ASAFPN: An End-to-End Defect Detector With Adaptive Spatial Attention Method

SenShan Ouyang\textsuperscript{1, a}, Yucheng Li\textsuperscript{1,2, b*, MingSheng Cao\textsuperscript{1,2, c}, Chen Chen\textsuperscript{3, d}}

\textsuperscript{1}Network and Data Security Key Laboratory of Sichuan Provence \\
\textsuperscript{2}University of electronic science and technology of China \\
\textsuperscript{3}Chengdu Aircraft Industrial (Group) Co., Ltd

\textsuperscript{a}email: oyss2008@163.com; \textsuperscript{b}email: 201922090431@std.uestc.edu.cn; \\
\textsuperscript{c}email: cms@uestc.edu.cn; \textsuperscript{d}email: sunnymoonchen@126.com

Abstract: Industrial defect inspection aims to detect unqualified products from those samples under detection, which plays a vital role in ensuring the quality of products. In this paper, an end-to-end defect detection network is proposed to locate the defect position of steel plate images and predict the defect category. In the detection network, we propose a novel feature pyramid network module, which is called the adaptive spatial attention feature pyramid. It can effectively fuse the texture features at low levels with semantic features at high levels. Based on the proposed module, persuasive experiments are carried out using different backbone networks. Experimental data shows that the feature pyramid module can effectively extract different levels of information and adaptively fuse multiple hierarchical features to improve the detection effect of the network.

1. Introduction

With the advent of industry 4.0 and intelligent manufacturing, industrial automation has become a future trend. And the quality of the product for the enterprise is its lifeline, so for unqualified product detection is of great importance. In industrial scene, defect inspection refers to finding out some "salient" samples that are different from the normal samples to be tested. For such tasks, we usually use unsupervised learning methods. However, due to the particularity of industrial scenes, there are clear indicators for the accuracy and recall rate of the algorithm, and the samples and annotations of industrial scenes are easier to obtain compared with other scenes, so the supervised method is more suitable for industrial scenes.

The main objective of industrial defect inspection is to find out the defective samples and classify the types of defects which are given to be tested. As mentioned above, most of the current defect detection methods mainly adopt unsupervised or semi-supervised methods, while this paper adopts supervised methods. Also, before the appearance of deep learning, most defect inspectors adopt the hand-craft features designed differently from one person to another. These methods greatly reduce the reliability and accuracy of the defect inspection algorithm. Besides, these methods can only classify image defect, but do not carry out fine-grained object analysis on images which means they can't effectively locate the specific defect locations and extract significant defect features from the sample images. This makes the results of sample classification lack conviction.

We define the task of defect inspection as follows. It is assumed that some sample images as \( \{x_1, x_2, \ldots, x_N\} \in X \) are existing while these images have their corresponding classification and...
annotations as \( \{y_1, y_2, \ldots, y_n\} \in Y \) where \( y_i = \{[c_{x_1}, c_{y_1}, h_i, w_i, c_i], \ldots, [c_{x_m}, c_{y_m}, h_m, w_m, c_m]\} \). In this formula, \( c_{x}, c_{y} \) represents the x-coordinate and y-coordinate of defect object center, \( h, w \) represents the height and width of the defect object, while \( c_i \) represents its defect type. Assuming that the model we designed is \( M \), image \( x_i \) can generate predicted defect results like \( \hat{y}_i = \{[\hat{c}_{x_1}, \hat{c}_{y_1}, \hat{h}_i, \hat{w}_i, c_i], \ldots, [\hat{c}_{x_j}, \hat{c}_{y_j}, \hat{h}_j, \hat{w}_j, c_j]\} \) through the model \( M \). If the matching degree between \( y_i \) and \( \hat{y}_i \) is greater than the preset threshold, it means the model can correctly detect the defect, otherwise, it cannot. In practice, the matching degree is generally measured by IoU (Intersection-over-Union) index as below in which \( A, B \) represent the defect object bounding box of prediction and ground truth.

\[
\text{IoU} = \frac{A \cap B}{A \cup B} \tag{1}
\]

