ISSN: 1813-4890

Fabric Defect Detection and Classification based on Faster Region Convolution Neural Network Deep Learning Algorithm with LabView Deployment

Othman Paschal Msalika[a], Hongchang Sun[b,*] and Yongxiang Jiang[c]

School of Mechanical Engineering, Tianjin University of Technology and Education, Tianjin, China

[a]msalikaothman@yahoo.com, [b,*]sunhog@126.com, [c]jiangyongxiang@tute.edu.cn

Abstract

Considering to the manual and inefficiency methods of textile industries' fabric defect detection methods, the study and research on automated method based on Faster Region based Convolution Neural Network (Faster RCNN) machine vision algorithm is conducted. In this paper the feature extraction and detection algorithm are developed based on Faster RCNN structure. Faster RCNN is improved based on the fabric dataset presented.

Keywords

Fabric Defects; Faster RCNN; Deep Learning; Labview; Tensorflow APIs.

1. Introduction

Detecting defects in fabric images is one of the fundamental tasks in computer vision. It has a lot of applications in textile industries in detecting fabric raw materials and woven finished products defects. Referring computer vision techniques reviewed in [1–3], the task of detecting fabric defects can be challenging especially when different instances of fabric defects vary significantly in terms of shape, colour, texture or size [4]. Nowadays, deep learning approaches give the best digital image processing performance.

2. Methodology

Based on the nature of our dataset two stage detector Faster RCNN [5] model is chosen as the base model to detect Multi-class defects with six types of defects from our constructed data set.

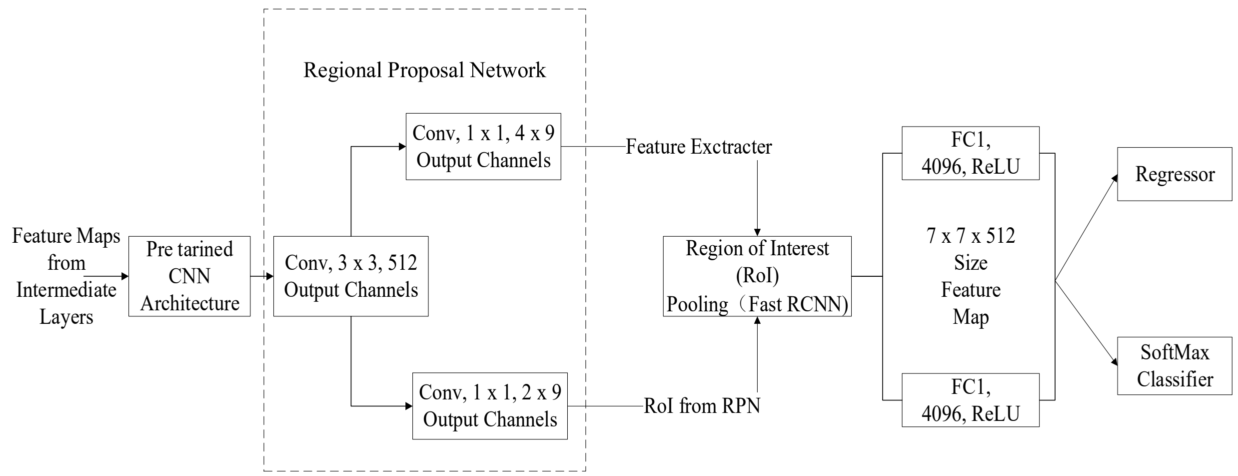


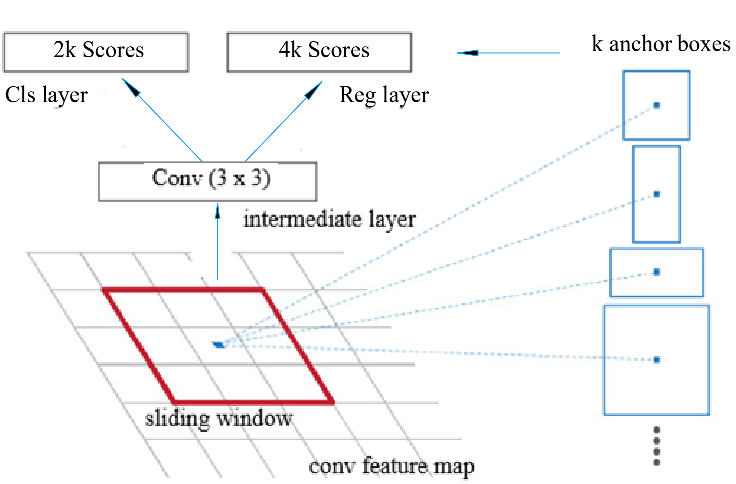
Figure 1. An Overall Network Architecture of Faster RCNN

