNN extract proposals upon the last convolutional feature maps). In order to obtain quality region proposals, single-level features should

TABLE I
ARCHITECTURE OF BASELINE NETWORKS

Block name	Conv1	R2	R3	R4	R5	Output
ResNet34	7×7,64, stride=2	$\begin{bmatrix} 3\times 3, 64 \\ 3\times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 3\times 3, 128 \\ 3\times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 3\times 3, 256 \\ 3\times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 3\times 3, 512 \\ 3\times 3, 512 \end{bmatrix} \times 3$	GAP Fc Softmax
ResNet50		$\begin{bmatrix} 1\times 1, 64 \\ 3\times 3, 64 \\ 1\times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times 1, 128 \\ 3\times 3, 128 \\ 1\times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times 1, 256 \\ 3\times 3, 256 \\ 1\times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times 1, 512 \\ 3\times 3, 512 \\ 1\times 1, 2048 \end{bmatrix} \times 3$	
Output size	112×112	56×56	28×28	14×14	7×7	1×1

be extended to multilevel features. Obviously, the simplest way is to assemble feature maps from multiple layers [31]. Therefore, now comes the question, which layers should be combined? There are two essential conditions: nonadjacent, because adjacent layers have highly local correlation [32], and coverage, including features from low level to high level. For a ResNet, the most intuitive way is to combine the last layers in each residual block.

To fuse features at different levels, the proposed network MFN is appended on the pretrained model. MFN has four branches, denoted as {B2, B3, B4, B5}, and each branch is a small network. B2, B3, B4, and B5 are sequentially connected to the last layer of R2, R3, R4, and R5. When an image flows through the baseline ConvNet, the R_i features are produced in order. The R_i feature means the feature map output from the last layer of the residual block R_i , $i = 2, \dots, 5$. Similarly, the B_i feature is the feature map produced from the last layer of the MFN batch B_i , $i = 2, \dots, 5$. Then, each of R_i features is led to the corresponding branch in MFN producing B_i features. Finally, multilevel features are obtained via concatenating the B2, B3, B4, and B5 features, which come from different stages of a CNN.

As a final note, MFN is efficient in computation and strong in generalization. MFN can reduce required parameters via modifying the number of filters of 1×1 conv. This operation may hurt accuracy but prevent overfitting in the case of insufficient training data.

C. Extract Region Proposals

The RPN is employed to extract region proposals by sliding on the multilevel feature maps. RPN takes an image of arbitrary size as input and outputs anchor boxes (candidate boxes), each with a score representing whether it is a defect or not. The originality of RPN is the “anchor” scheme that makes anchor boxes in multiple scales and aspect ratios. Then, anchor boxes are hierarchically mapped to the input image so that region proposals of multiple scales and aspect ratios produced. As a result of the resolution size of MFN feature, the RPN can be considered as sliding on the R4 feature. Follow [14], we set three aspect ratios {1:1, 1:2, 2:1}. Considering multiple sizes of defects, we set four scales $\{64^2, 128^2, 256^2, 512^2\}$. Therefore, RPN produces 12 anchor boxes at each sliding location.

The region proposal extractor always ends with an ROI pooling layer. This layer performs a max-pooling operation

over a feature map inside each ROI to convert it into a small feature vector (512-d for ResNet34 and 2048-d for ResNet50) with a fixed size of $W \times H$ (in this paper, 7×7). At last, based on these small cubes, calculate the offset of each region proposal with an adjacent groundtruth box and the probability whether there exist defects.

For a single image, RPN may extract thousands of region proposals. To deal with the redundant information, the greedy nonmaximum suppression (NMS) is often applied for eliminating high-overlap region proposals. We set the intersection over union (IOU) threshold for NMS at 0.7, which can discard a majority of region proposals. After NMS, the top- K ranked region proposals are selected from the rest. In the following, we fine-tune DDN using top-300 region proposals owing to the extracted quality region proposals, but reduce this number to accelerate the detection speed without harming accuracy at test-time.

IV. TRAINING

A. Multitask Loss Function

The defect detection task can be divided into two subtasks, hence DDN has two output layers. The cls layer outputs a discrete probability distribution, $k = (k_1, \dots, k_C)$, for each ROI over $C + 1$ categories (C defect categories plus one background category). As usual, k is computed by a softmax function. The cls loss L_{cls} is a log loss over two classes (defect or not defect). $L_{\text{cls}} = -\log(k, k^*)$ where k^* is the groundtruth class. The loc layer outputs bounding box regression offsets, $t = (t_x, t_y, t_w, t_h)$, for each of the C defect categories. As in [28], the loc loss L_{loc} is a smooth $L1$ loss function. $L_{\text{loc}} = \text{SmoothL1}(t - t^*)$ where t^* is the groundtruth box associated with a positive sample. For bounding box regression, we adopt the parameterizations of t and t^* given in [27]

$$\begin{aligned}
 t_x &= (x - x_a)/w_a, t_y = (y - y_a)/h_a \\
 t_w &= \log(w/w_a), t_h = \log(h/h_a) \\
 t_x^* &= (x^* - x_a)/w_a, t_y^* = (y^* - y_a)/h_a \\
 t_w^* &= \log(w^*/w_a), t_h^* = \log(h^*/h_a)
 \end{aligned} \tag{1}$$

where the subscripts x, y, w , and h denote each box's center coordinates and its width and height. The variables x, x_a , and x^* separately represent the predicted box, anchor box, and groundtruth box (the same rules for y, w , and h).

