Fully Convolutional Networks for Surface Defect Inspection in Industrial Environment

Paper ID: 37

Abstract. In this paper, we propose a reusable and high-efficiency two-stage deep learning based method for surface defect inspection in industrial environment. Aiming to achieve trade-offs between efficiency and accuracy simultaneously, our method makes a novel combination of a segmentation stage (stage1) and a detection stage (stage2), which are consisted of two fully convolutional networks (FCN) separately. In the segmentation stage we use a lightweight FCN to make a spatially dense pixel-wise prediction to inference the area of defect coarsely and quickly. Those predicted defect areas act as the initialization of stage2, guiding the process of detection to refine the segmentation results. We also use an unusual training strategy: training with the patches cropped from the images. Such strategy has greatly utility in industrial inspection where training data may be scarce. We further show that the receptive field of the top layer has crucial influence on this training strategy. This paper will validate our findings by analyzing the performance we obtained on the dataset of DAGM 2007.

Keywords: Fully Convolutional Networks, Surface Defect Inspection, Segmentation.

1 Introduction

Most of the state-of-the-art computer vision based defect inspection algorithms are pattern-based approaches [1] that involve feature extraction or modeling and classifier design (support vector machine, nearest neighbor classifier, etc.). Feature extraction is the crucial step. It describes the correlation between an input image containing millions of pixels and an information-rich feature vector, which can be used for classification stage [2]. The feature extractor is commonly well designed manually by experienced algorithm designer case-by-case.

A number of recent papers have made efforts in this step. Reference [3] uses Proposed IFCM method based on a histogram for segmentation of defect area on mobile phone screen glass. Reference [4] uses AGLR method to monitor grayscale images. Those 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 to the image conditions found in real runnel images [1]. The special pre- and post-processing and the case-by-case feature extractor-designing make the development cycle relative complex and time-consuming. Therefore, the goal of this paper is to design a general and reusable algorithm framework for surface defect in-

spection in industrial environment. The general and reusable algorithm aims to help us process various types of defect in a similar framework end-to-end without any prior knowledge for pre-/post-process or case-by-case designing. This framework can shorten the deployment cycles and make the development process more automatic and user friendly.

Deep learning, particularly Convolutional Neural Network (CNN), whose feature extractor and classifier can be trained end-to-end automatically from the input images themselves [5], can cover those shortages of the traditional manual approaches. Besides that, the highly nonlinear reflecting capability of CNN makes it very effective solving real problems of surface defect inspection, which are mostly highly nonlinear problems. However, due to the individual requirements for specific application in industrial environment, we should take accuracy, efficiency, exploitativeness and ease of use totally into consideration. In this paper, we introduce a two-stage Fully Convolutional Networks based algorithm framework to specialize the implementation of CNN for the issue of surface defect inspection in industrial environment.

The paper is organized as follows: §2 Reviews related works of CNN in various visual task, and reify the issue of surface defect inspection. §3 Introduces our two-stage FCN for the implement of industrial inspection. §4 Validates our approach experimentally. §5 Draws conclusion.

2 Related Work

Convolutional neutral networks have been proved effective end-to-end solution to various vision tasks such as detection [6], segmentation [7], super-resolution [8], tracking [9], etc. There have existed several algorithms for defect inspection with the method of CNN. Reference [10] trains individual networks at different scales independently, and then combines networks of different scales into a new network. The method of combination is just the concat 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 [11] developed by a Swiss company is the first ready-to-use deep learning vision software dedicated to automated aesthetic inspection and classification. Its red tool is used for anomaly detection. No detail information can be achieved about the underlying algorithm of this software, so we just assess this software from the inspection results. More details will be found in section 4.

3 Two-stage Fully Convolutional Networks

Convolutional networks have shown compelling quality and efficiency in various visual tasks. How to take full advantage of CNNs in our task depends on how we define the issue of surface defect inspection. The issue of defect inspection is more like a semantic segmentation task, which requires us making a prediction at every pixel. So, we choose fully convolutional networks as our basic network structure, which has been demonstrated to outperform other approaches in image segmentation.

