IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 14, 2018, accepted January 4, 2019, date of publication January 23, 2019, date of current version February 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2894420

A High-Efficiency Fully Convolutional Networks for Pixel-Wise Surface Defect Detection

LINGTENG QIU[1], XIAOJUN WU[1,2], (Member, IEEE), AND ZHIYANG YU[1]
[1]School of Mechanical Engineering and Automation, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, China
[2]Shenzhen Key Laboratory for Advanced Motion Control and Modern Automation Equipment, Shenzhen 518055, China

Corresponding author: Xiaojun Wu (wuxj@hit.edu.cn)

This work was supported in part by the Basic Research Key Project of Shenzhen Science and Technology Plan under Grant JCYJ20170413152535587 and Grant JCYJ20150928162432701, and in part by the Basic Research Project of Shenzhen Science and Technology Plan under Grant JCYJ20170307151848226.

ABSTRACT In this paper, we propose a highly efficient deep learning-based method for pixel-wise surface defect segmentation algorithm in machine vision. Our method is composed of a segmentation stage (stage 1), a detection stage (stage 2), and a matting stage (stage 3). In the segmentation stage, a lightweight fully convolutional network (FCN) is employed to make a pixel-wise prediction of the defect areas. Those predicted defect areas act as the initialization of stage 2, guiding the process of detection to correct the improper segmentation. In the matting stage, a guided filter is utilized to refine the contour of the defect area to reflect the real abnormal region. Besides that, aiming to achieve the tradeoff between efficiency and accuracy, and simultaneously we use depthwise&pointwise convolution layer, strided depthwise convolution layer, and upsample depthwise convolution layer to replace the standard convolution layer, pooling layer, and deconvolution layer, respectively. We validate our findings by analyzing the performance obtained on the dataset of DAGM 2007.

INDEX TERMS Depthwise convolution, fully convolutional networks, surface defect segmentation, machine vision.

I. INTRODUCTION

Surface defect detection, which assesses the quality of product, acts as an important role in industry. With the development of computer vision, automated computer visual inspection is currently the main form that improves the industrial automation level significantly. However, most of the state-of-the-art computer vision based defect detection algorithms are knowledge-based approaches [1] that involve feature extraction and classification. The feature extractor is commonly well designed manually by experienced algorithm engineers case-by-case which makes the development cycle relative complex and time consuming. Besides that, such methods can hardly be generalized or reused. Therefore, the goal of this paper is to design a general, pixel-wise and highly efficient algorithm framework for surface defect segmentation in product quality control. The algorithm is based on fully convolutional networks [2], which can be trained end-to-end, pixel-to-pixel without any prior knowledge for pre-/post-process or case-by-case designing. We choose FCN as our base network for three reasons:

- FCN can take input of arbitrary size and produce corresponding output.
- Its inference and training can be efficient.
- It can be trained to make dense predictions.

The third reason is more crucial for our task which is proposed to make a per-pixel detection for defect area. Overall, the contributions of this study are mainly in three aspects:

- It novelly defines the problem of defect detection as a pixel-wise segmentation task. This definition makes the prediction of the defect area more accurate which is benefit for quality evaluation.
- It designs a cascaded framework in which we apply a detection stage to correct the segmentation results and use guided filter to refine the segmentation result. The output of every former stage acts as the initialization of the later stage. This framework is less time-consuming and can make a dense prediction for the defect area.
- It comes up with a strategy to reduce the consumption of our algorithm. We use depthwise&pointwise convolution, strided convolution and bilinear upsample convolution to replace the standard convolution layer, pooling layer and deconvolution layer, respectively.

The paper is organized as follows: section II reviews related works of surface defect detection; section III

introduces our 3-stage FCN for surface defect segmentation; section IV introduces our strategy for reducing the consumption of our algorithm; section V validates the proposed approach experimentally; section VI draws conclusion.

II. RELATED WORK

In recent years, many defect detection tasks can be solved by designing a set of features for a certain defect, then providing these features to a simple classifier, and these methods are also called the knowledge based approaches. Jian *et al.* [4] use IFCM(improved fuzzy c-means cluste) method which is based on histogram for segmentation of defect area on mobile phone screen glass. In [5], the defect detection is formulated as a novel voting procedure depending on an ideal lattice artificially generated by investigating the distributions of responses given by convolving lattices with Gabor filters. Feature extractor designing requires designers to have rich prior knowledge, and the challenge is that such methods can hardly be generalized or reused and may be inapplicable in real application [1]. One solution to this problem is to use deep learning method to extract features and classify each feature automatically by training. This approach is also known as representation learning. There have existed several algorithms for defect detection with the method of CNN [6]. Weimer *et al.* [7] propose a detection framework including a region of interest(ROI) extractor and a classifier. They just slide windows of a fixed size across the whole image to extract ROIs, and train a CNN to classify each ROI into defect patch or defect-free patch. This framework can only segment the defect areas in patch-wise, and sliding window densely across a high-resolution image is time-consuming. Xiao et al. train individual networks at different scales independently, and then combine networks of different scales into a new network [8]. The method of combination is just the concatenation of top-layer feature maps of different networks where computation is not amortized. This means the computation increases approximately linearly with the number of combined networks. Besides that, VIDI—a commercial software [9], is the first ready-to-use deep learning vision software dedicated to automated aesthetic detection and classification. Its red tool is used for defect detection. No detail information can be achieved about the underlying algorithm of this software, so we just assess this software from the detection results. More details will be found in section V.

III. THREE-STAGE FCN FOR SURFACE DEFECT SEGMENTATION

As we treat the issue of defect detection as a pixel-wise semantic segmentation task, we choose fully convolutional networks as our basic network structure, which has been demonstrated to outperform other approaches in image segmentation [10]. Ahead of the introduction of our algorithm framework, we propose three hypotheses based on the observation of the datasets to introduce our approach:

- Defect area is locally connected and isotropous on industry products.
- Defects of a certain type coming from the products of a certain type are independent identically distributed.
- Defect detection have little relation with the geometric shape of the whole defect area.