With these definitions, we minimize a multitask loss function, which is defined as

$$L(k, k^*, t, t^*) = L_{\text{cls}}(k, k^*) + \lambda p^* L_{\text{cls}}(t, t^*) \tag{2}$$

Algorithm 1 Five-Step Joint Training Algorithm

Defect images with annotations
 Train the merged network for initializing MFN with the pretrained model, obtaining model M_P .
 Train RPN based on M_P , generating proposals P^* .
 Train the detector network using proposals P^* , obtaining model M_D^* .
 Fine-tune RPN based on M_D^* , generating proposals P and obtaining model M_R .
 Fine-tune the detector network using proposals P , obtaining model M_D .
 Combine M_R and M_D as the final model.

where λ is the weight parameter balancing both cls and loc terms. During training, we set $\lambda = 2$ indicating that DDN is devoted to achieving better defect locations. p^* is the activation parameter of the loc term. The localization loss is involved in the subsequent calculation only for positive samples ($p^* = 1$) and is disabled otherwise ($p^* = 0$). We follow the “IOU” strategy in [14] to determine the positive and negative samples from anchors.

B. Joint Training

For pretrained network, MFN and RPN are new layers. Hence, we need to make these three networks share the common convolutional features through training. The pretrained model is essentially a classification network, and multilevel features generated from MFN can be directly fed into the cls layer. Therefore, the pretrained network and MFN can be merged into one network, and then performed an end-to-end training. Without RPN, the rest of DDN is a detector network. To share features with RPN, the four-step alternating training strategy in [14] is adopted, alternating between training RPN and training detector network. Combining these two strategies, we develop a practicable five-step joint training algorithm, which is shown in Algorithm 1.

After step 2 and step 3, RPN and the detector network are initialized with the ImageNet pretrained model in succession. However, these two networks have not shared the convolutional features at this point. They get it until the fine-tuning processes of step 3 and step 4 are finished. Specifically, we freeze the shared convolutional layers and only fine-tune the unshared layers. Finally, we combine two networks as a united network.

C. Implementation

For DDN, we adopt image-centric training strategy. Images are resized such that their short side is 600 pixels. We use stochastic gradient descent to train with a weight decay of 0.0001 and a momentum of 0.9. We take a single image per minibatch iteration. The minibatch size is 64 for detector network training (include MFN training) and 128 for RPN training. We fine-tune the model using a learning rate of 0.001 for 200k minibatch iterations and 0.0001 for another 100k minibatch iterations. We use “Xavier” initialization for

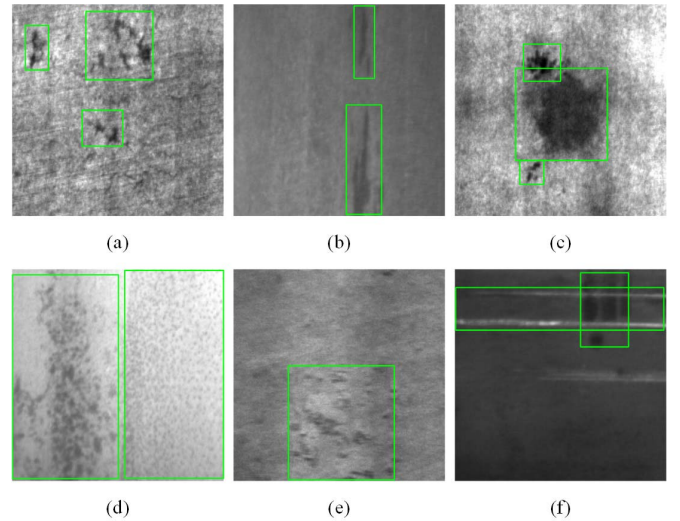


Fig. 5. Examples of defect images with annotations in NEU. The green box is a groundtruth box, which has a class label and two corner coordinates of the box (top-left and bottom-right). The category to which the image belongs (a) crazing, (b) inclusion, (c) patches, (d) pitted surface, (e) rolled-in scale, and (f) scratches.

all new layers [33]. To avoid overfitting, we also use several data augmentation methods such as rotation, reflection, and shift, but remove the dropout module.