ISSN: 1813-4890

To reduce and prevent model over-fitting ResNet-101 is preferred as the feature extractor under the pre-trained Convolution Neural Network with Faster RCNN. Res-Net [6] is a deep convolution neural network that employs a residual learning method Figure 3 with a skip connection to avoid the vanishing gradient problem that happens as the number of layers in neural network grows [7].

2.1. Region Proposal Networks, RPN

The RPN serves as the detector's attention mechanism allowing it to produce precise bounding boxes while keeping computation tractability [8]. From Figure 2 it can be noted that the default filter size for each sliding window is 3 × 3. At each sliding-window location, the RPN creates several region proposals, with a maximum number of region proposals of k. The classification layer generates 2k elements that are needed to classify the k proposals which correspond to the RPN's k oriented anchors [10]. The total number of anchors for feature maps with size W × H is k×W×H [11].



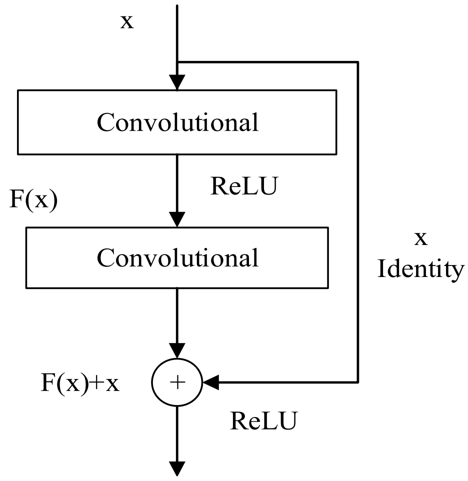


Figure 2. Region Proposal Network (RPN) Architecture **Figure 3.** Residual learning block

2.2. Region of Interest (RoI) Pooling, Regression and Classification

The RoI pooling layer Figure 4 is used to collect the region proposal and input feature maps. The non-fixed size of feature maps distinguishes this layer. Its output is a vector of channel size (k) * w*h, where w and h refer to the feature map's width and height, respectively, and channel refers to the feature map's dimension. The goal of bounding box regression is to obtain the final exact position and return a more accurate target detection box, shown in Fig 4.

$$\text{IoU} = \frac{\text{Intersection Region}}{\text{Union Region}} \tag{1}$$

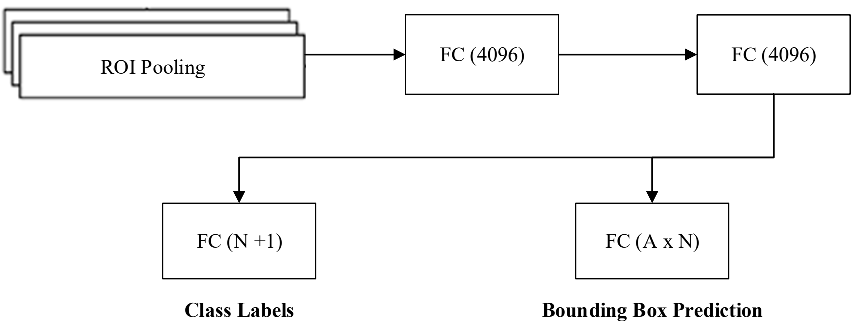
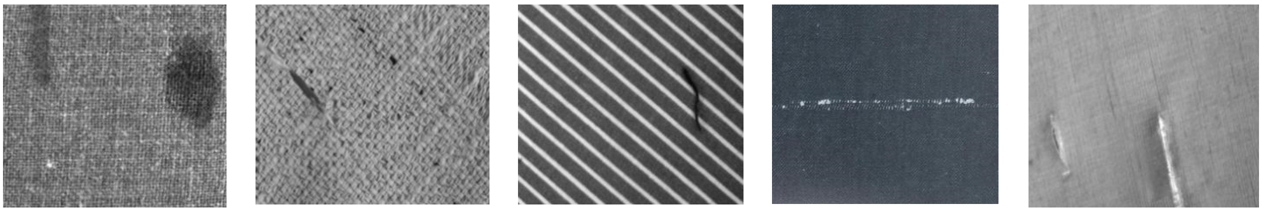


