Exploring Faster RCNN for Fabric Defect Detection

Hao Zhou, Byunghyun Jang, Yixin Chen, and David Troendle
Computer and Information Science
University of Mississippi
University, MS USA
hzhou3@go.olemiss.edu, {bjang, ychen, david}@cs.olemiss.edu

***Abstract*—This paper presents a fabric defect detection network (FabricNet) for automatic fabric defect detection. Our proposed FabricNet incorporates several effective techniques, such as Feature Pyramid Network (FPN), Deformable Convolution (DC) network, and Distance IoU Loss function, into vanilla Faster RCNN to improve the accuracy and speed of fabric defect detection. Our experiment shows that, when optimizations are combined, the FabricNet achieves 62.07% mAP and 97.37% AP50 on DAGM 2007 dataset, and an average prediction speed of 17 frames per second.**

***Keywords*-Fabric defect detection, Faster R-CNN, Object detection, Automation**

I. Introduction

Defect inspection is a critical fabric production step in the textile industry. Automated inspection is technically challenging due to the large variety of defect types and variations in size, shape, and texture. Human fabric defect detection dominates in industry. However, human inspection is time-consuming, inconsistent, error-prone, and expensive. Recent advances in software and hardware in computer vision and artificial intelligence have made automatic machine inspection a viable approach for this challenging task.

In this paper, we present a fabric defect detection network (FabricNet) built upon vanilla Faster RCNN to automate the fabric defect detection task. To improve the accuracy and speed of defect detection, our proposed model employs several effective techniques such as Feature Pyramid Network (FPN), Deformable Convolution (DC) network, and Distance IoU Loss function. The proposed FabricNet achieves 62.07% mAP and 97.37% AP50 on DAGM 2007 dataset [1], and an average prediction speed of 17 frames per second on an NVIDIA GTX 1070 GPU.

II. Related Work

A. Vanilla Faster RCNN

Region-based CNNs use selective search techniques [2] to generate region proposals [3]. Compared to original RCNN, Fast RCNN [4] has drastically reduced computational cost by sharing convolutions for region proposals. Faster RCNN [5] further improves the region proposal algorithm of Fast RCNN; instead of using a fixed region proposal algorithm, it introduces a novel Region Proposal Network (RPN) that can share convolutional layers with other networks. By sharing these convolutional layers, Faster RCNN boosts its prediction speed by 10 times over Fast RCNN.

B. Deep Learning For Fabric Defect Detection

There have been several reports that propose to use deep learning algorithms for fabric defect detection. For example, [6] uses one CNN for fabric classification and the other for defect detection. A hybrid approach [7] has been proposed to combine CNN with auto-correlation where the result of auto-correlation is added into CNN as an activation layer. A fully convolutional network [8] is proposed to detect surface defects where two FCNs are adopted for segmentation and detection individually. A similar fully convolutional networks approach [9] is published that proposed a third stage called Matting stage to refine the contour of defects to achieve more accurate results. [10] and [11] propose autoencoder by introducing a method to reconstruct images, and sharing the similarity of image patches respectively.

III. Proposed Method

Our proposed network embraces the basic structure of Faster RCNN [5] as visualized in Figure 1. However, to make it more suitable for fabric defect detection task, we make several effective modifications. First, we replace the backbone of Faster RCNN with ResNet101 [12]. Second, we replace the last stage of ResNet-101 with a Deformable Convolution (DC) block. Third, we introduce the Feature Pyramid Networks (FPN) [13]. Fourth, we replace RoI Pooling with RoI Align [14]. Lastly, we incorporate Distance IoU loss [15].

A. Backbone, Feature Fusion, and Deformable Convolution

The backbone of Faster RCNN usually serves as a feature extractor. It extracts features by several convolutional layers. We propose to use ResNet-101 over ResNet-50 as our backbone. ResNet-101 has five stages, each of which contains a different number of convolutional and some other layers. After the last stage, it generates a feature map of a fabric image, which contains only 'local' features of the images. We observe that most of defects are small taking up only about 10% of fabric images, and they are different in shapes and sizes. Due to locality, features from stage-5 are not very discriminative in separating normal background of fabric images and defects. Therefore, to solve this problem, we adopt Feature Pyramid Networks [13] which is a way to fuse features in an image so