V. EXPERIMENTS

The performance of DDN is evaluated on our defect data sets: NEU-CLS and NEU-DET. We demonstrate that DDN achieves a reasonable design and promising results.

A. NEU-DET Data Set

NEU surface defect¹ is a defect classification data set that we opened seven years ago [3]. There are six types of defects from hot-rolled steel plates, including crazing, inclusion, patches, pitted surface, rolled-in scales, and scratches. Each class has 300 images, but it does not mean that an image consists of a single defect. Examples of defect images are shown in Fig. 5.

To perform defect detection tasks, we provide annotations saved as XML files. With them, the classification data set is upgraded to a detection data set. The annotation marks the class and bounding box of each defect appearing in an image. Each bounding box is regarded as a groundtruth box, which is represented by its top left and bottom right coordinates. There are nearly 5000 groundtruth boxes in total. For simplicity, we call the original data set NEU-CLS, and the complemented data set NEU-DET. Examples of annotations are also shown in Fig. 5.

B. Defect Classification on NEU-CLS

As mentioned above, MFN can be merged into baseline CNNs for defect classification tasks. Therefore, we first

¹The NEU data set has been introduced in our previous work [3]. If you want to know the details about the data sets, visit the website: http://faculty.neu.edu.cn/yunhyan/NEU_surface_defect_database.html

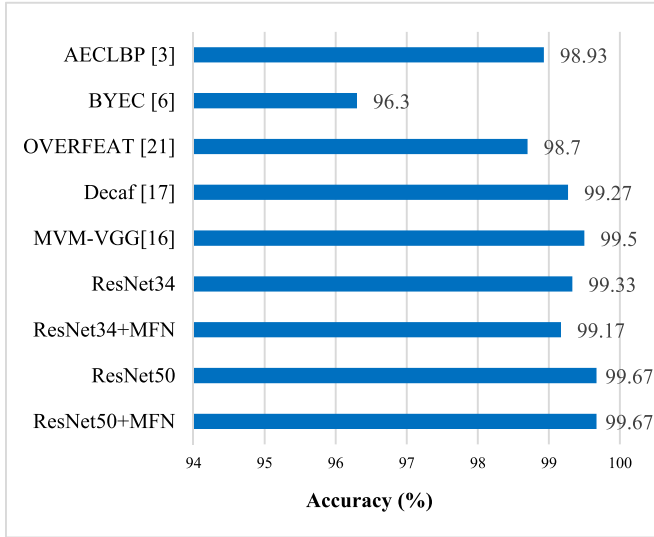


Fig. 6. Classification results on NEU-CLS data set.

report results on defect classification to demonstrate that our approach can achieve the competitive accuracy over other related methods, and merging MFN does not significantly affect the classification ability. Fig. 6 shows the defect classification results compared with other methods. According to Fig. 6, we can get the following conclusions.

- 1) The networks with MFN can perform well on defect classification so the multilevel features still have strongly semantical capability.
- 2) For ResNet34, MFN slightly harms the classification results. However, this influence is vanished for the deeper network ResNet50. It indicates that features extracted from deeper network are more distinctive hence the entire network becomes more robust.
- 3) With MFN, the ResNet34 obtains 99% of the accuracy of the ResNet50, which indicates that, in practice, a very deep network is not really required for defect classification task.

As we know, stronger performance on defect classification should be positively correlated with stronger performance on defect detection. A good classification result is the prerequisite for subsequent defect detection experiments.

C. Defect Detection on NEU-DET

We carry out defect detection experiments on NEU-DET data set. Conventionally, we divide the NEU-DET into training set and test set, and fix the training/testing split. The training set containing 1260 images used for fine-tuning the network introduced in Section IV-B, and the test set containing 540 images. We compare DDN with faster R-CNN and HyperNet [34] on the test set and both methods use the same baseline network (VGG16 [40]) mentioned in their papers. In addition, DDN and faster R-CNN are also experimented on ResNet34/50 due to the similar proposals generator. Unlike defect classification, only accuracy is not an appropriate performance measure in case of defect detection. Therefore,

we evaluate the results of detection experiments by average precision (AP), which is a good tradeoff between the two significant detection indexes: Precision and Recall. These indexes are defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (4)$$

$$\text{AP} = \frac{\text{Precision} + \text{Recall}}{2} \quad (5)$$

where TP, FP, and FN represent the number of true positives, false positives, and false negatives, respectively. The mean AP (mAP) is also calculated to evaluate the overall performance, which is the mean value of the AP of all the classes.