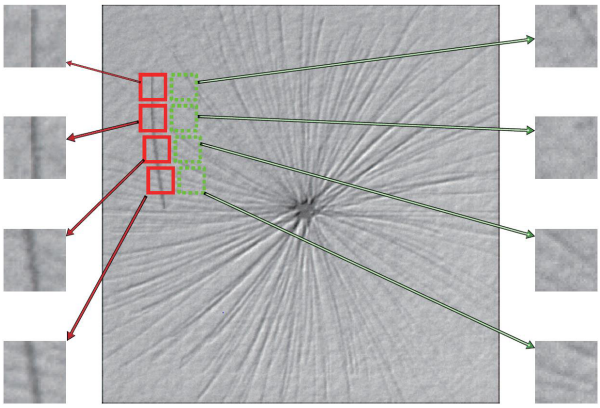


FIGURE 1. A schematic diagram of patches cropped from images. The red-solid-line rectangles illustrate patches cropped from defect areas and the green-dotted-line rectangles illustrate patches cropped from defect-free areas.

A. DATA AUGMENTATION

Paucity of defect data in industry is the crucial challenge for training a well-performance FCN. According to the 3rd observation we proposed above, we can determine the existence of a certain type of defect from a patch cropped from the defect area. For example, according to the patches cropped from a certain statistically textured surface image illustrated in Fig. 1, we can easily tell whether the patches have defect areas (red boxes) and which pixel belongs to defect area in each patch. Therefore, we can extend the dataset by cropping patches in the original images. Each patch can act as a training data instead of the whole image.

B. THE METHODOLOGY

We propose a cascaded framework for highly efficient defect segmentation, which combines a segmentation stage, a detection stage and a matting stage. The relationship between each stage is illustrated in Fig. 2. More details are explained in the following sections.

1) SEGMENTATION STAGE

The first stage (stage1 in Fig. 2) of our algorithm is to give a pixel-wise inference of defect areas with the help of FCN. As aforementioned, in the training stage, we divide the data into a training set and a test set. We sample patches from training set as input, and the label whose area in the patch exceeds the threshold T($T = 0.7$) is marked as a positive, vice versa. When testing, the entire image is taken as input. This strategy has special requirement which the receptive fields [11] should not be too large on our FCN architecture. If so, the information viewed by neurons located on the last layer from training data is similar as that from testing data.

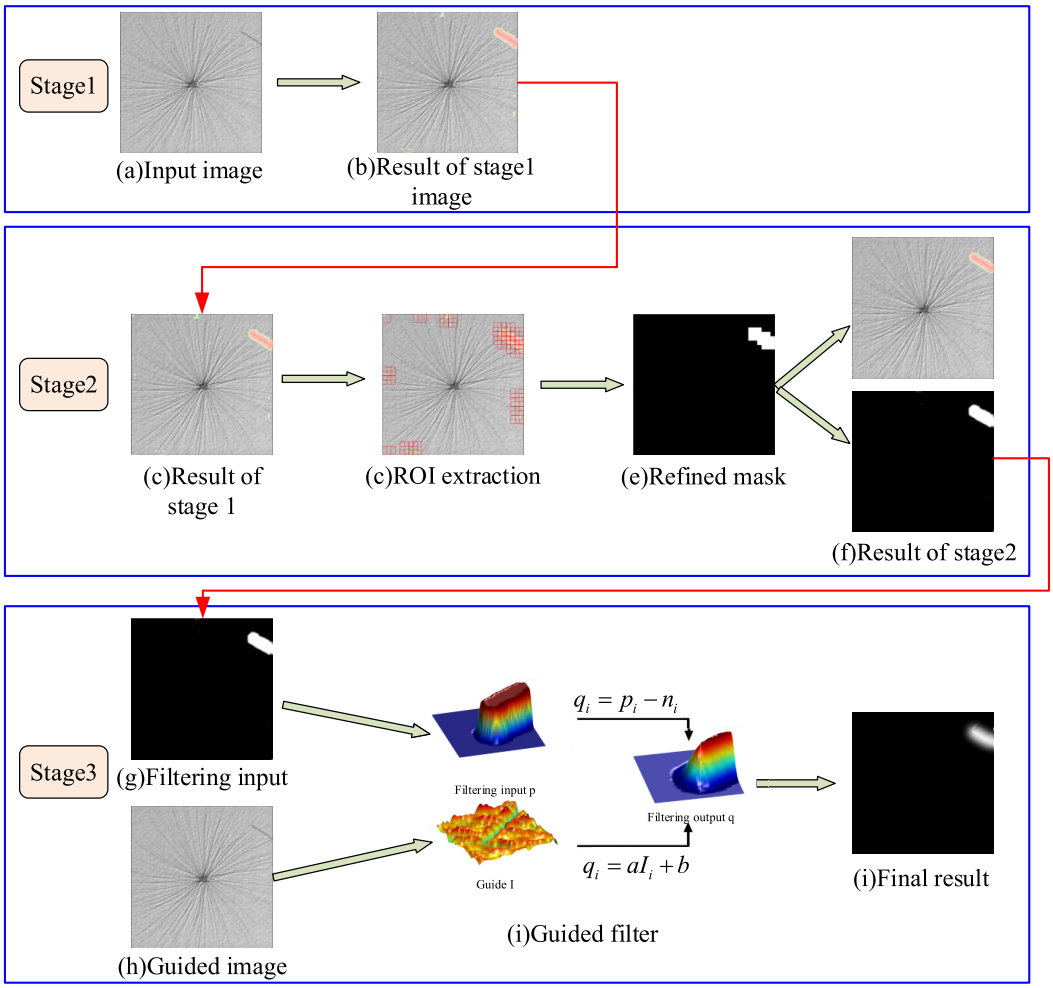


FIGURE 2. Algorithm framework. Stage 1 is a segmentation stage. stage 2 is a detection stage: (c) is the same as (b) which is the initialization of stage 2, (d) is the task of ROI extraction around the defect area predicted by stage 1, (e) is the refine mask predicted by the detection network, (f) is the corrected segmentation result. stage 3 is a matting stage: (g) Result of stage 2, (h) acts as the guided image, (i) is the algorithm of guided filter [3], (j) is the final result of the matting stage.