In this paper, we adopted Faster RCNN[20] as the baseline model. Since its emergence in 2015, this method has achieved amazing results in many fields, and numerous papers have been published to modify the original method. The current method has been able to achieve competitive results in multiple object detection data sets. To improve the accuracy of the defect detector, we propose a novel feature fusion method, which is called Adaptive Spatial Attention FPN. It can effectively fuse different scale features of multi-levels, including texture features of pixel-level and semantic features of cognitive level. Compared with traditional FPN[11], it can adaptively learn the feature weights between multi-level features. Moreover, at the spatial level, more semantic information is extracted and enriched for the features with larger activation values, so that features at different levels and positions can be effectively integrated. Although the number of parameters has increased, it is negligible compared to the network.

The contributions of this paper are listed as follows:

- A novel feature fusion network is proposed, which can fuse features in different levels and spaces adaptively and efficiently;
- The method of object detection is applied to the defect inspector, so that not only correct defects can be classified, but also the specific location can be located;
- Extensive experiments have been conducted in Part IV. Persuasive results show the proposed ASAFPN module can effectively improve the accuracy of the defect inspector.

2. Related work

2.1 Defect Inspection

Automatic defect inspection is mainly used to ensure the quality of products which can be divided into two types. One is simply to classify images and identify whether there are defects in the images. Another requires not only the classification but also the location of defects. In the past, this process was usually done by numerous people, which required the identification of workers[3][10], and was obviously time-consuming. To make matters worse, the quality of the tested product could not be guaranteed. To ensure the quality of the product and reduce meaningless duplication of work, some automated methods have emerged, such as [1][2]. However, we observe that manually designed features are all used by these methods, which can have a significant impact on the success of the defect inspector and are greatly affected by human subjectivity. Later, some more intelligent methods appear, such as [3][5], which can only simply classify the defect or carry out incomplete detection. These methods cannot precisely locate the defect which is not conducive to further analysis of the defect causes.

Since the emergence of deep learning, SOTA results have been achieved in many fields, such as computer vision, natural language processing, recommendation system, etc. Usually, deep learning is stacked by perceptron with multiple layers. In computer vision, distinguishable representational features are extracted through a deep convolutional network from images, which usually contain some abstract semantic information. Through these distinguishable features, some challenging tasks can be completed such as object detection[20], pedestrian reidentification[16] and etc. At present, many deep learning methods[8][9] have been applied to automatic defect inspection. They usually use convolutional neural networks as feature extractors and then use dense neural networks as classifiers. However, due to a lack
of data set, these methods can only do the classification task in defect inspection, which seems too simple. Different from the methods mentioned above, our method can not only classify the sample images but also accurately locate the specific defects. Moreover, compared with other object detection algorithms, our method has a better performance. It is supposed to be pointed out that methods based on deep learning require a relatively large amount of computation, and some simple tasks[6][7] based on edge devices don’t require deep learning.

2.2 Feature Fusion Method
Feature fusion aims to fuse multi-level features with different semantic scales, which is mainly used to solve the problem of small objects in the detection task at begin. The earliest feature fusion network is Feature Pyramid Network[11], whose main step is to successively fuse features of different scales in accordance with the up-sampling, addition, and convolution flow. According to the experiments in the paper, this special network has indeed achieved better results in small object detection. Subsequently, different feature pyramid networks appeared successively, such as PANet[12], NASFPN[13]. On the basis of the former FPN, these methods further carry out an iterative fusion of features following the bottom-to-top order, by which better performance has been achieved in the object detection task.

Nevertheless, these methods don’t take the weight contribution of different features into account where the deeper features have richer semantic information compared with shallow features. The influence of spatial location information on the final feature is also neglected. Inspired by SAM[14], ASFF[15], we consider integrating the spatial information and weights contribution of multi-level features into a novel feature pyramid network called ASAFPN. It is worth mentioning that our network can be freely embedded into a variety of mainstream object detection networks to improve its accuracy with negligible parameters.