Table II shows the results of defect detection experiments. Under the baseline ResNet34/50, DDN achieves a mAP of 74.8/82.3, 4.6/4.4 higher than faster R-CNN. This result demonstrates that the proposals extracted from multilevel features are superior to the proposals extracted from single-level features. Under the same baseline network (VGG16), faster R-CNN achieves an mAP of 72.3 and HyperNet achieves an mAP of 74.8. DDN achieves an mAP of 76.6, 4.3 points higher than faster R-CNN and 1.8 points higher than HyperNet. HyperNet is also a detector based on the multiple features, but our method can extract higher quality region proposals, which will be discussed in Section VI in detail. The examples of detection results on NEU-DET are shown in Fig. 7.

Through the previous defect classification experiments, it is proven that MFN effects slightly on classification accuracy. Therefore, the improvement of mAP is benefited from the quality region proposals extracted from multilevel features. That means that MFN contributes to improve the localization accuracy. We specifically evaluate the performance of MFN in Section V-D.

D. Analysis on MFN

To verify MFN is able to improve the localization accuracy, we compare with several region proposal extractors, sliding window, Edge Boxes [35], and Selective Search [36]. In addition to these methods, RPN + MFN is also compared with the naive RPN (extract proposals based on single-level features). If the quality of proposals gets improved, the detector can use fewer proposals and stricter IOU thresholds without harming recall. Therefore, we evaluate recall on NEU-DET test set with different numbers of proposals and IOU thresholds. The number of proposals is the top-K ranked region proposals selected by these methods. IOU denotes a ratio between intersection and union of the predicted boxes and the groundtruth boxes.

Fig. 8 shows the defect recall with various IOU thresholds at three different numbers of region proposals. The larger the IOU threshold, the more quality the selecting proposals. Unsurprisingly, the performance of the methods based on convolutional features is strongly higher than the methods without CNN [37]. When $\text{IOU} > 0.7$, the recall of naive RPN drops sharply compared with RPN + MFN. The naive RPN only extracts proposals from high-level features and some location information is filtered by the preceding

TABLE II
DETECTION RESULTS ON NEU-DET

Method	Network	mAP	crazing	inclusion	patches	pitted surface	rolled-in scale	scratches
FRCN	VGG16	72.3	42.9	67.9	84.9	79.1	68.8	89.9
HyperNet	VGG16	74.8	54.1	68.0	86.5	87.0	65.2	88.1
DDN	VGG16	76.6	50.8	71.2	90.7	88.5	69.0	89.3
FRCN	ResNet34	70.2	46.7	61.3	82.8	76.5	70.7	83.4
FRCN	ResNet50	77.9	52.5	76.5	89.0	84.7	74.4	90.3
DDN	ResNet34	74.8	48.0	75.9	87.4	78.3	68.4	90.8
DDN	ResNet50	82.3	62.4	84.7	90.7	89.7	76.3	90.1

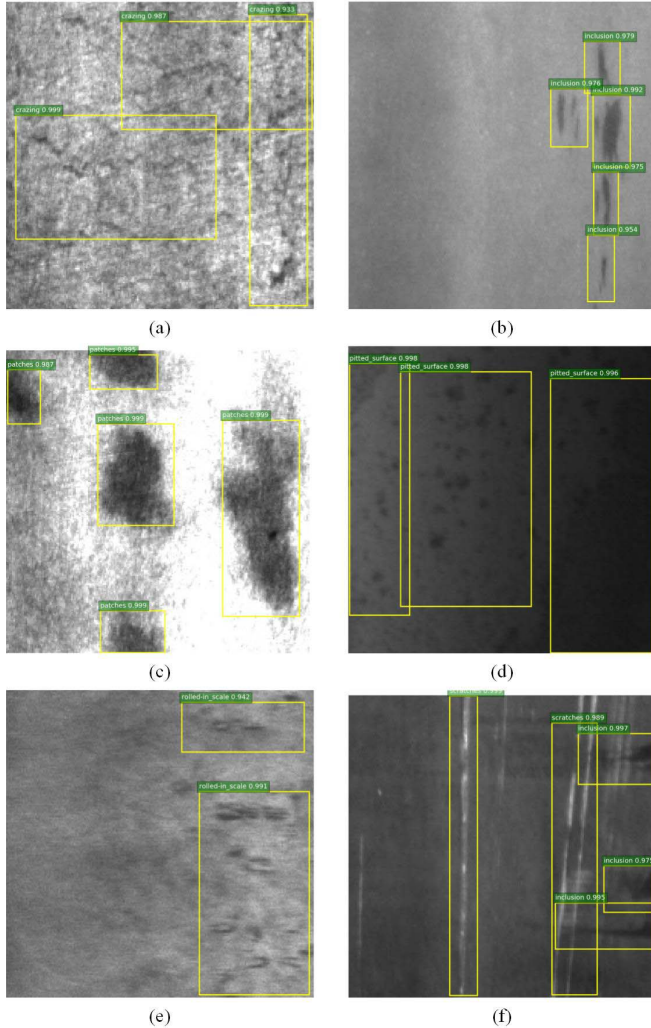


Fig. 7. Examples of detection results on NEU-DET. For each defect, the yellow box is the bounding box indicating its location and the green label is the class score. The subset to which the image belongs (a) crazing, (b) inclusion, (c) patches, (d) pitted surface, (e) rolled-in scale, and (f) scratches.