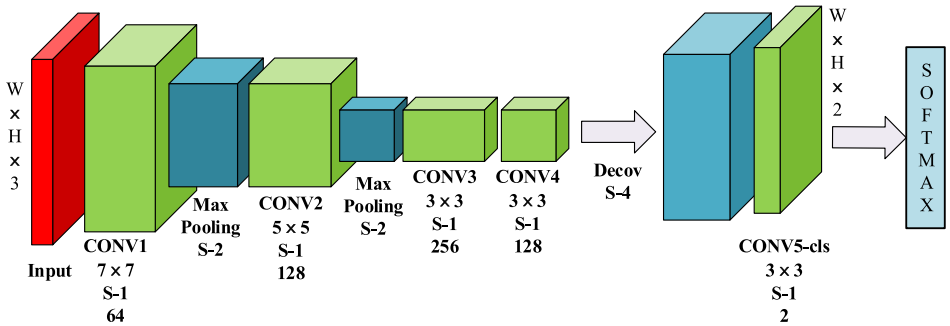


FIGURE 3. Structure of FCN in stage 1.(W×H×C×N) is on behalf of width, height, channels and number respectively. The size of label is the same as the input patch and the value on each label is the same as the corresponding mask annotated by human.

The receptive fields of our designed network is 32×32 pixels. To maintain the resolution of feature map used for classification, we insert a deconvolutional layer [12] before the score layer. We design 4 convolution layers with the stride of 1, and 2 pooling layers with the stride of 2. All convolution layers are followed by batch normalization (BN) and Relu. More details about the network structure are shown in Fig. 3.

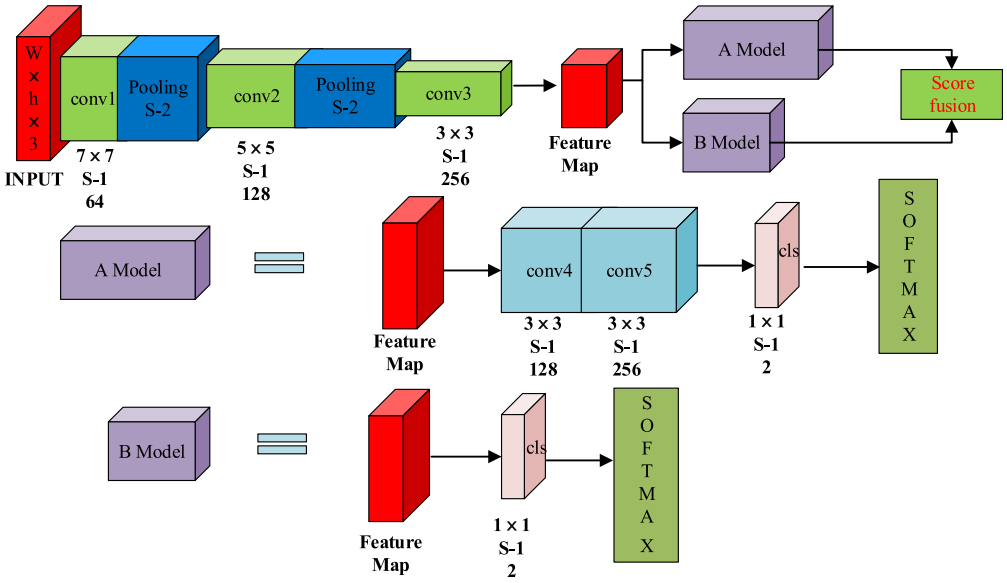


FIGURE 4. Multi-loss-function structure FCN in stage 2. A Model use an additional three convolution layers to extract richer semantic information for classification. B Model directly use the feature obtained by conv3.

2) DETECTION STAGE

This stage (stage 2 in Fig. 2) is designed to correct the improper segmentation of stage 1. The detection task can be divided into two subtasks:ROI extraction and ROI classification. From the stage 1, we sample the patches from training set which size is 32×32 in our experiment. We also sample patches the same as detection task using the sliding window sampling on predicted defect areas of first stage (shown in Fig. 2(d)). Finally, we generate a binary mask (shown in Fig. 2(e)). The corrected segmentation result (denoted as O) can be calculated by the segmentation result (denoted as S) and the binary mask (denoted as M) as below.

$$O = S \odot M \tag{1}$$

where $\odot$ denotes the Hadamard product. Multi-loss-function structure FCN in stage 2 is illustrated in Fig. 4: A is a deep branch which is designed to extract the global information, B is a shallow branch which is designed to extract local information, score 1 and score 2 are probabilities of defects predicted by each branch along the dimension of width and height.

The input to the network is a small patch on the ROI of a stage1. The feature extraction of the conv3 is obtained through the three-layer feature extraction. After conv3, we use A and B model to extract features. A Model uses an additional three convolution layers to extract richer semantic information for classification. For the B Model, we directly use the feature obtained by conv3 to classification, which is low-dimensional semantic information. Finally, we use the voting mechanism to obtain our final result. There are two methods to fuse two models. One is combining the results of the A and B model to get the maximum, and the other is to get the average. According to the experimental results, the average can reach the best metrics(MeanIU, AR, PA) in the test set. All detail is shown in Table. 1.

TABLE 1. Result of different fusion method.