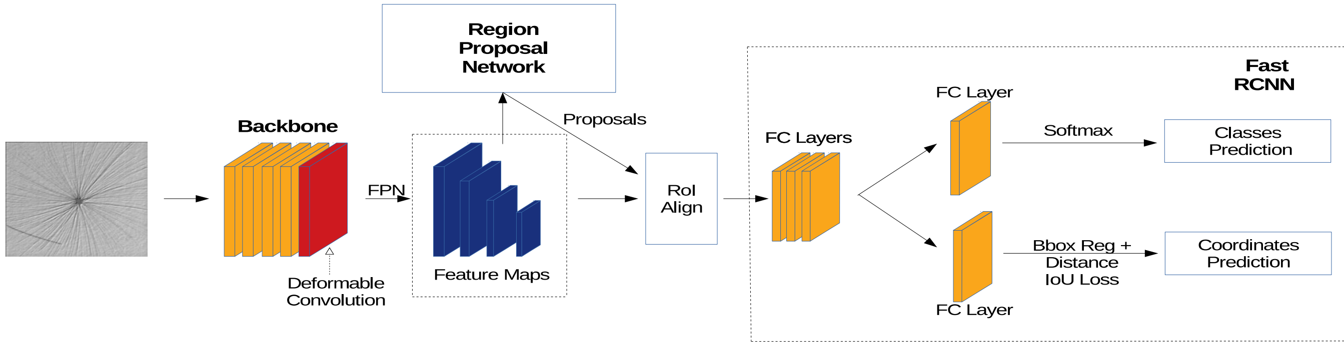


Fig. 1: Architecture of the proposed FabricNet.

that feature maps fuse both high and low resolution features. This technique helps the model distinguish between fabric background texture and defects. Inspired by prior work [16], we also adopt Deformable Convolution (DC) network to solve the problem that fabric defects have different shapes and sizes. DC Network can better adjust for different shapes like thick curves, lines, etc. We replace the last stage, which usually has several convolutional layers, with a DC block.

B. RPN and RoI Align

The role of Region Proposal Network (RPN) is to generate region proposals, so called Region of Interests (RoIs). It includes a binary classification layer and a bounding box regression layer. Traditional Faster RCNN uses RoI Pooling [5] to filter out RoIs. However, RoI Pooling has rounding operations so that coordinates of RoIs tend to be slightly inaccurate. To address this problem, we embrace RoI Align technique from Mask RCNN [14]. RoI Align utilizes bi-linear interpolation to compute the floating-point location values from input. It makes coordinates of RoIs more accurately approximate their original positions.

C. Distance IoU Loss

We also introduce a new loss function specifically designed for object detection problems. Originally, Smooth L1 loss is used for bounding box coordinates in Fast RCNN, where the Smooth L1 loss individually calculates the loss of predicted four bounding box coordinates.

$$L_{SmoothL1} = \begin{cases} \frac{1}{2}x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

where x represents coordinates of each corner of bounding box. We see that smooth L1 loss does not compute loss as a whole since it only cares about each coordinate of bounding boxes. However, when quantifying the performance of the model, Intersection over Union (IoU), $IoU = \frac{B \cap B^{gt}}{B \cup B^{gt}}$, is used, where the ground-truth box $B^{gt} = (x^{gt}, y^{gt}, w^{gt}, h^{gt})$ and predicted bounding box $B = (x, y, w, h)$ affect the performance heavily. With this observation, we employ a new loss, Distance IoU (DIoU) Loss [15], to calculate bounding box regression, which is defined as

$$L_{DIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2}$$

where $\rho(\cdot)$ is the Euclidean distance, b and b^{gt} are the central points of B and B^{gt} respectively, and c is the diagonal length of the smallest enclosing box covering the ground truth box and predicted box. Compared with Smooth L1 loss, Distance IoU loss directly maximizes IoU and minimizes the distance between two central points.

IV. EXPERIMENTS

A. Dataset

Our proposed method is tested on DAGM 2007 dataset [1] which has six categories of fabric images. Each category has 150 images that are manually labeled. All images are 512×512 pixels, with each pixel having 256 grayscale levels. Out of the total 900 images, we randomly selected 540 images as the training set, 180 images as the validation set and fixed 180 images as the testing set. To avoid any unsettled results, we repeated the random selection process 3 times.

B. Data Augmentation

We utilize data augmentation techniques during training to enhance the performance of our model [17], [18]. We use a factor of 0.7 to crop images including ground truth labels. If the cropped images contain no defects at all, we simply discard those images. We also randomly flip the images to cover different orientations of defects and for overall better coverage and accuracy.

C. Implementation Details

We use an NVIDIA GTX 1070 GPU for training and testing. More specifically, transfer learning is used in our model. We used pre-trained weights from ImageNet [19] for ResNet-101. Anchor ratio is $(0.5, 1, 1.5)$ and scale is $(32^2, 64^2, 128^2, 256^2)$. The anchors with IoU of ground truth greater than 0.7 are considered as foreground, while the anchors with IoU of ground truth lower than 0.3 are considered as background. Those IoU scores between 0.3 and 0.7 are ignored in our case.