3. Method

3.1 Overview
The overall structure of the network is shown in Figure 1, which can be divided into five parts, namely backbone network, feature fusion module, regional proposal network, RoIAlign module and detection head. The backbone with the largest number of parameters contains most of the computational complexity of the whole network. The common backbones includes ResNet[17], DenseNet[18], EfficientNet[19] and

Figure 1. The overall network architecture includes a backbone network, ASAFPN module, regional proposal network, RoIAlign module, and detection head. Through the whole network, the defect location and classification can be marked on the image.
etc. In this paper, we adopt the ResNet series network as backbone networks to extract feature maps. Multiple bottlenecks constitute the ResNet. Each bottleneck has different channels and resolution of features according to the location in the network and automatically extracts multi-scale features. The backbone network structure used in experiments is shown in Table 1. The feature pyramid with adaptive spatial attention mechanism is proposed as the second part of the network, whose main function is to fuse the features of different levels including the texture information in the lower level and the semantic information in the higher. Previous methods usually only include feature fusion modules from top to bottom, such as FPN. The latter method does not take into account the feature weights between different levels, such as PANet, NASFPN, and BiFPN. For different detection tasks, corresponding weights should be given to different features, so that different features can be treated differently, to improve the distinctiveness between features.

Table 1. The backbone network of the object detector where the difference between $\text{conv}$ and $\text{conv}'$ is whether they had the activation function ReLU and $\text{identity}$ represents the skip connection.

|       | stem | layer1 | layer2 | layer3 | layer4 |
|-------|------|--------|--------|--------|--------|
| Res50 |      | $[7 \times 7 \text{conv} \atop \text{maxpool}]$ | $[1 \times 1 \text{conv} \atop 3 \times 3 \text{conv}]$ | $[1 \times 1 \text{conv} \atop 3 \times 3 \text{conv}]$ | $[1 \times 1 \text{conv} \atop 3 \times 3 \text{conv}]$ |
| Res10 | 1    | $[1 \times 1 \text{conv} \atop \text{identity}]$ | $[1 \times 1 \text{conv} \atop \text{identity}]$ | $[1 \times 1 \text{conv} \atop \text{identity}]$ | $[1 \times 1 \text{conv} \atop \text{identity}]$ |

The last three parts of the network are relatively classic, which are absolutely necessary for the two-stage object detector[20]. The RPN is mainly used for searching possible alternative bounding boxes in features after ASAFPN, which will generate 2000–3000 bounding boxes. After RoIAlign[21] module, around 300 bounding boxes with higher confidence of object will be chosen as our candidate boxes and cropped into feature maps with a fixed size, involved in the calculation of detecting head. After that, the detection head will further regression the generated feature to obtain more accurate position coordinates and classify the feature to obtain the category represented by the region of interest. Finally, the network will perform NMS operation on all generated results to filter out regions with high coincidence degrees and generate the final detection results.

3.2 Adaptive Spatial Attention Feature Pyramids Network

On the basis of FPN, the feature fusion module is reconstructed inspired by the idea in ASFF[15]. Multi-scale features at different levels are combined subtly to solve the nagging problem of inconsistency in the mesoscale. Meanwhile, we found that the spatial attention mechanism[14] can effectively improve the performance of the model with a negligible number of parameters. Therefore, based on the above two points and bottom-to-top idea, the ASA module is proposed aiming to solve the problem of multi-scale semantic feature alignment. First, the input features will be processed by FPN from top to bottom. Then output features will be sent to the ASA module for multi-scale bottom-to-top flow feature fusion. Finally, the topmost features of output will be sampled down to obtain the smallest feature.