layers making the decline of proposals in quality. With the increasing number of proposals, the naive RPN drops more sharply when $\text{IOU} > 0.7$. This is because RPN extract too many low-quality proposals and it is more obvious with the increase of proposals. The naive RPN works badly with the strict IOU threshold (e.g., $\text{IOU} > 0.7$). MFN can help RPN to obtain location information from low-level and mid-level features, which makes RPN is under a higher tolerance for strict IOU threshold.

TABLE III
COMBINING LAYERS IN DIFFERENT MANNERS

Combine layers from:				mAP:	
R2	R3	R4	R5	without L2	with L2
			✓	70.2	70.2
		✓	✓	69.2	72.9
	✓	✓	✓	67.0	72.9
✓	✓	✓	✓	59.9	73.3
✓	✓	✓	✓	58.9	74.8

Detection mAP of DDN+ResNet34 on NEU-DET. L2: use L2 Normalization described in Section III-B.

Increasing the number of proposals can get a promising recall, but this will greatly increase the runtime of the detection [38], and what is worse, low-quality proposals would be involved in the process of detection, leading to failure of defect detection in some cases. Therefore, a good detector should select as few proposals as possible and meanwhile a relatively strict IOU threshold. Fig. 9 shows the defect recalls with various numbers of proposals at three different IOU thresholds. The naive RPN achieves a desirable recall with top-300 proposals, but RPN + MFN only needs top-100 proposals to get a similar performance.

As shown in Fig. 10, for RPN + MFN with ResNet34, we achieve 92% of the performance of selecting 300 proposals by selecting only 50 proposals, which reduces the run time by half. We consider selecting top-50 proposals as a good tradeoff in practical defect detection task.

VI. DISCUSSION

In this section, to demonstrate our design is logical and advanced, we discuss several implicit factors that can influence on defect detection.

A. Combine Which Layers for MFN?

MFN combines features from various levels into a multilevel feature, which is effective for improving detection. In Section III-B, it is briefly discussed that what kind of layers should be combined. In DDN, we select four layers that are the last layers of R1, R2, R3, and R4. Therefore, whether other combination manners of these four layers may result in better performance. Therefore, we train DDN + ResNet34 in five different combination manners on NEU-DET data set. As shown in Table III, combining all the four layers outperform the other manners. It indicates that the multilevel feature is effective for improving the accuracy of detection.

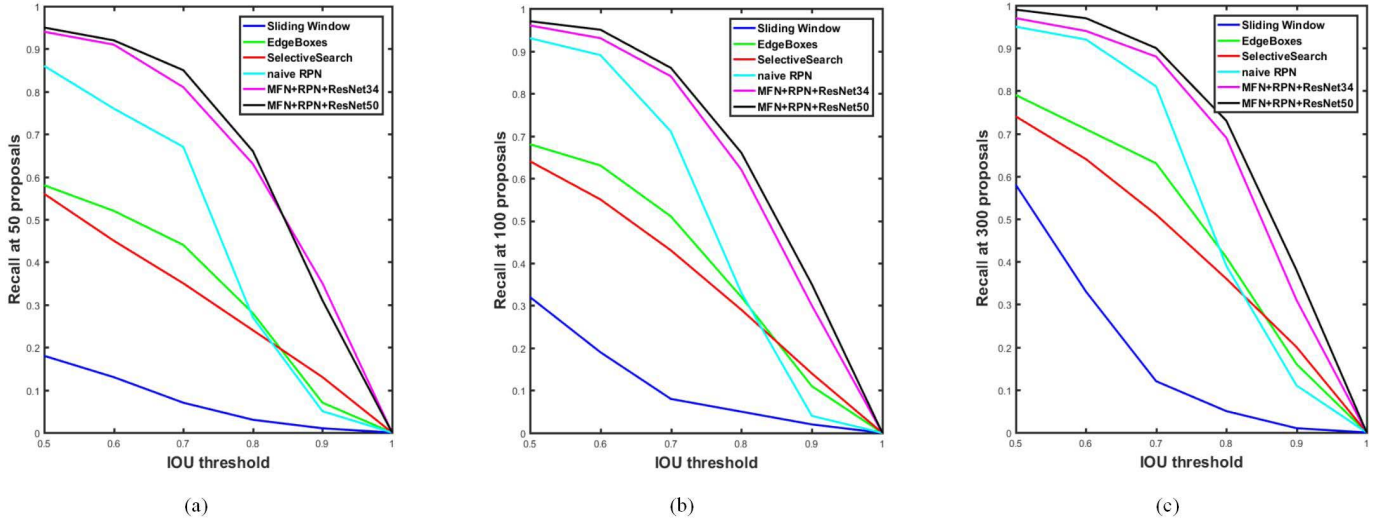


Fig. 8. Recall versus IOU threshold on the NEU-DET at different numbers of region proposals. (a) 50 region proposals. (b) 100 region proposals. (c) 300 region proposals.