Fusion Method	AR	MeanIU	PA
AB average	0.684322	0.732663	0.994454
AB max	0.672123	0.723951	0.994416
A	0.669614	0.718326	0.994407
B	0.665122	0.715809	0.994404

3) MATTING STAGE

This stage (shown in Fig. 2 stage 3) is used to refine the contour of a sort of defect with distinct edges aiming to make a more accurate prediction of defect area. Guided Filter [3] is adopted in this stage and we take original images as the guided images and the corrected segmentation result as the filtering input. Guided filter is a self-adapting algorithm, which is independent of training, so we can adopt it as the last stage of our algorithm without destroying its generalization or reusability. According to the experimental results, we set the local radius $\gamma = 3$, and $\epsilon = 1e-5$ which can get best result.

IV. ALGORITHM OPTIMIZATION

A. OVERVIEW

In practical machine vision applications, the speed of the algorithm is very important. It is related to the productivity efficiency. By increasing a little speed of the algorithm, productivity efficiency can be greatly improved. In order to achieve tradeoff between efficiency and accuracy, we optimize the FCN of stage 1 from the following three aspects:(1) transform the standard convolution into depthwise convolution combined with pointwise convolution; (2) change the

pooling layer into strided depthwise convolutional layer; (3) replace the deconvolution layer with bilinear upsample operation combined with depthwise convolution.

B. THE METHODOLOGY

1) CONVOLUTION MODULE

As mentioned in [13], we change all the standard convolutional layer in stage 1 into the depthwise convolutions combined with pointwise convolutions.

Suppose that the dimension of a tensor (denoted as F) is $D_F \times D_F \times M$, which is the input of a standard convolution module, and the dimension of the corresponding output (denoted as G)is $D_G \times D_G \times N$, where D_* means the resolution along the dimension of width and height and M and N are denoted as the number of channels of the input (output). The computation complexity of the standard convolution(denoted as S_0) is

$$S_0 = D_k \times D_k \times M \times N \times D_F \times D_F \times D_F \tag{2}$$

After the translation, the computation complexity of the corresponding depthwise&pointwise(denoted as S_1) is

$$S_1 = D_k \times D_k \times M \times D_F \times D_F \times D_F + M \times N \times D_F \times D_F \tag{3}$$

The computation complexity is

$$\frac{S_0}{S_1} = \frac{1}{N} + \frac{1}{D_k^2} \tag{4}$$

It means that the larger the number of output channel is, the more promotion will the proposed transformation bring. Besides that, as the transformation does not change the dimension of input and output, we can replace the standard convolution module with the proposed module directly. It will not influence the flow of information along other modules, which is illustrated in Fig. 5. We denote this module as "DWS-Conv".

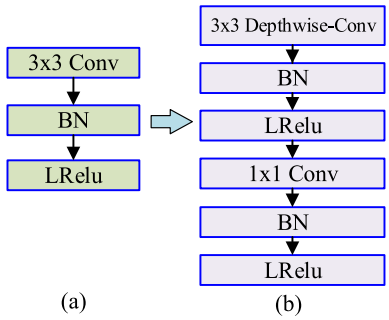


FIGURE 5. DWS-Conv:(a)standard convolution,(b)depthwise&pointwise convolution.

2) POOLING MODULE

Standard pooling module can be viewed as a special depthwise convolution module, whose weights is "one-hot". It means that only a certain weight in the local area is equal to "one", while others are equal to "zero", and the stride of pooling is equal or greater than 2, which gives rise to the reduction of the resolution of feature maps. But different from depthwise convolution, pooling module does not introduce any parameters. So, aiming to make up the parameters reduction raised by the transformed convolution module, we change all the standard pooling layer into depthwise convolution layer with the stride of 2, which is illustrated in Fig. 6. This transformation can bring more parameters without introduce any redundant computation. We cite this module as the "S-Conv" (short for strided depthwise convolution).

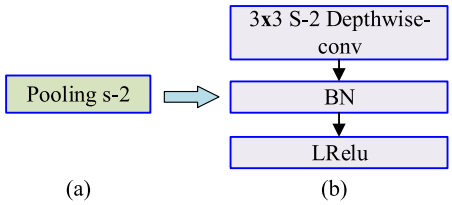


FIGURE 6. S-Conv: (a)Standard pooling module,(b)strided depthwise convolution.

3) DECONVOLUTION MODULE

As mentioned in [14], we use a bilinear interpolation to upsample the features and adopt a depthwise convolution to filter the enlarged feature map instead of deconvolution. It is illustrated in Fig. 7.

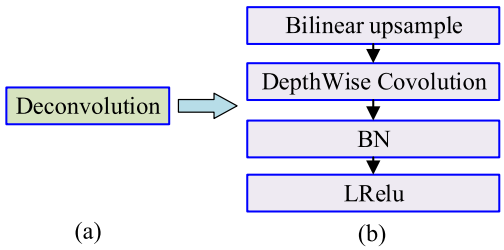


FIGURE 7. Up-Conv: (a)Standard deconvolution module, (b)bilinear upsample&depthwise convolution.

The computation cost of standard deconvolution is the same as related convolution, except that deconvolution can increase the resolution of the output(double at least):

$$S_{deconv} = D_k \times D_k \times M \times N \times D_F \times (D_F \times 2)^2 \tag{5}$$

We adopt a depthwise convolution behind the upsample option to reduce the noise from bilinear interpolation. The computation cost of the upsample&depthwise convolution is:

$$S_{up\&dw} = D_k \times D_k \times M \times D_F \times D_F \times 2)^2 \times 2 \tag{6}$$

and the computation complexity is:

$$\frac{S_{up\&dw}}{S_{deconv}} = \frac{2}{M} \tag{7}$$

So, the larger the number of input channel is, the more promotion will the proposed transformation bring. We cite this

module as the ''Up-Conv'' (short for upsample&depthwise convolution).

All in all, in order to get our optimization model we only need to change the corresponding layers of FCN(shown in Fig. 3) excpet conv5-cls layer in which we change Conv layer to DWS-Conv layer(Fig. 5),pooling layer to S-Conv layer (Fig. 6) and the decov layer to Up-Conv layer(Fig. 7) respectively.