The structure of the ASA module is shown in Figure 2. Through the backbone and feature pyramid network, four multi-scale features $A_i$ in different levels will be generated from the network. The adjacent features will be weighted and fused successively from the bottom to the top. For the problem of mismatching between input and output feature maps, we use a convolution operator with stride 2 to down sample the feature map. Then the maximum and average values of the feature maps on the channel dimension are obtained to concatenate into a feature with a channel number of 2. Subsequently, the spatial attention feature $G_i(A)$ can be obtained through the convolution operator changing the number of feature channels from 2 to 1. More details can be found in Formula (2).
5

\[ G^s(A) = \text{concat}\&\text{conv}\left( \frac{1}{C} \sum_{C}^{k=1} (A_{k,:,:}), \max_{C}^{k=1}(A_{k,:,:}) \right) \]  \hspace{1cm} (2)

Since the above process does not take into account the differences between multi-scale features, in order to integrate more semantic information, we have integrated spatial attention features at different levels into a single one. Through the softmax operator on channel dimension, the weights of different features were obtained. Finally, the output features can be get by a weighted sum between features with different scales. More details can be found in Formula (3).

\[ F^k_{i,j} = A^k_{i,j} \cdot G^s(A^k)_{i,j} + A^{k-1'}_{i,j} \cdot G^s(A^{k-1'})_{i,j} \]  \hspace{1cm} (3)

4. Experiments

4.1 Data Set and Hyper-parameter

In this experiment, the NEU-DET data set was used[1], which collected steel plate data with different defect types in real production scenarios. There are six categories in total, namely crazing, inclusion, patches, pitted surface, rolled-in scale, and scratches. The data set contains a total of 1800 images, 300 images in each category, which are divided into 1440 images as the training set, and 360 images as the test set. The subsequent experimental results are all carried out in test set.

We carried out the same control experiment in different backbone networks and adopted AP with the threshold of 0.5 as the performance index of the experiment. The intermediate layer output channel of ASAFPN is set to 256, and RoIAlign is used as the down sampling method of the detection head. The thresholds of positive and negative samples are set to 0.7 and 0.3 respectively, and the IoU threshold of NMS is set to 0.5. Cross entropy and L1 loss were used as classification and regression loss respectively. The optimization algorithm adopts the SGD algorithm.
4.2 Experimental Results on NEU-DET

Table 2. The experimental results with different backbone networks and feature fusion modules are presented, in which average precision with IoU 0.5 is adopted as the experimental index.

| Method    | backbone | crazing | inclusion | patches | pitted surface | rolled-in scale | scratches | mAP     |
|-----------|----------|---------|-----------|---------|----------------|-----------------|-----------|---------|
| FRCN      | Res50    | 0.576   | 0.743     | 0.859   | 0.784          | 0.506          | 0.891     | 0.727   |
| FRCN+FPN  |          | 0.605   | 0.808     | 0.850   | 0.829          | 0.501          | 0.885     | 0.747   |
| FRCN+PAFPN|          | 0.599   | **0.822** | 0.854   | **0.796**      | 0.545          | 0.893     | **0.752**|
| Ours      |          | 0.650   | 0.805     | **0.869**| 0.792          | **0.594**      | 0.890     | **0.767**|
| FRCN      | Res101   | 0.582   | 0.809     | 0.857   | 0.822          | 0.572          | 0.882     | 0.754   |
| FRCN+FPN  |          | 0.629   | 0.819     | 0.878   | 0.840          | 0.558          | 0.897     | 0.770   |
| FRCN+PAFPN|          | 0.605   | 0.821     | 0.868   | 0.841          | 0.594          | 0.897     | 0.771   |
| Ours      |          | **0.639**| **0.823** | **0.892**| **0.857**      | **0.627**      | 0.914     | **0.792**|

The backbone network will first load the pre-training model provided by Pytorch. Then the whole network will be trained on the training set with 24 epochs before performance evaluations on the test set. As shown in Table 1, a total of 8 experiments were carried out in this paper. And every 4 experiments were taken as a control group, with only the module of feature fusion changed. In this part, we used Res50 and Res101 networks as the feature extractor of the object detection algorithm, where two control groups of experiments were conducted respectively. In the first control group, Res50 was used as the backbone network. Among the six categories, crazing, patches, and rolled-in scale all achieved the best performance, where the rolled-in scale category had the best effect, exceeding the PANet method by 4.9%. The mAP index also achieved the best results, exceeding the other three methods by 4%, 2%, and 1.5%, respectively. In the control experiment with the backbone network of Res101, the best results were obtained for all categories of AP index. Compared with the other three methods, the mAP index exceeds 3.8%, 2.2%, and 2.1% respectively.