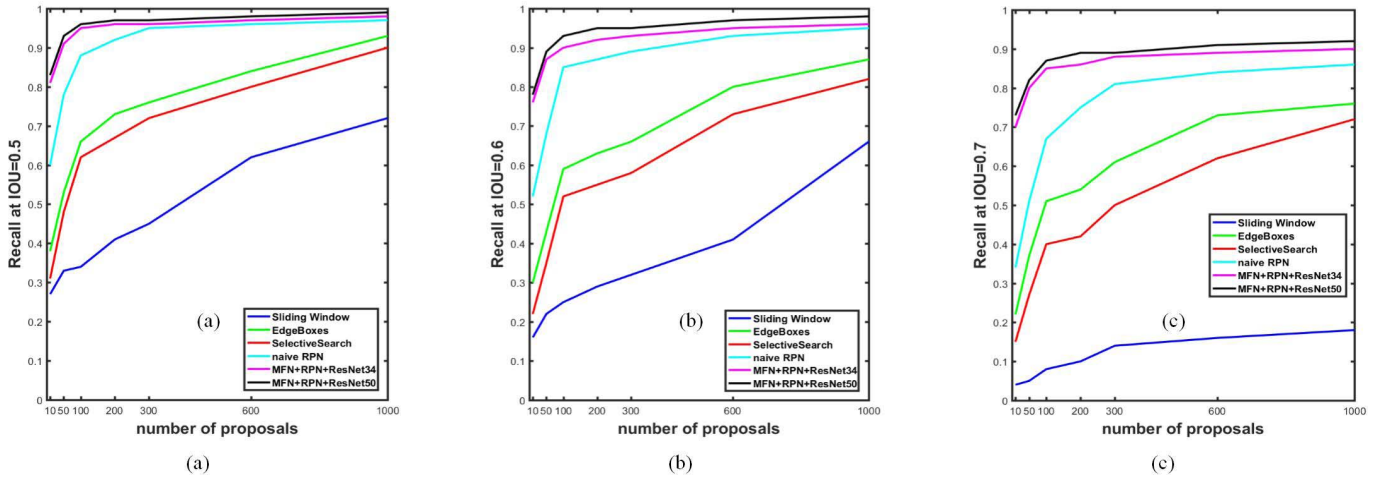


Fig. 9. Recall versus number of proposals on the NEU-DET at different IOU thresholds. (a) IOU threshold is 0.5. (b) IOU threshold is 0.6. (c) IOU threshold is 0.7.

Furthermore, low-level feature (e.g., R1 feature) should be paid more attention than high-level feature (e.g., R5 feature) for defect detection because R2 feature has richer location information than R5 feature.

B. Is the Simple Design More Effective for MFN?

The major role of MFN is to uniform the features from different levels in resolution and dimensionality. To keep the dimension consistent, a straightforward approach is using 1×1 conv to reduce/increase the dimensionality. There are two placement patterns for 1×1 conv: front-mounted and back-mounted. The front-mounted pattern means that 1×1 conv is placed before concatenating multilevel feature. What we use in this paper is the front-mounted pattern, that is, a 1×1 conv is placed at the end of each branch of MFN, and the back-mounted pattern means that a 1×1 conv is placed after concatenating multilevel feature. This pattern seems simple but in fact needs more parameters. Similar to [34], we use

multiple 5×5 convs to uniform the resolution and dimensionality simultaneously. However, the 5×5 conv is an expensive operation, which has the same effect as the double stacked 3×3 conv but requiring additional parameters. Table IV shows the comparable results among three patterns in detail. The front-mounted style uses three times fewer parameters than the back-mounted, and five times fewer than hyperstyle. Therefore, MFN in the front-mounted style has less possibility to be overfitting. Moreover, in case of the same resolution size, MFN features can preserve more complete information due to its larger dimensionality than Hyper feature's (512 vs. 126).

C. Do We Need More Defect Data?

As we known, an object detector can improve performance with more training data [39]. Therefore, whether this rule is also effective for industrial defect data? In order to make clear this problem, we train the DDN on not only the complete NEU-DET data set but also each subset separately. As shown

the ability to handle the overlapped defects and the success case is shown in Fig. 7(f). We guess the reason is that the “inclusion” and the “patches” in the figure are similar, and they influence each other when they are very close. For the “rolled-in scale,” the bounding box may ignore some edge defects shown in Fig. 12(d) due to such defects that are too scattered to define their scope. A more ideal defect detector is yet wanted because there is still room for improvement.