V. EXPERIMENTS

We train and test our framework on the dataset of DAGM 2007 [15]. This dataset is artificially generated, but similar to real world problems. It contains 6 different types of defective dataset, and each development dataset consists of 150 defective images which are saved in grayscale 8-bit. Each dataset is generated by a different texture model and defect model. Therefore, for each texture model or defect model we only have 150 images for both training and testing which is too scarce to train a FCN for segmentation. So we extend the dataset as mentioned in section III-A. The develop environment is as follows: CPU: i5-6500, GPU: GTX1080, RAM: 16G, software environment: python 3.5 and tensorflow [16].

A. METRICS

For fear of one-sidedness, we use three metrics: Average Recall(AR) [17], Mean IU [2] and Pixel Accuracy (PA) [18] to evaluate the performance of our algorithm. These metrics are originally designed to evaluate the performance of general object segmentation. The ground truth is obtained from manual annotation and is used to train and validate our model. We expect that this framework can give detection results which are similar to the artificial detection results.

B. COMPARISON WITH OTHER ALGORITHM

To validate the effectiveness and superiority of our algorithm, we compare the proposed method with several off-the-shelf deep-learning-based methods. We train and test our algorithm on the dataset of DAGM 2007 and evaluate the performance by those three metrics mentioned above. We take ViDi (mentioned in section II) and three FCNs of different architecture for general object semantic segmentation as our contrast. The quantitative results are shown in Table. 2.

In Table. 2 and Fig. 8, stage1,stage2 and stage3 stand for different stage of the proposed framework respectively(segmentation and detection). And the segmentation stage is optimized with the method introduced in section IV. The AR(average recall) curve is shown in Fig. 8 that the curves of proposed method (three stages) are located on the upper right of other curves, which means the proposed method is preferable. It can be summarized from Table. 2 that under the judgement of three metrics, our proposed method outperforms the other four methods, and the performance of multi-stage method is a little better than that of the one-stage method. It means that the proposed method is effective and the multi-stage framework can improve the performance of the corresponding one-stage method.

TABLE 2. Overall performance of segmentation.

Algorithms	AR	MeanIU	PA
FCN(voc-fcn32s) [2]	0.439822	0.522574	0.992461
FCN(voc-fcn16s) [2]	0.620614	0.671290	0.993790
FCN(voc-fcn8s) [2]	0.606272	0.665455	0.993636
ViDi [9]	0.560456	0.619984	0.992371
Ours(stage1)	0.663561	0.713392	0.993402
Ours(stage2)	0.684300	0.732641	0.994418
Ours(stage3)	0.684322	0.732663	0.994454
Optim(stage1)	0.661344	0.713152	0.993398
Optim(stage2)	0.679135	0.729981	0.994322
Optim(stage3)	0.679144	0.730336	0.994354

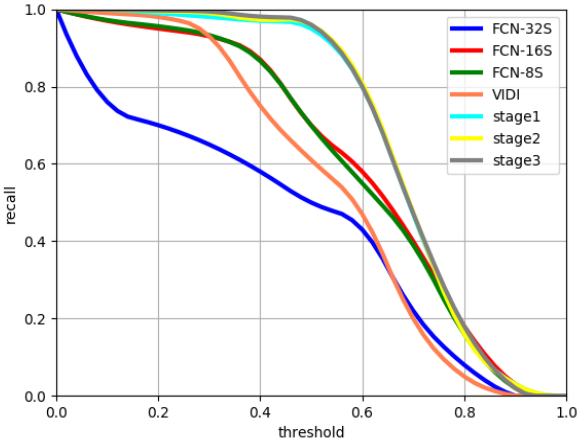


FIGURE 8. AR curve, it includes the results of three FCN structures [2], three stages of our proposed method and software VIDI [9].

In Fig. 9, we show a few qualitative segmentation results of proposed method and other methods. Fig. 9(a) is the initial input defected images, Fig. 9(b) is the manual-annotated ground truth, Fig. 9(c) illustrates results obtained by proposed method, Fig. 9(d) contains the results obtained by FCN [2] and Fig. 9(e) shows results of ViDi [9]. Compared with other results, our results are visually more consistent with ground truth. FCN [2] is also one of the state-of-the-art algorithm for generic object segmentation, however the related results are not so as satisfactory as ours, our analysis suggested that FCN [2] is used in a general-purpose environment which have a lot of data to train but our algorithm is specifically designed for small data sets. so, using fewer convolution layers to prevent overfitting can get better result. The slicing method(in section III-A) is a method for small data sets in industrial applications, which improves the accuracy of our algorithm. In addition, our stage2 in Fig. 4 uses CNN classification network to filter out the errors of segmentation, which also improves our detection accuracy.

C. EXPERIMENT OF THE EFFICIENCY OPTIMIZATION

In section IV we propose three modules to optimize the computational efficiency of our algorithm. To validate the benefits of optimization, we build two isobathymetric networks