The networks can thus be trained end-to-end, pixel-to-pixel, given the category-wise semantic segmentation annotation. Besides that, FCN has the property to allow arbitrary size of image as the input of the networks. This property facilitates the processing of images in different size.

Ahead of the introduction of our algorithm framework, we propose 4 principles based on the datasets from industrial environment to guide our approach: (1) In industrial environment, defect area is locally connected and isotropous; (2) Datasets of products coming from the same production line are independent identically distributed; (3) Local information or the patch of defect area is rich enough to represent the existence of defect area, and the inspection results have little relation with the geometric shape of the whole defect area; (4) The object-level information is also necessary to segment a defect area. Besides that, we also take computational efficiency and inspection speed in consideration.

3.1 Stage 1—Coarse Segmentation of Defect Area

This stage aims to giving a quick but coarse inference of defect area which is also called the region of interest (ROI). The predicted ROIs will be the initialization of stage 2 to limit the search range of stage 2. The ultimate goal is to improve inspection efficiency. However, using an off-the-shelf FCN like [7] in this task is not so proper, as the model designed for general segmentation is always deep, complex and computationally expensive in order to classify various objects of different classes, but the training dataset coming from industrial environment is scarce which may cause over-fitting. And the images of defect surface are quite different from that in nature, so the availability of annotated datasets such as ImageNet [12] can hardly be assumed in many applications, particularly industrial inspection, and the pre-training for classification and then transfer-learning the feature by fine-tuning from classification to segmentation in [7] may not work, as datasets for training classification is quite different from defect datasets. Therefore, we need to find a proper convolutional network structure and a proper training strategy to avoid over-fitting.

Dataset Extension. We choose the dataset of DAGM 2007 [13] as our development and validation dataset. The data is artificially generated, but similar to real world problems. The first six out of ten datasets, denoted as development datasets, are supposed to be used for algorithm. Each development dataset consists of 150 'defective' images saved in grayscale 8-bit PNG format. 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. Due to the (1), (3) principles we proposed in §3, the defect have the property that we can judge the existence of a certain type of defect from a patch cropped from the defect area. For example, for patches cropped from a certain statistically textured surface image illustrated in 错误!未找到引用源。, we can tell easily whether the patches have the defect areas and which pixel belongs to defect area in each patch.

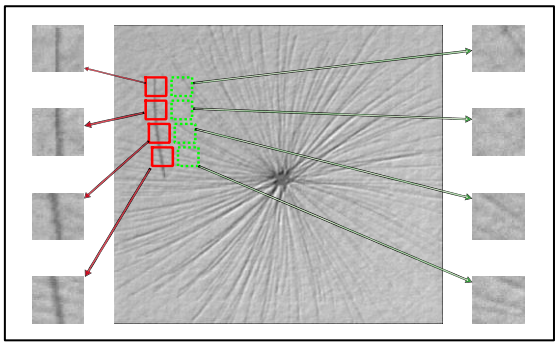


Fig. 1. A schematic diagram of patches cropped from images. The red-solid-line rectangles illustrate patches cropped from defect areas and the green-dotted-line rectangles illustrate patches cropped from non-defect areas

Therefore, we can extend the dataset by cropping patches in the original images. Each patch can act as a training data instead of only using the whole image as one training data. We design the size of each patch 32*32-pixel or 64*64-pixel, and crop in the original image with the stride of 16-pixel or 32-pixel. This task is just like computing convolution in an image. However, unlike a linear operation, it just copies the patches under the 32*32(64*64) window as the training sub-images. By this means, we can decompose 150 training images of one texture model or defect model into approximately 16000 training images.

FCN Architecture. As introduced in above section, in the training phase, we sample patches as the training data, however, in the testing phase we use the whole image as input. This training strategy has special requirement on our FCN architecture, especially the receptive field of the units located on the top-layer. Most related to our training strategy, Reference [8] also trains with "sub-images" and test with the whole image. Although the task of inspection, which makes a pixel-wise inference, is quite different from super-resolution where each output pixel is a real-number regressor of intensity values, the architecture design has something to learn from each other—the receptive field should not be too large. Receptive field is the "vision field" of each neuron located on the top layer; it is the area of connected pixels in the input image to an output neuron. If the receptive field is small, the network can focus on rich local spatial information rather than global object-level information. To interpret what influences the receptive field, we assume a network, the kernel size of the i-th layer (layer i) is K_i , s_i is the stride of layer i and S_i is the integral stride before layer i. We denote R_i as the receptive field of each neuron located on the i-th layer (noted as layer-i). Then the recurrence relation of R_i and S_i can be calculated as follows:

$$R_i = R_{i-1}+S_{i-1}(K_i-1) \tag{1}$$

$$S_i=s_i\times S_{i-1} \tag{2}$$

It can be concluded from the recursion formula that the receptive field is influenced by K_i , s_i and the depth of network layer-*i*. We pick Zeiler and Fergus model trained for RPN in [14] as our basic architecture. We use only the first 4 layers as our feature extractor layer and append a score layer at the end of feature layers, and we change the strides of all convolutional layers from s_i to 1 (s_i is the original stride of layer *i* in ZF). Overlap pooling used in RPN controls model capacity and increases receptive field size, resulting in a coarse, highly-semantic feature representation. While effective and necessary for extracting object-level information, this general architecture results in low resolution features that are invariant to pixel-level variations. This is beneficial for classification and identifying object instances but poses challenge for pixel-labeling tasks. So we change overlap pooling to non-overlap pooling as the former cause lager R_i in the following layers. The receptive field in our last layer is 35 pixels^2. To maintain the resolution of feature map used for classification, we insert a deconvolutional layer [15] before the score layer. We use logistic regression as the loss function for segmentation. More details about the network structure are shown in **错误!未找到引用源。**

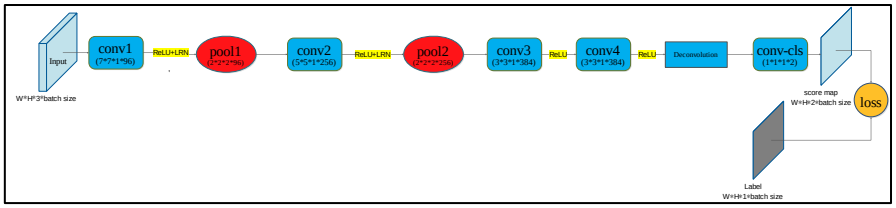


Fig. 2. Structure of FCN in stage 1. (W*H*C*N) is on behalf of width, height, channels and number respectively. The size of label is the same as the input patch and the value on each label is the same as the corresponding mask annotated by human.

3.2 Stage 2—Segmentation Refinement with Instance-Sensitive Patches

An Overview of Stage 2. This stage is to refine the segmentation result of stage 1 with a method of detection. In stage 1, we focus too much on local information. However, due to the 4th principle, the object-level information is also necessary to segment a defect area. That means "If this piece of area is a defect area or not" is also important information in defect inspection. Therefore, stage 2 is a detection task to refine stage 1 with object-level information. We still use the patches cropped in stage 1 as our training datasets, however, stage 2 is a task of detection not segmentation. Therefore, we should not use the manual annotated segmentation mask of the training data. In this paper, we label the patches whose defect area covers over 40% of the total area as the defect patches, and others as the non-defect ones. Therefore, stage 2 is to classify the patches cropped from the test images into binary class. We train a CNN model that outputs the defect patch probability of each input patch and select only patches with higher probability values as defect ones. It is obvious that the defect area is covered with a patchwork of all so-called defect patches. Reference [16] also uses a patch-based method. Quite different from their work, in this paper we do not

sample patches in the completely input image, we only get the samples around the ROI output by stage 1, which raises the inspection speed significantly.

The Fusion of Stage 1 and Stage 2. The segmentation result (ROI) is the initialization of stage 2. We crop patches around ROIs and classify each patch into 2 class—defect patch or not. Then we retain the intersection of the two stages. More details are shown in 错误!未找到引用源。