We use Stochastic Gradient Descent (SGD) to train our model because we found that in our model, SGD is stable and does not cause gradient explosion.

V. Analysis

A. Accuracy

Table I compares the results of different settings and configurations of neural nets we have tested. From Model 1 and 2, it is clear that FPN improves accuracy by approximately 4%. FPN fuses high and low level features to generate semantic-rich feature maps for better understanding the relation of fabric background and defects. Model 3 shows that ResNet101 improves performance over ResNet50 by 2%. We believe that a deeper backbone helps generate semantic-richer feature maps. Model 4 demonstrates that Deformable Convolution (DC) layers make the net slightly better as they are capable of conveying the context of irregular shapes of defects. By comparing Model 4 with 5 and 3 with 6, we see 7% and 10% accuracy improvements respectively. Distance IoU loss can regress the four coordinates of a predicted box as a whole and directly minimize the center distance of the predicted bounding boxes and the ground truth boxes. However, model 5 and model 6 tell us that DIoU loss decreases mAP with DC. We believe that DC brings predicted B-boxes closer to ground-truth B-boxes, which compromises DIoU loss, especially when aspect ratio of both differ.

We visualize the results in Figure 2. The first and third columns show the results of Model 3 and the second and fourth columns show the results of the proposed model. With Model 3, some of the defects are detected with multiple boxes and some are overlooked. Some predicted boxes are larger than defects. Our proposed model, on the other hand, predicts defects with single compacted boxes covering whole defects with more precise locations.

B. Speed

We trained and tested the model on an NVIDIA GTX 1070 GPU. Table II demonstrates that the proposed FabricNet achieves an average prediction speed of 17 frames per second (fps). By comparing model 1 with 2, we notice that there is a significant fps improvement, we believe the number of pre-proposals matters in this case. It is also noticeable that, as incorporating more techniques into the model, we do not encounter any significant speed drop. From Model 2 and 6 of Table II and Model 2 and 6 of Table I, we can say that with around 5 fps drop, we achieve about 12% accuracy improvement. In terms of speed and accuracy ratio, we think it is a reasonable trade-off.

VI. Conclusions

We presented a fabric defect detection network (FabricNet) for fabric defect detection. Our proposed model improves Faster RCNN by integrating several effective techniques tailored for fabric defect detection. Our results with DAGM 2007 dataset demonstrate that artificial neural network is a viable solution for automated defect detection.

References

[1] 29th Annual Symposium of the German Association for Pattern Recognition, "DAGM 2007," 2007.

[2] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "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.

[4] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, 2015.

[6] T. Wang, Y. Chen, M. Qiao, and H. Snoussi, "A fast and robust convolutional neural network-based defect detection model in product quality control," *The International Journal of Advanced Manufacturing Technology*, vol. 94, no. 9-12, pp. 3465–3471, 2018.

[7] W. Ouyang, B. Xu, J. Hou, and X. Yuan, "Fabric defect detection using activation layer embedded convolutional neural network," *IEEE Access*, vol. 7, pp. 70130–70140, 2019.

[8] Z. Yu, X. Wu, and X. Gu, "Fully convolutional networks for surface defect inspection in industrial environment," in *International Conference on Computer Vision Systems*, pp. 417–426, Springer, 2017.

[9] L. Qiu, X. Wu, and Z. Yu, "A high-efficiency fully convolutional networks for pixel-wise surface defect detection," *IEEE Access*, vol. 7, pp. 15884–15893, 2019.

[10] S. Mei, Y. Wang, and G. Wen, "Automatic fabric defect detection with a multi-scale convolutional denoising autoencoder network model," *Sensors*, vol. 18, no. 4, p. 1064, 2018.

[11] H. Tian and F. Li, "Autoencoder-based fabric defect detection with cross-patch similarity," in *2019 16th International Conference on Machine Vision Applications (MVA)*, pp. 1–6, IEEE, 2019.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