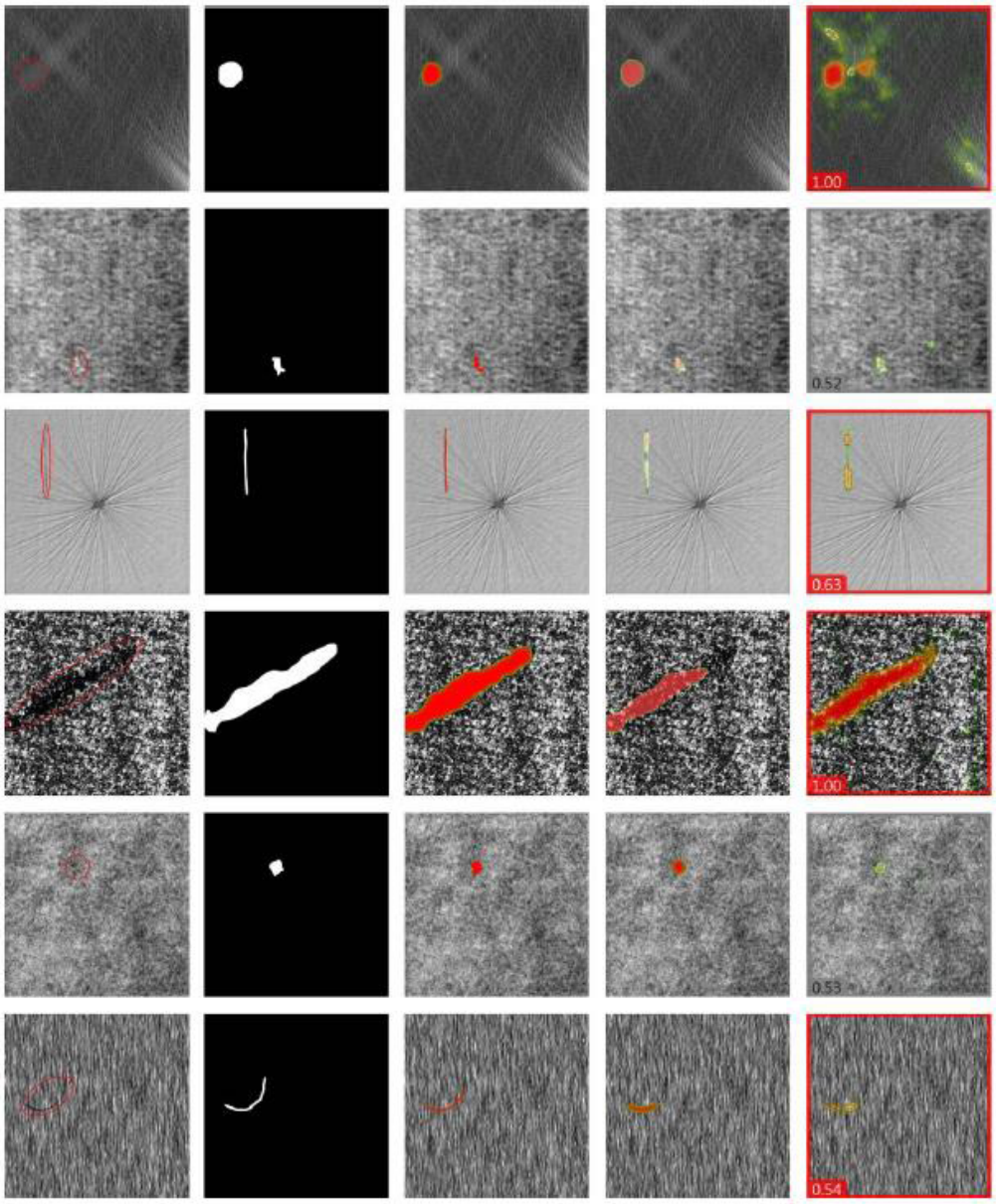


FIGURE 9. Results comparison: (a) is the initial input images, defect area is annotated by the red circle. (b) is the ground truth annotated per-pixel manually, (c) Results of ours(stage 3), (d)Results of FCN(voc-fcn8s) [2], (e)Results of ViDi [9].

TABLE 3. Computing resource consumed by each network.

–	Before optimization	After optimization	Decrease of relative consumption
Time consumption	44ms	36ms	18.18%
Power consumption	100W	85W	15%
Max resolution of input	2690×2690	2739×2739	–

constructed by normal modules and the proposed modules separately. And we calculate the expectation of computing resource consumed by each network. The quantitative results are shown in Table. 3.

It is obvious in Table. 3 that the optimized network relies on fewer computing resource, and is time-saving. For the input with the size of 512×512, the optimized algorithm can process 25 frame per second. As the algorithm is designed for industrial environment, where it will run in wide-range and full load, though the decrease of relative consumption is non-significant, considerable cost can be saved in the long run.

D. EXPERIMENT OF THE RECEPTIVE FIELDS OF STAGE 1

As mentioned in section III, the inconsistence between the training data and testing data requires that the receptive fields of the unit located on the last layer of FCN should not be too large. To validate it, we change the receptive fields of the FCN in Fig. 3 by changing the strides of conv1 and conv2 from 1 to 2. The testing result is shown in Fig. 10. Fig. 10(a) is the original input Fig. 10(b) is the ground truth and Fig. 10(c) is the segmentation result. It is obvious that the network has lose the ability to detect the defect areas in the testing images, if the receptive fields are enlarged blindly. So, in this training and testing strategy, the receptive fields of network should better not be too large.

FIGURE 10. Results of experiment: (a) is the initial input image with defect area. (b) is the ground truth annotated per-pixel manually, (c) is the result of stage 1 with the large-receptive-fields network.

E. EXPERIMENT OF THE MATTING STAGE

As mentioned in section III-B.3, we design a strategy to refine the contour of the segmented area with the method of guided image filter. The results are illustrated in Fig. 11.

Fig. 11(a) is original images of a type of contour-apparent defect, Fig. 11(b) is the coarse segmentation results obtained by stage 2. We treat (a) as the guided image and (b) as the filter input, after the guided filter processing and threshold processing, we get (c) and (d). Through the comparison of (b) and (d), it is obvious that (d) is more anastomotic with the real defect area. Guided filter can transfer the information of gradient of the guided image(Fig. 11(a)) into the filtering image(Fig. 11(b)), so it is reasonable to refine the contour of defect area with this method.