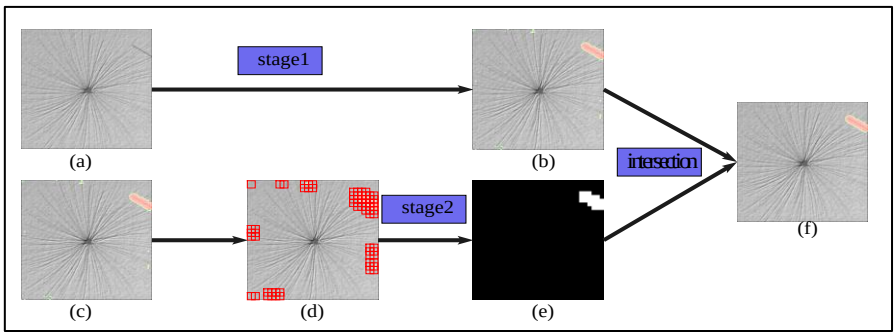


Fig. 3. Fusion of stage 1 and stage 2. (a) is the original image to be inspected, (b) is the coarse segmentation result of stage 1 and this result is also the initialization of stage 2(numbered as (c)), (d) shows the patches cropped around the ROIs from stage 1. (e) is the detection result of stage 2, each patch is classified into either defect patch or non-defect one. (f) is the intersection of stage 1 and stage 2, (f) is also the final inspection result.

The Structure of FCN in Stage 2. We also take ZF net as our basic network and transform it for our task. Rather than averaging independent outputs from multiscale networks, we design a multi-loss-function in one FCN to fuse information across layers to make a skip connection in order to increase the detection accuracy. All the loss function is logistic regression. As we still use FCN to finish a detection task, we label the patch with a label map that has the same resolution as output layer and its values are all the same—0 for defect patches and 1 for non-defect patches (illustrated in 错误!未找到引用源。).

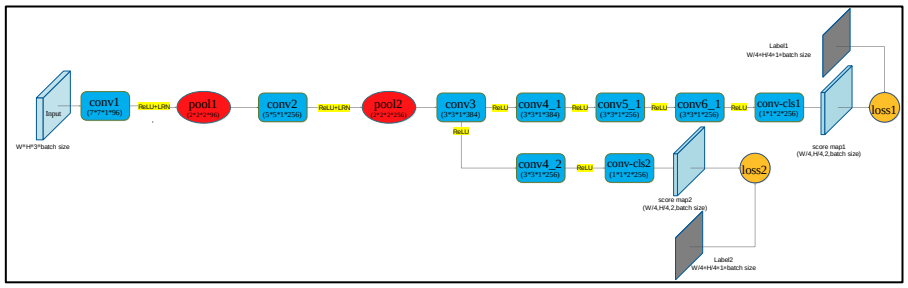


Fig. 4. Multi-loss-function structure of FCN in stage 2.ReLU is the activation function and LRN is short for local response normalization layer.

During the inspection process, similarly to [17] we vote the score map of one single softmax layer and then average the results from different softmax layers. This approach can be view as two merging propose: (1). Average the results of certain-size receptive fields under one patch. (2). Average the results of different-size receptive field under one patch. More details is illustrated in 错误!未找到引用源。. Besides that, hard negative mining method [18] is applied during the training phrase.

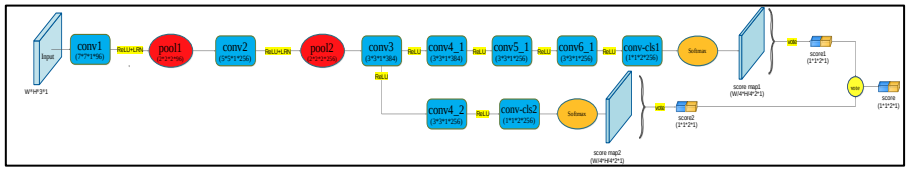


Fig. 5. Illustration of inspection process. Score1 and score2 is the mean value of each corresponding score map and score is the average number of score 1 and score 2.

4 Experiments

4.1 Experimental Framework

We train by SGD with momentum. We use a minibatch size of 128 patches and fixed learning rates of 10-3 for both stage 1 and stage 2. In stage 2, the ratio of training defect patches and non-defect patches is 4:1. We use momentum 0.9, weight decay of 5*10-4 and doubled learning rate for biases. We initialize the retained ZF's feature layer with the ZF's weights trained as a RPN (done in [14]) and randomly initialize the modified layer. Local Response Normalization is included where used in the RPN net. ReLu is used as activation function. We train and test our framework on the dataset of DAGM 2007 [13] and extension the dataset as mentioned in section 3.

