Defect Detection in Plain Weave Fabrics by Yarn Tracking and Fully Convolutional Networks

Leon Weninger
Institute of Imaging & Computer Vision
RWTH Aachen University
Aachen, Germany
leon.weninger@rwth-aachen.de

Marcin Kopaczka
Institute of Imaging & Computer Vision
RWTH Aachen University
Aachen, Germany
marcin.kopaczka@lfb.rwth-aachen.de

Dorit Merhof
Institute of Imaging & Computer Vision
RWTH Aachen University
Aachen, Germany
dorit.merhof@lfb.rwth-aachen.de

***Abstract*—Weaving is a highly automated industrial process. Due to small inaccuracies during the production process, different types of weave defects can occur, by which the quality of the produced fabric is heavily impaired. The defects can diminish the selling price by up to 50%. Current automated visual defect detection systems need to be adjusted by a trained operator to every new fabric, making them impractical for industrial use. We present a novel automated visual defect detection framework which localizes and tracks yarns in new and unseen fabrics without the need for tedious settings, and which consecutively detects anomalies. The detection of weave defects is based on three consecutive steps, (1) the identification of single weft and warp float-points with fully convolutional networks, (2) the tracking of single yarns based on a set of rules, and finally (3) the recognition of defects using statistical analysis.**

***Index Terms*—Fabric Defect Detection, Plain Weave Fabrics, Yarn Tracking, Fully Convolutional Network, Anomaly Detection**

I. Introduction

The textile industry is one of the biggest industries in the world and produces several million tons of fabric every year. However, fabric defect detection is mostly provided by human operators. Whether due to fatigue, inattention or simply brief distraction, human operators are quite prone to missing even important defects in textiles. Undetected weaving defects lead to low quality finished products. In the end, the selling price of these low quality products diminishes, or they remain unsaleable.

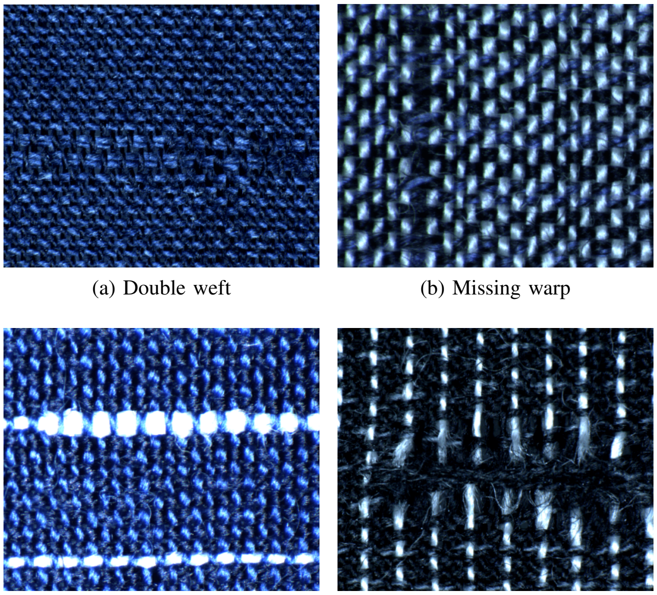
Automatic defect detection for fabrics may overcome this problem. The presently most promising approaches are all based on image analysis techniques: it is easy to take pictures of the fabric, either on-loom or off-loom using a digital camera, and to analyze the picture with a machine vision system. A variety of visual fabric defect detection frameworks have already been presented in the past. However, most algorithms rely, either implicitly or explicitly, on a skilled human operator who needs to adjust various settings of the framework before it can be used on a new fabric. Other frameworks are only able to work on specific kinds of fabric, or are only able to detect large flaws.

With today's rapidly changing fashion and clothing trends, fabric production is confronted with an ever-increasing need for fast and technically convenient fabric design switching. In response, there is a wide need for a visual defect detection system that is capable of flaw detection even in previously unseen fabrics, without any prior knowledge of color, and which can furthermore cope with a wide range of yarn thicknesses and yarn spacings. Such an algorithm would continue its analytical work on the loom output, even following changes in the produced fabric, without any intervention of a human operator.

Convolutional neural networks (CNNs) have emerged in recent years as the new state-of-the art technique in image processing. Numerous improvements in CNNs have already now revolutionized our envisioning of machine capabilities, and many of the advances are actually implemented in a variety of industrial applications. These networks are reliant on a large number of training images, i.e. a vast ground-truth is needed. Such a vast ground-truth dataset does not exist for textile defects, since a vast number of potentially different defects can occur for the many different fabrics and textures under study. This has so far prevented the application of CNNs to fabric defect analysis.

Fabrics are composed of horizontal yarns, called wefts, and vertical yarns, named warps. The points at which the yarns intersect are either termed warp float-points, or weft float-points, depending on which of the two yarn types lies on top. Independent of the specific fabric, both types of float-points have a similar structure, and it is this specific property that can be utilized for the purpose of defect detection in unknown fabrics.