VII. CONCLUSION

In this paper, the DDN, a defect inspection system for steel plates is proposed. This system is a DL network that can obtain the specific category and detailed location of a defect by fusing the multilevel features. For defect detection tasks, our system can provide detailed and valuable indicators for quality assessment system, such as the quantity, category, complexity, and area of a defect. Furthermore, we set up a precious defect detection data set—NEU-DET. Experiments show that DDN can achieve 99.67% accuracy for defect classification task and 82.3 mAP for defect detection task. In addition, the system can run at a detection speed of 20 ft/s while keeping the mAP at 70.

In the future, we will focus on two directions as follows: the one is data augmentation technology due to the expensive manual annotations in detection data sets. The other is to perform the defect segmentation task with DL technologies, which can obtain a more precise defect boundary.

REFERENCES

- [1] D. Marr, *Vision: A Computational Investigation Into the Human Representation and Processing of Visual Information*. Cambridge, MA, USA: MIT, 2010, pp. 3–4.
- [2] D. A. Forsyth, *Computer Vision: A Modern Approach*. Upper Saddle River, NJ, USA: Prentice-Hall, 2002, pp. 482–539.
- [3] K. Song and Y. Yan, “A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects,” *Appl. Surf. Sci.*, vol. 285, pp. 858–864, Nov. 2013.
- [4] P. Caleb-Solly and J. E. Smith, “Adaptive surface inspection via interactive evolution,” *Image Vis. Comput.*, vol. 25, no. 7, pp. 1058–1072, Jul. 2007.
- [5] Y. Dong, D. Tao, X. Li, J. Ma, and J. Pu, “Texture classification and retrieval using shearlets and linear regression,” *IEEE Trans. Cybern.*, vol. 45, no. 3, pp. 358–369, Mar. 2015.
- [6] M. Xiao, M. Jiang, G. Li, L. Xie, and L. Yi, “An evolutionary classifier for steel surface defects with small sample set,” *EURASIP J. Image Vid. Process.*, vol. 2017, no. 48, pp. 1–13, Dec. 2017.
- [7] Y. Park and I. S. Kweon, “Ambiguous surface defect image classification of AMOLED displays in smartphones,” *IEEE Trans. Ind. Inform.*, vol. 12, no. 2, pp. 597–607, Apr. 2016.
- [8] M. Chu, J. Zhao, X. Liu, and R. Gong, “Multi-class classification for steel surface defects based on machine learning with quantile hyperspheres,” *Chemom. Intell. Lab. Syst.*, vol. 168, pp. 15–27, Sep. 2017.
- [9] S. Ghorai, A. Mukherjee, M. Gangadaran, and P. K. Dutta, “Automatic defect detection on hot-rolled flat steel products,” *IEEE Trans. Instrum. Meas.*, vol. 62, no. 3, pp. 612–621, Mar. 2013.
- [10] Q. Luo and Y. He, “A cost-effective and automatic surface defect inspection system for hot-rolled flat steel,” *Robot. Comput.-Integr. Manuf.*, vol. 38, pp. 16–30, Apr. 2016.
- [11] K. Liu, H. Wang, H. Chen, E. Qu, Y. Tian, and H. Sun, “Steel surface defect detection using a new Haar–weibull-variance model in unsupervised manner,” *IEEE Trans. Instrum. Meas.*, vol. 66, no. 10, pp. 2585–2596, Oct. 2017.
- [12] M. Chu, R. Gong, S. Gao, and J. Zhao, “Steel surface defects recognition based on multi-type statistical features and enhanced twin support vector machine,” *Chemom. Intell. Lab. Syst.*, vol. 171, pp. 140–150, Sep. 2017.
- [13] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask R-CNN,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 2980–2988.
- [14] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in *Proc. Neural Inf. Process. Syst. (NIPS)*, Montreal, QC, Canada, Dec. 2015, pp. 91–99.
- [15] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in *Proc. Neural Inf. Process. Syst. (NIPS)*, Montreal, QC, Canada, Dec. 2014, pp. 3320–3328.
- [16] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
- [17] R. Ren, T. Hung, and K. C. Tan, “A generic deep-learning-based approach for automated surface inspection,” *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 929–940, Mar. 2018.
- [18] Y. Li, G. Li, and M. Jiang, “An end-to-end steel strip surface defects recognition system based on convolutional neural networks,” *Steel Res. Int.*, vol. 88, no. 2, Feb. 2017, Art. no. 1600068.
- [19] S. Zhou, Y. Chen, and D. Zhang, “Classification of surface defects on steel sheet using convolutional neural networks,” *Mater. Technol.*, vol. 51, no. 1, pp. 123–131, Feb. 2017.
- [20] V. Natarajan, T.-Y. Hung, S. Vaikundam, and L.-T. Chia, “Convolutional networks for voting-based anomaly classification in metal surface inspection,” in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Toronto, ON, Canada, Mar. 2017, pp. 986–991.
- [21] P.-H. Chen and S.-S. Ho, “Is overfeat useful for image-based surface defect classification tasks?” in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Phoenix, AZ, USA, Sep. 2016, pp. 749–753.
- [22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *Proc. IEEE Comput. Vis. Pattern Recognit. (CVPR)*, Anchorage, AK, Jun. 2009, pp. 248–255.
- [23] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 770–778.
- [24] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “OverFeat: Integrated recognition, localization and detection using convolutional networks,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Banff, AB, Canada, Apr. 2014, pp. 1–16.
- [25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Neural Inf. Process. Syst. (NIPS)*, Las Vegas, NV, USA, Dec. 2012, vol. 60, no. 2, pp. 1097–1105.
- [26] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception architecture for computer vision,” in *Proc. IEEE Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 2818–2826.
- [27] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proc. IEEE Comput. Vis. Pattern Recognit. (CVPR)*, Columbus, OH, USA, Jun. 2015, pp. 3431–3440.
- [28] R. Girshick, “Fast R-CNN,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1440–1448.
- [29] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proc. IEEE Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788.
- [30] W. Liu *et al.*, “SSD: Single shot multibox detector,” in *Proc. Springer Euro. Conf. Comput. Vis. (ECCV)*, Amsterdam, Netherlands, Oct. 2016, pp. 21–37.
- [31] L. Zhang, Y. Gao, C. Hong, Y. Feng, J. Zhu, and D. Cai, “Feature correlation hypergraph: Exploiting high-order potentials for multimodal recognition,” *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1408–1419, Aug. 2014.
- [32] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [33] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, vol. 9, May 2010, pp. 249–256.
- [34] T. Kong, A. Yao, Y. Chen, and F. Sun, “HyperNet: Towards accurate region proposal generation and joint object detection,” in *Proc. IEEE Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 845–853.
- [35] C. L. Zitnick and P. Dollar, “Edge boxes: Locating object proposals from edges,” in *Proc. Springer Euro. Conf. Comput. Vis. (ECCV)*, Zurich, Switzerland, Oct. 2014, pp. 391–405.
- [36] J. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, “Selective search for object recognition,” *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Sep. 2013.
- [37] Y. Wei *et al.*, “Cross-modal retrieval with CNN visual features: A new baseline,” *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 449–460, Feb. 2017.