4.2 Metrics

We use mean accuracy mentioned in [7] to evaluate the performance of our system. The ground truth is obtained from manual annotation and is used to train our model. This approach is a reasonable one as it is to train a human-like model in defect inspection. We expect that this framework can give an inspection results the same as artificial detection results. Therefore, we also use manual annotation ground truth to assess algorithm performance.

4.3 Results

We test our framework on the dataset of DAGM 2007 [13] which provides six types of defect area, each type contains 150 images for develop and test. We train the

framework with 120 images of each type of defect, and test the framework with the rest 30 images per type. Overall, 720 images are used for training and 180 for testing. The develop environment is as follows: CPU: i5-6500, GPU: GTX1080, memory: 16G, develop environment: matlab R2015a and caffe. It takes about 0.04s for stage1 and 0.05s for stage2 with the size of input image 512*512.

In 错误!未找到引用源。 we show a few qualitative segmentation results comparison of ours, and we show the overall performance in Table 1. While our results are visually more consistent with ground truth and achieve higher mean accuracy in quantitative analysis.

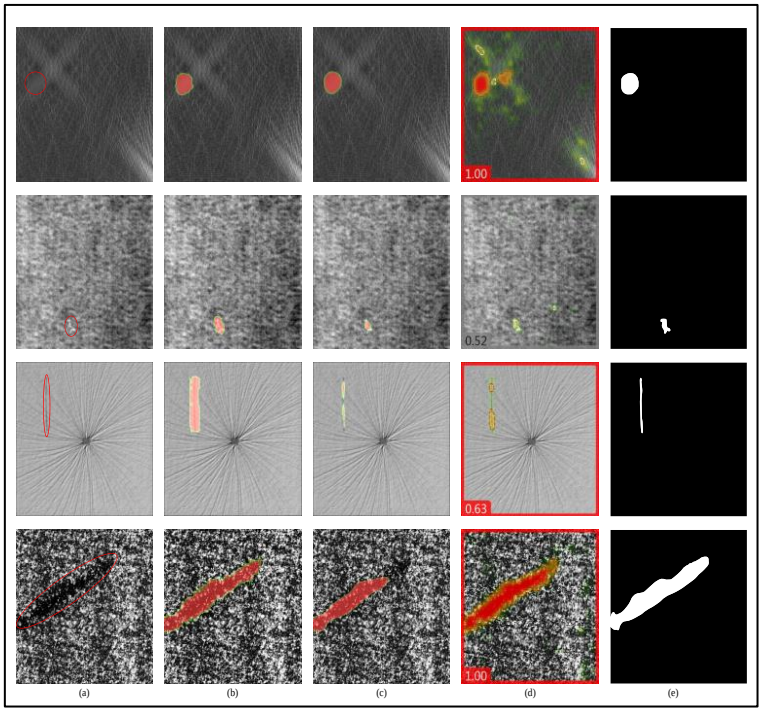


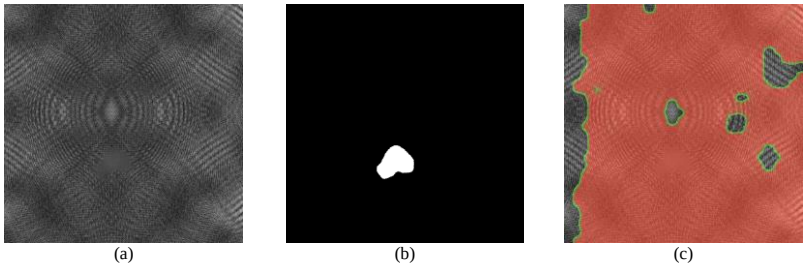
Fig. 6. Results comparison: (a) Initial input images. (b) Results of ours. (c) Results of FCN (d) Results of ViDi. (e) Ground truth.