In this paper we describe a new approach to fabric defect detection based on float-point analysis. Relying on CNNs, single yarn float-points can be detected. The yarns themselves can then be tracked throughout the entire image which finally allows localizing all existing defects. This specific approach requires neural networks only for one sub-step, for which annotated data with a sufficient number of labels for each class can easily be generated. In comparison, other approaches rely on annotated defects. A database with enough variety of defects and fabrics for the training of a CNN is very difficult to obtain. In this paper, neural networks are used to detect irregularities in textiles without the need of annotated defects. This makes our approach much easier to use than previous approaches.



(a) Double weft (b) Missing warp

(c) Slub (d) Hole

Fig. 1. Typical fabric defects found in the used textiles

II. Previous Work

Visual defect detection systems are widespread for various industrial use cases. For fabrics, a variety of distinct approaches have been published, their complexity ranges from simple thresholding techniques [1] to more sophisticated algorithms such as Gabor Wavelets [2], [3], or correlation approaches [4].

A. Literature surveys

The importance and wide relevance of our present topic is reflected in the large number of available publications. An overview over the different methods is provided in four different surveys that were carried out in recent years. The most influential and extensive one with more than 160 publications covered and more than 170 citations was written by Kumar [5] in 2008. Two further surveys were published in 2009 by Mahajan et al. [6], and in 2011 by Ngan et al. [7]. The most recent one was published in 2016 by Hanbay [8].

An exhaustive quantitative comparison between different approaches was published by Schneider [4] in 2015. He implemented several of the most promising published algorithms and evaluated them on a a large independent dataset. He found that essentially none of the published algorithms can actually achieve the accuracies claimed by the respective authors on this dataset.

B. Classic approaches

Classic approaches can be divided in to:

- Basic approaches
- Spectral approaches
- Statistics based approaches

By far the majority of publications use spectral approaches to detect defects in fabrics. Spectral approaches are indeed well-suited for this purpose, due to the highly repetitive structure of wefts and warps in fabrics. For example, a Fourier transformation can reveal deviations from a criss-cross structure. Exemplary approaches for this techniques are [9], [10]. However, in fabrics, the repetitive structure can have small deviations due to the production process. Since many of these small deviations do not actually represent defects, they lead to a large number of false positive detections using Fourier transformation techniques. Similarly, small defects can remain undetected. Another possibility is the use of Wavelets or Gabor filters, as demonstrated by [2], [3]. These filters can detect anomalies in the frequency domain while preserving locality information. As a result, they can detect much smaller defects – some of the best fabric defect detection results have been reported using these techniques.

Statistics based approaches often rely on Local Binary Patterns (LBP) or Gray-Level Co-occurrence Matrices (GLCM). These techniques can be used to compare new images to known defect-free images, and are computationally efficient. This makes them suitable for real-time implementation on embedded devices. Exemplary implementations can be found in [11], [12].

One of the most promising approaches relies on detection of single warp float-points by cross-correlating a warp float-point template with fabric images [4]. The located float-points are used together with a yarn and grid matrix in order to get a complete model of the fabric. With this model, fabric defects can be detected with different thresholding and histogramming techniques. in [4], a very high accuracy is reported, as even small defects can be detected with this approach.

However, all of the presented approaches require fine-tuning by hand for every new fabric, which necessitates either providing an exemplifying defect-free image, or function only under very specific conditions. In the end, for an industrial use of the respective frameworks, a trained operator is still needed in order to robustly detect the defects.

C. Deep learning approaches

In recent years, deep learning techniques have started to revolutionize computer vision systems in various industries. For example, Masci et al. [13] report on the successful use of CNNs for steel defect classification. With this approach, much better results than any other approach in this area were achieved. For fabric defect detection, similar approaches remain a niche so far due to the diversity of fabrics and defects.

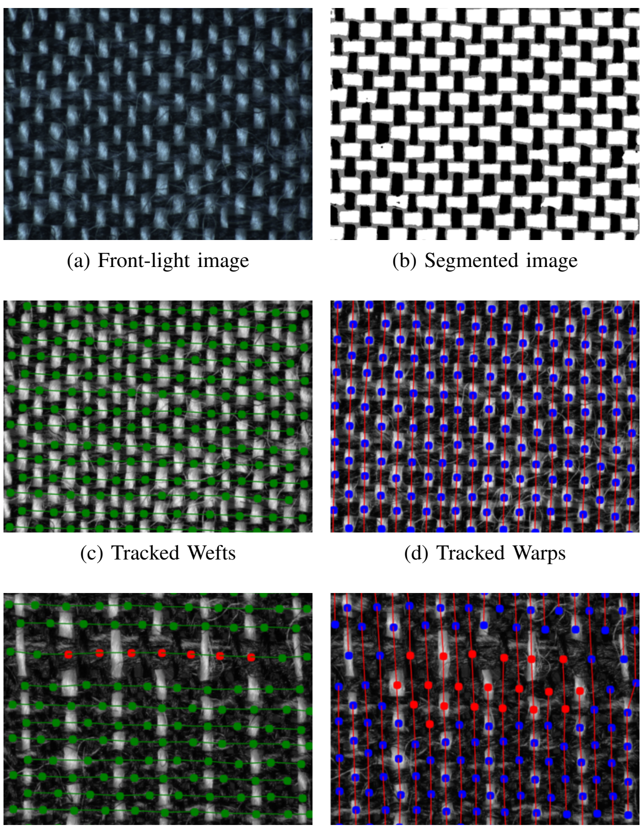
An equally promising approach using deep learning is reported by [14]. They trained two stacked denoising autoencoders, one for defect detection and one for defect localization. This approach was compared to a single stacked denoising autoencoder, and both approaches seem to perform well.