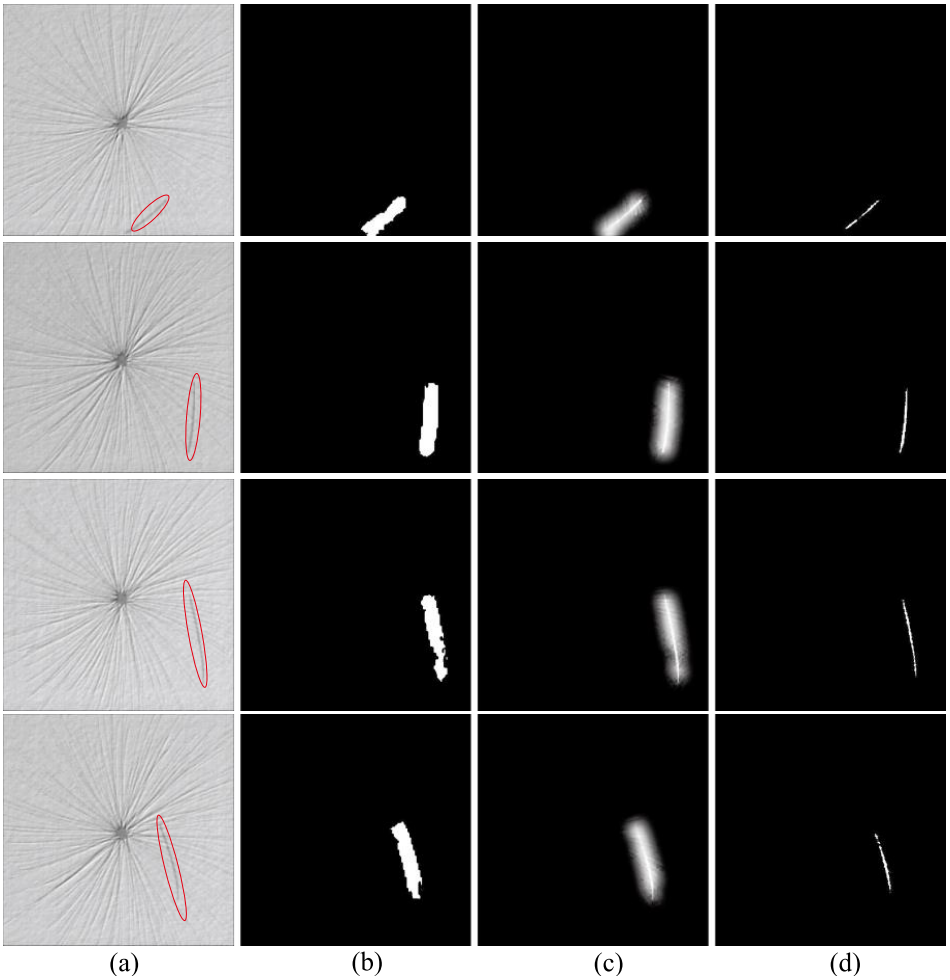


FIGURE 11. Results of refinement by guided image filter: (a) is the initial input images with defect area, (b) is the course segmentation results produced by stage 2 in Fig. 2, (c) is the results processed by guided filter(stage 3 in Fig. 2), (d) is the refined results after the threshold processing.

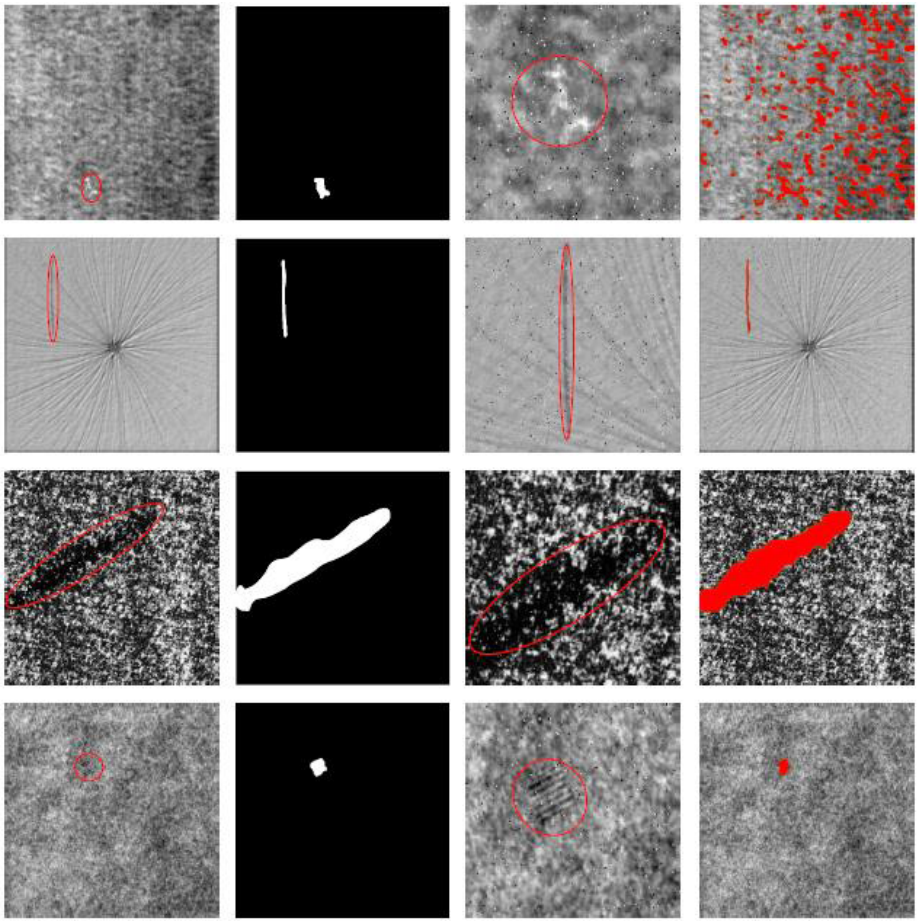


FIGURE 12. Experiments for robustness:(a) is the initial images, defect area is annotated by the red circle, (b) is the ground truth annotated per-pixel manually, (C) is the partially-enlarged defect area superposed by salt&pepper noise, (d) is the results produced by our method.