The contrast experiments above can effectively prove the superiority of the ASAFPN module in feature fusion and improve the accuracy of object detector. By carefully comparing the differences between the ASAFPN model and other feature fusion models, we can attribute the effect of improving accuracy to the spatial attention and multi-level feature weight fusion mechanism of the ASAFPN model. See Figure 3 for more model details.

5. Conclusion

In this paper, we propose an object detection method for steel plate defect inspection, which adopts the current mainstream two-stage object detection architecture. In order to improve the accuracy of the defect inspection algorithm, a novel ASAFPN feature fusion module is proposed. This module can adaptively extract the important part based on the activation value of the feature maps, which is called the spatial attention module. Moreover, in order to better integrate the information between different features, we attach adaptive weights to the features of different levels. Experiments show that this method can effectively improve the effect of the model on the NEU-DET data set, and the mAP index exceeds the traditional method by more than 2%, showing the excellence of this method.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (No.62072074, No.62076054, No.62027827, No.61902054), the Frontier Science and Technology Innovation Projects of National Key R&D Program (No.2019QY1405), the Sichuan Science and Technology Innovation Platform and Talent Plan (No.2020JDQ0020), and the Sichuan Science and Technology Support Plan (No. 2020YFSY0010).

References

[1] 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.
[2] 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.
[3] Z. Qin et al., ”Learning-Aided User Identification Using Smartphone Sensors for Smart Homes,” in IEEE Internet of Things Journal, vol. 6, no. 5, pp. 7760-7772, Oct. 2019, doi: 10.1109/JIOT.2019.2900862.
[4] 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.
[5] 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.
[6] Wang, Yilei, Tang, Y., Ma, J., & Qin, Z. (2015). Gender Prediction Based on Data Streams of Smartphone Applications. In Yu Wang, H. Xiong, S. Argamon, X. Li, & J. Li (Eds.), Big Data Computing and Communications (pp. 115–125). Cham: Springer International Publishing.
[7] Z. Qin et al., "Demographic information prediction based on smartphone application usage," 2014 International Conference on Smart Computing, Hong Kong, 2014, pp. 183-190, doi: 10.1109/SMARTECOMP.2014.7043857.
[8] 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.
[9] 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.
[10] Zhen Qin, Gu Huang, Hu Xiong, Zhiguang Qin, and Kim-Kwang Raymond Choo. A Fuzzy Authentication System Based on Neural Network Learning and Extreme Value Statistics. IEEE Transactions on Fuzzy Systems. 2019. DOI: 10.1109/TFUZZ.2019.2956896
[11] Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2117-2125.
[12] Liu S, Qi L, Qin H, et al. Path aggregation network for instance segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 8759-8768.
[13] Ghashi G, Lin T Y, Le Q V. Nas-fpn: Learning scalable feature pyramid architecture for object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2019: 7036-7045.
[14] Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 3-19.
[15] Liu S, Huang D, Wang Y. Learning spatial fusion for single-shot object detection[J]. arXiv preprint arXiv:1911.09516, 2019.
[16] Zhen Qin, Wei He, Fuhu Deng, Meng Li, and Yao Liu. SRPRID: Pedestrian Re-identification based on Super-resolution Images. IEEE Access, Volume 7, Issue 1, Dec. 2019: 152891-152899. https://doi.org/10.1109/ACCESS.2019.294926.
[17] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
[18] Huang G, Liu Z, Van Der Maaten L, et al. Densely connected convolutional networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 4700-4708.
[19] Tan M, Le Q V. Efficientnet: Rethinking model scaling for convolutional neural networks[J]. arXiv preprint arXiv:1905.11946, 2019.
[20] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 1 June 2017, doi: 10.1109/TPAMI.2016.2577031.
[21] He K, Gkioxari G, Dollár P, et al. Mask r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2961-2969.