Figure 4. ROI Pooling, Regression and Classification

3. Data Preparation and Augmentation

Oil stain type 1, Oil stain type 2, Hole, Line, draw back and Brocken Pick are fabric defect images in six different categories from common fabric defect images databases. Data augmentation was used to increase the number of training images from existing training samples while also reducing the learning tendency of model over-fitting through a series of random transformations that resulted in believable looking images [12, 13]. Horizontal and vertical flips, Rotation zooming ad some width and height shifts are used to reduce model over fitting and increase the detection accuracy [14]. Figure 1 shows the full network Faster RCNN architecture used. The six different categories of common defects shown in Figure 5 selected to form the data base has mixed characteristics from both raw materials and final real products.



(a) Oil Stain (b) Hole (c) Line (d) Broken Pick (e) Draw back

Figure 5. Samples of fabric defect classes from dataset

4. Experimental Results with LabView Deployment

The metric that used to determine the performance of the model is the average precision (AP) referenced from the Common Objects in Context (COCO) dataset. The AP50 is a calculated value when thresholding the Bounding Boxes at different Intersection over unions (IoUs) say 0.5. The mean average precision (mAP) is the average of AP for (IoUs) that consisted of 0.5 to 0.95 at an interval of 0.05. Table 1 shows the results.

Table 1. AP50/mean average precision (mAP) for each class. IoU = intersection over union.

Class Category	AP (IoU = 0.5) (%)	AP (IoU = 0.5:0.05:0.95) (%)
All Categories	60.9	**30.48**
Oil Stain 1	65.65	31.89
Oil Stain 2	69.72	38.21
Line	45.03	27.82
Hole	70.60	34.62
Draw Back	58.32	28.67
Brocken Pick	40.10	21.24

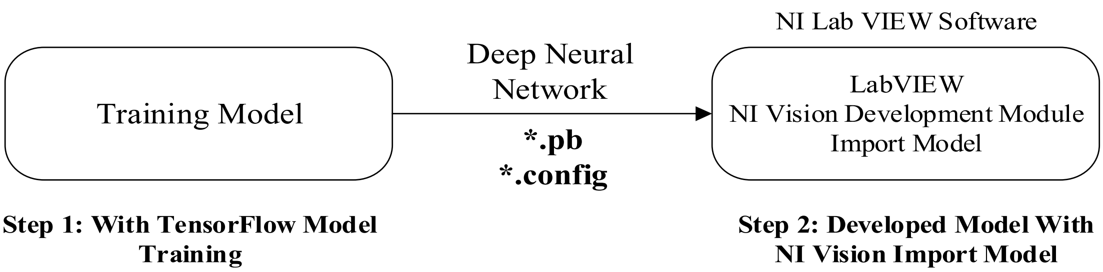
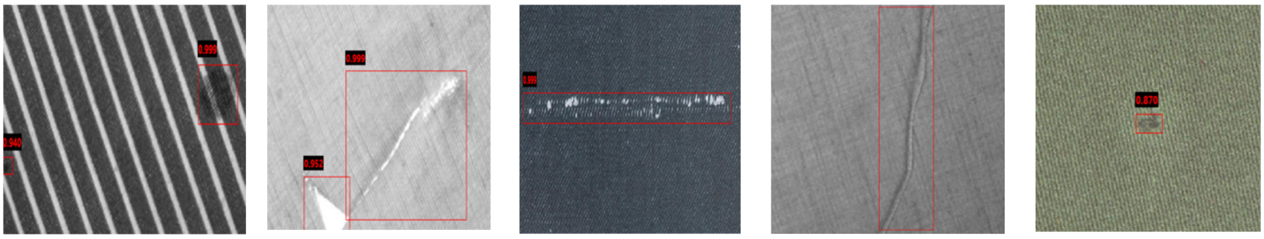