Table 1. Overall Performance of Segmentation

Inspector	Mean accuracy	Time for inspection (512*512 input)
FCN(voc-fcn32s)[7]	79.3547%	71ms
FCN(voc-fcn16s)[7]	90.0371%	75ms
FCN(voc-fcn8s)[7]	92.2488%	78ms
ViDi[11]	93.7488%	20ms
Ours(stage 1)	95.9830%	48ms
Ours(stage 2)	95.9934%	20-50ms

4.4 Experiment of the Receptive Field of Stage 1

As mentioned in section 3 the inconsistence between the training data and testing data requires the receptive fields of the unit located on the last layer of FCN should not be too large. In this experiment, we change the receptive field of the FCN in figure 2 by changing the strides of conv1 and conv2 from 1 to 2. The testing result is shown as follows (**错误!未找到引用源。**). It is obvious that, in this training and testing strategy, the receptive field of network should better not be too large.



(a) (b) (c)

Fig. 7. Results of experiment: (a) Initial input image. (b) Ground truth. (c) Results of stage 1.

In another CNN based surface defect detection paper, we utilize the sliding windows based scheme to locate the defect position. Although the detection rate is higher than the method in this paper, it uses blocks to denote the position of the defects and cannot achieve the pixel-wise segmentation accuracy.

5 Conclusion

In this paper, we have presented a novel 2-stage FCN framework for surface defect inspection in industrial environment. Our framework achieves meaningful results in terms of performance and speed. Different from the recent method based on sliding window, we design a combination of a segmentation task and a detection task. The segmentation stage provides coarse ROIs and the detection stage refines the ROIs. This framework is more general and reusable than traditional ones. Besides that, it can provide pixel-wise inference of defect area and is computed efficiently. We also introduce an unusual training strategy for the scarce training data in industrial environment and investigate its special requirement on our FCN architecture. Since we provide a framework to treat the defect inspection as an object segmentation problem, more advanced algorithm for refining the segmentation might be applied to achieve better results. We leave this for future studies.

References

1. Koch, C., Georgieva, K., Kasireddy, V., Akinci, B., & Fieguth, P.: A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. Advanced Engineering Informatics 29(2), 196-210 (2015).
2. Duda, R. O., Hart, P. E., & Stork, D. G.: Pattern classification. John Wiley & Sons (2001).
3. Jian, C., Gao, J., & Ao, Y.: Automatic surface defect detection for mobile phone screen glass based on machine vision. Applied Soft Computing, 52, pp.348-358 (2017).
4. Wells, L. J., Shafae, M. S., & Camelio, J. A.: Automated Surface Defect Detection Using High-Density Data. Journal of Manufacturing Science and Engineering, 138(7), 071001 (2016).
5. Krizhevsky, A., Sutskever, I., & Hinton, G. E.: Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pp. 1097-1105 (2012).
6. Girshick, R., Donahue, J., Darrell, T., & Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580-587 (2014).
7. Long, J., Shelhamer, E., & Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431-3440 (2015).
8. Dong, C., Loy, C. C., He, K., & Tang, X.: Learning a deep convolutional network for image super-resolution. In: European Conference on Computer Vision, pp.184-199.Springer International Publishing (2014).
9. Nam, H., & Han, B.: Learning multi-domain convolutional neural networks for visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4293-4302 (2016).
10. Bian, X., Lim, S. N., & Zhou, N.: Multiscale fully convolutional network with application to industrial inspection. In Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on, pp. 1-8. IEEE (2016).
11. https://www.vidi-systems.com, last accessed 2017/4/10
12. Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pp. 248-255. IEEE (2009).
13. https://hci.iwr.uni-heidelberg.de/node/3616, last accessed 2017/4/10
14. Ren, S., He, K., Girshick, R., & Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, pp. 91-99 (2015).
15. Zeiler, M. D., Krishnan, D., Taylor, G. W., & Fergus, R.:Deconvolutional networks. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on pp. 2528-2535. IEEE (2010).
16. Hou, L., Samaras, D., Kurc, T. M., Gao, Y., Davis, J. E., & Saltz, J. H.: Patch-based convolutional neural network for whole slide tissue image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2424-2433 (2016).
17. Li, Y., He, K., & Sun, J.: R-fcn: Object detection via region-based fully convolutional networks. In Advances in Neural Information Processing Systems, pp. 379-387 (2016).
18. Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE transactions on pattern analysis and machine intelligence 32(9), pp.1627-1645 (2010).