F. EXPERIMENT OF ROBUSTNESS

As industrial camera is the main equipment to capture images in industry, images can easily be disturbed by pulse signal which will superpose the captured images with salt&pepper noise intuitively, and it will do harm to the performance of algorithm. To validate the robustness of our algorithm, we superpose our test image with simulative salt&pepper noise which occupies 0.3% of the whole image. The experiment results are illustrated in Fig. 12. Fig. 12(a) illustrates the original images without noise, Fig. 12(b) is the ground truth, Fig. 12(c) is the image superposed by salt&pepper noise and Fig. 12(d) is the segmentation result. It can be concluded that our algorithm is robust for the most type of defect image except for the defect of dead pixel. Maybe the feature of dead pixel is too similar to the salt&pepper noise and the network for segmentation focus too much on local information that cause the degeneration about the defect of dead pixel. The quantitative results are shown in Table. 4.

It can be concluded from Table. 4 that although the performance declines slightly, the proposed algorithm still has the ability to segmentation the defect area. We can draw the conclusion that the proposed method has certain robustness against the disturbance of salt&pepper noise.

TABLE 4. Overall performance for robustness experiment.

Algorithms	AR	MeanIU	PA
stage1_none	0.681522	0.730193	0.993819
stage2_none	0.684300	0.732641	0.994418
stage1_noise	0.536226	0.603022	0.969071
stage2_noise	0.545024	0.610266	0.970354

VI. CONCLUSION

In this paper, we have presented a novel 3-stage FCN framework for pixel-wise surface defect segmentation in industrial environment. We design a combination of a segmentation task, a detection task and a matting task. Besides that, we propose a method to improve the efficiency of the algorithm. Our framework achieves meaningful results in

terms of performance and speed. We can process 25 defect images (with the size of 512 × 512) per second, and the pixel accuracy is more than 99%. As the proposed algorithm is based on the local information, it is weaker in detecting the structural defect than the texture defect. We leave this for future studies.

REFERENCES

[1] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, and P. Fieguth, "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure," *Adv. Eng. Inform.*, vol. 29, no. 2, pp. 196–210, Apr. 2015.

[2] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3431–3440.

[3] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, Jun. 2013.

[4] C. Jian, J. Gao, and Y. Ao, "Automatic surface defect detection for mobile phone screen glass based on machine vision," *Appl. Soft Comput.*, vol. 52, pp. 348–358, Mar. 2017.

[5] L. Jia, C. Chen, J. Liang, and Z. Hou, "Fabric defect inspection based on lattice segmentation and Gabor filtering," *Neurocomputing*, vol. 238, pp. 84–102, May 2017.

[6] 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.

[7] D. Weimer, H. Thamer, and B. Scholz-Reiter, "Learning defect classifiers for textured surfaces using neural networks and statistical feature representations," *Procedia CIRP*, vol. 7, pp. 347–352, May 2013.

[8] X. Bian, S. N. Lim, and N. Zhou, "Multiscale fully convolutional network with application to industrial inspection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2016, pp. 1–8.

[9] Cognex, (2017). *Vidi of Cognex*. [Online]. Available: http://www.cognex.cn/productstemplate.aspx?id=19178&langtype=2052

[10] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, Apr. 2016.

[11] T. Lindeberg, "A computational theory of visual receptive fields," *Biol. Cybern.*, vol. 107, no. 6, pp. 589–635, Dec. 2013.

[12] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2528–2535.

[13] A. G. Howard *et al.* "MobileNets: Efficient convolutional neural networks for mobile vision applications." Accessed: Oct. 5, 2017. [Online]. Available: https://arxiv.org/abs/1704.04861

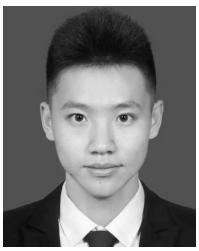
[14] Z. Wojna *et al.* "The devil is in the decoder." Accessed: Sep. 12, 2017. [Online]. Available: https://arxiv.org/abs/1707.05847

[15] (2007). *29th Annual Symposium of the German Association for Pattern Recognition*. Accessed: Sep.2, 2017. [Online]. Available: http://www.meta-net.eu/meta-system/kb/communication-and-events/events/conferences/obj_97914/view

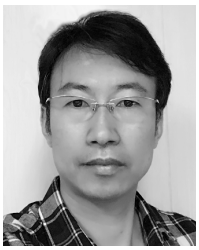
[16] M. Abadi *et al.* "Tensorflow: Large-scale machine learning on heterogeneous distributed systems." Accessed: Aug. 21, 2017. [Online]. available: https://arxiv.org/abs/1603.04467

[17] J. Dai, K. He, Y. Li, S. Ren, and J. Sun, "Instance-sensitive fully convolutional networks," in *Proc. Eur. Conf. Comput. Vis.* Basel, Switzerland: Springer, Sep. 2016, pp. 534–549.

[18] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, 2015.



LINGTENG QIU received the B.S. degree through the National Undergraduate Scholarship from the Institute of Electrical and Automation Engineering, Fuzhou University, Fuzhou, China, in 2017. He is currently pursuing the M.Sc. degree with the Harbin Institute of Technology, Shenzhen, China. His research interests include machine learning, pattern recognition, and computer vision.



XIAOJUN WU received the B.S. and M.S. degrees from Jilin University, Changchun, China, in 1994 and 1998, respectively, and the Ph.D. degree in mechatronics from the Shenyang Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2001. He is currently an Associate Professor with the Harbin Institute of Technology, Shenzhen, China. His current research interests include image processing, machine vision, deep learning-based defect detection, and image-based 3D modeling. He has received several best paper awards, or he was a finalist of the Best Paper Award from the IEEE ICIA, CAD&E, and ICARCV.

ZHIYANG YU received the B.S. and M.Sc. degrees in control science and technology from the Harbin Institute of Technology, Harbin, China, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree in computer science through the National Graduate Scholarship. His current research interests include image processing, pattern recognition, and computer vision.