Figure 6. Developing TensorFlow Deep Learning program in LabView

A Vision Development Module (VDM) from LabVIEW includes a TensorFlow model importer for importing TensorFlow deep learning model for deep leaning related applications [13]. A total of two important steps in Figure 6 needed to deploy TensorFlow deep learning model in LabVIEW. The Protocol Buffer (. pb) is a supported model files for frozen model.



(a) Oil Stain (b) Draw Back (c) Brocken Pick (d) Line (e) Hole

Figure 7. Images with the detected defects after model deployment

After the development of a compatible model, LabVIEW is used to load the trained model, the inputs and outputs were configured, and the model run to analyses the results was shown in Fig.7.

5. Conclusion

In fact, fabric data sets collected from different databases were used to conduct this study with various detection complexity i.e., size, colour variations, background and shapes. Finally, the model trained is deployed in LabView using TensorFlow APIs to verify the performance of the model in detecting defects. As part of suggestions, is necessary to expand the database especially for drawback defect, broken pick defect and line defect for training and make it well balanced for the model to be strong.

Acknowledgments

This paper is supported by School Project of TUTE (KJ2003, KJ2004).

References

[1] A. Rasheed et al., “Fabric Defect Detection Using Computer Vision Techniques: A Comprehensive Review,” Math. Probl. Eng., vol. 2020, pp. 1–24, 2020, doi: 10.1155/2020/8189403.

[2] C. Li, J. Li, Y. Li, L. He, X. Fu, and J. Chen, “Fabric Defect Detection in Textile Manufacturing: A Survey of the State of the Art,” Secur. Commun. Networks, vol. 2021, 2021, doi: 10.1155/2021/9948808.

[3] K. Hanbay, M. F. Talu, and Ö. F. Özgüven, “Fabric defect detection systems and methods--A systematic literature review,” Optik (Stuttg)., vol. 127, no. 24, pp. 11960–11973, 2016, doi: 10.1016 /j. ijleo.2016.09.110.

[4] B. G. Chhapkhanewala and S. L. Vaikole, “Automated Fabric Fault Detection System,” no. 4, pp. 36–40, 2017.

[5] R. Girshick, “Fast R-CNN,” Proc. IEEE Int. Conf. Comput. Vis., vol. 2015 Inter, pp. 1440–1448, 2015, doi: 10.1109/ICCV.2015.169.

[6] S. R. J. S. Kaiming He, Xiangyu Zhang, “Deep Residual Learning for Image Recognition Kaiming,” pp. 770–778, 2016.

[7] K. Lee, G. Hong, L. Sael, S. Lee, and H. Y. Kim, “Multidefectnet: Multi-class defect detection of building façade based on deep convolutional neural network,” Sustain., vol. 12, no. 22, pp. 1–14, 2020, doi: 10.3390/su12229785.

[8] T. Vu, H. Jang, T. X. Pham, and C. D. Yoo, "Cascade RPN: Delving into high-quality region proposal network with adaptive convolution," arXiv, no. NeurIPS, 2019.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.

[10] Q. X. Yu-Peng Chena, Ying Li, Gang Wanga, "A Multi_strategy Region Proposal Network.pdf." pp. 1–17, 2018.

[11] B. Wei, K. Hao, X. S. Tang, and L. Ren, Fabric defect detection based on faster RCNN, vol. 849. Springer International Publishing, 2019.

[12] F. CHOLLET, Deep Learning with Python. United States of America: MANNING SHELTER ISLAND, 2018.

[13] National Instrument, "Engineer Ambitiously - NI," 2016. https://www.ni.com/es-es.html.

[14] "Concepts - NI Vision 2019 for LabVIEW Help - National Instruments." http:// zone.ni. com/ reference/ en-XX/help/370281AG-01/nivisionconcepts/spatial_calibration_concepts/.