- [38] J. Hosang, R. Benenson, P. Dollár, and B. Schiele, "What makes for effective detection proposals?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 4, pp. 814–830, Apr. 2016.
- [39] X. Zhu, C. Vondrick, C. C. Fowlkes, and D. Ramanan, "Do we need more training data?," *Int. J. Comput. Vis.*, vol. 119, no. 1, pp. 76–92, Aug. 2016.
- [40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–16.



Yu He received the B.S. degree from the School of Mechanical Engineering and Automation, Liaoning Technical University, Fuxin, China, in 2014, and the M.S. degree from the School of Mechanical Engineering and Automation, Northeastern University, Shenyang, China, in 2016, where he is currently pursuing the Ph.D. degree.

His current research interests include deep learning, pattern recognition, and intelligent inspection.



Kechen Song received the B.S., M.S., and Ph.D. degrees from the School of Mechanical Engineering and Automation, Northeastern University, Shenyang, China, in 2009, 2011, and 2014, respectively.

Since 2014, he has been a Teacher with Northeastern University. His current research interests include vision-based inspection system for steel surface defects, surface topography, image processing, and pattern recognition.



Qinggang Meng received the B.S. and M.S. degrees from the School of Electronic Information Engineering, Tianjin University, Tianjin, China, and the Ph.D. degree in computer science from Aberystwyth University, Aberystwyth, U.K.

He is currently a Professor with the Department of Computer Science, Loughborough University, Loughborough, U.K. His current research interests include biologically and psychologically inspired learning algorithms and developmental robotics, service robotics, robot learning and adaptation, multi-unmanned aerial vehicle cooperation, drivers distraction detection, human motion analysis and activity recognition, activity pattern detection, pattern recognition, artificial intelligence, and computer vision.

Dr. Meng is a fellow of the Higher Education Academy, U.K.



Yunhui Yan received the B.S., M.S., and Ph.D. degrees from the School of Mechanical Engineering and Automation, Northeastern University, Shenyang, China, in 1981, 1985, and 1997, respectively.

Since 1982, he has been a Teacher with Northeastern University, and became as a Professor in 1997. From 1993 to 1994, he stayed as a visiting scholar at the Tohoku National Industrial Research Institute, Sendai, Japan. His current research interests include intelligent inspection, image processing, and pattern recognition.