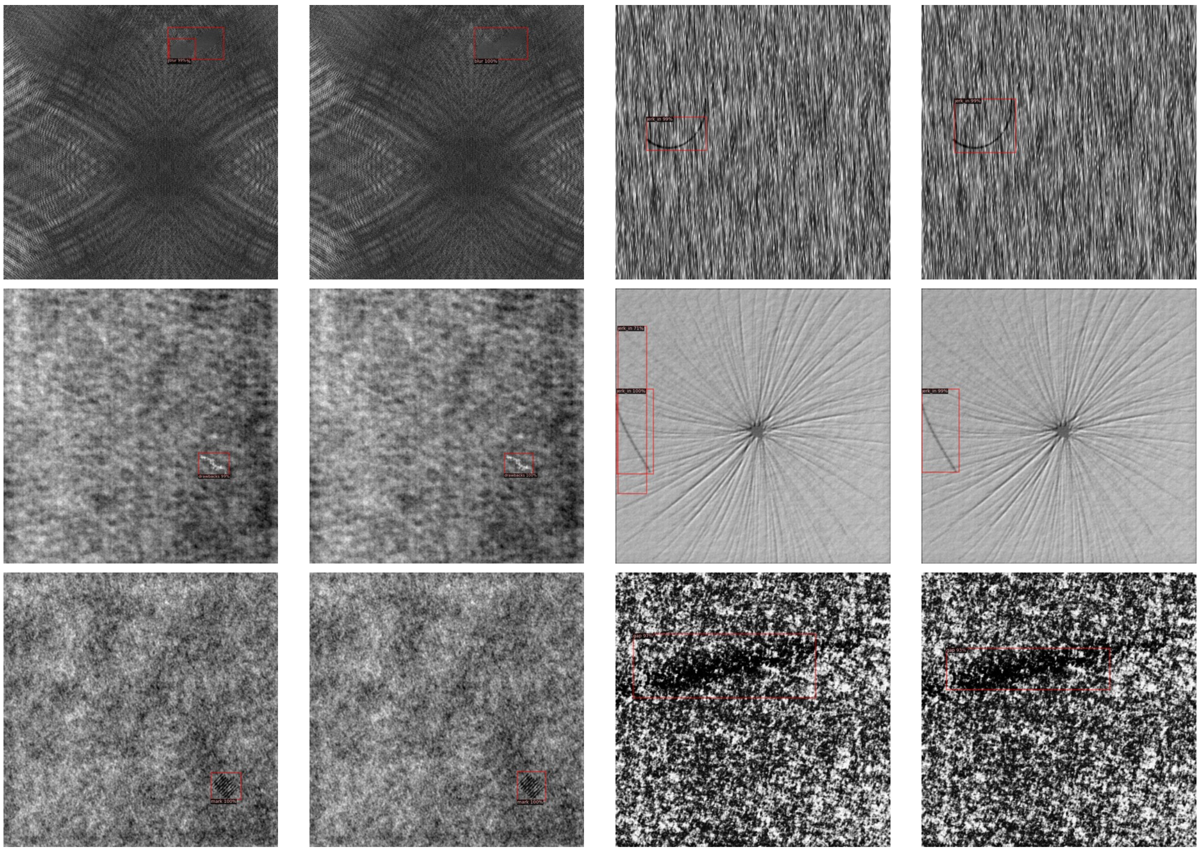
[13] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

TABLE I: Comparison between different settings of faster RCNN.

Model	Backbone	DC	FPN	Loss function	mAP(%)	AP50(%)	AP75(%)	AP80(%)
1	ResNet50			SL1	46.44	92.04	41.35	25.18
2	ResNet50		X	SL1	50.89	93.13	50.11	29.82
3	ResNet101		X	SL1	52.77	93.93	55.31	32.98
4	ResNet101	X	X	SL1	52.87	91.34	58.93	34.48
5 (proposed)	ResNet101	X	X	DIoU	59.70	94.10	69.64	50.81
6 (proposed)	ResNet101		X	DIoU	**62.07**	**97.37**	**69.71**	**52.97**

FPN: Feature Pyramid Network. **SL1**: Smooth L1 Loss. **DC**: Deformable Convolution. **DIoU**: Distance IoU loss.



Model 3 Model 6 (proposed) Model 3 Model 6 (proposed)

Fig. 2: Prediction results: comparison between Model 3 and 6 (proposed).

TABLE II: Prediction speed comparison.

Model	1	2	3	4	5	6
Prediction Speed (fps)	12	21	17	18	17	16

[15] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, “Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression,” *arXiv preprint arXiv:1911.08287*, 2019.

[16] Y. Li, W. Zhao, and J. Pan, “Deformable patterned fabric defect detection with fisher criterion-based deep learning,” *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 2, pp. 1256–1264, 2016.

[17] M. Kisantal, Z. Wojna, J. Murawski, J. Naruniec, and K. Cho, “Augmentation for small object detection,” *arXiv preprint arXiv:1902.07296*, 2019.

[18] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of Big Data*, vol. 6, no. 1, p. 60, 2019.

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.