In the end, however, this approach again relies on the availability of both defect-free as well as defective samples for the training of the autoencoders. Without prior knowledge of a fabric, this technique can not be used to detect defects.

III. Methods

The method proposed in this paper allows the detection of defects in unseen fabrics, without any prior knowledge about the fabric apart from the order of magnitude of yarn spacing. In particular, there is no need for defects to be given when training the algorithm. All that is needed is to provide a limited set of images with labelled locations of the weft and warp float-points. The highly superior advantage of this new method in comparison to other approaches is that the fabric itself is available in large amounts, and can therefore be easily analyzed, whereas an exhaustive set of all possible defects for corresponding fabrics, which would be very difficult to obtain, is not needed.

The complete image processing pipeline is illustrated in Figure 2: An image of the fabric (Figure 2a) is first segmented in single weft and warp float-points (Figure 2b). Using the segmented image, individual weft and warp yarns are tracked and followed through the whole image (Figure 2c and 2d). Every float-point which does not lie near the edge of the image is finally classified as either defective or defect-free (Figures 2e and 2f), depending on specific measurements.



(a) Front-light image
(b) Segmented image
(c) Tracked Wefts
(d) Tracked Warps
(e) Faults detected in wefts
(f) Faults detected in warps

Fig. 2. Fabric defect detection pipeline

A. Segmentation in weft and warp float-points

Different fabrics can have different yarn thickness, higher or lower yarn density, and one or more different colors. Depending on the fabric and possible defects, front-light or back-light images are favorable. As defects in a variety of different fabrics need to be detected, we use both front-light and back-light images for optimal results. If two pictures of a fabric are taken at exactly the same place of the fabric while the fabric is illuminated once from the front and once from the back, we can concatenate both images, creating a two-channel, or six-channel image if colored images are used. Such a six-channel image can be fed into a convolutional neural network.

In a first step, single weft and warp float-points are localized in such a six-channel image by semantic image segmentation using CNNs. The output segmentation consists of three different values: weft float-points (white, or 2), warp float-points (black, or 0) and neutral (grey, or 1). The spacings between single yarns and the interims between weft and warp float-points are classified as neutral. Finally, to suppress noise in the output image, morphological filtering is used. One binary opening and one binary closing operation with size 3x3 are carried out on weft-float as well as on warp-float areas.

B. Utilized networks

For the semantic image segmentation, which is a critical step in this pipeline, the U-net [15] and the fully convolutional network [16] provide appropriate network designs. Based on these designs, three different network architectures, labeled A, B, and C, were evaluated. Due to the nature of float-point detection, adjustments on the number of layers and feature maps per layers were carried out. Information about float-points is very local and easy to grasp for neural networks. In consequence, a reduction of the number of network layers and feature maps does not significantly reduce the quality of the output. Meanwhile, a decrease in the number of convolutions and parameters leads to much faster image processing, crucial for fabric defect detection. For training, the adam optimizer was used with a learning rate of 0.01. Further, batch-normalization layers [17] were introduced in all networks for improved training.

The U-net [15] served as inspiration for networks A and C. In contrast to the original network, both networks are shallower. Network A uses only two pooling steps, a total of 10 convolution layers, and 128 feature maps in the most contracted layers. Three pooling steps, a total of 14 convolution layers, and a maximum of 512 feature maps per layer were used for network C. For comparison, the original network uses 19 convolution layers, 4 pooling steps, and a maximum of 1024 feature maps.

Network B was derived from the fully convolutional architecture described in [16]. However, the total number of convolution layers was reduced from 14 to 7, the number of pooling operations from 5 to 3, and the number of feature maps in the fully convolutional layers from 4096 to 384.

C. Yarn Tracking

Every black (resp. white) region in the segmented and morphologically filtered image is considered as a single warp (resp. weft) float-point – the location is set at its center. To track single yarns through the image, these superpixels need to be connected. It is assumed that warps lie vertically in the image, and wefts horizontally. Thus, for each warp float-point, the upper and lower neighbor need to be identified. This is done by searching for the nearest neighbors in a vertical tube, centered at the float-point. The tube diameter is fixed based on the superpixel size. With the area of a float-point defined as the number of pixels of the superpixel, the tube diameter can be calculated as:

$$d_{tube} = \frac{\sqrt{Area(floatpoint)}}{2} \tag{1}$$

The nearest upper warp float-point is considered as the upper neighbor, while the nearest lower warp float-point in this tube is considered as the lower neighbor. The distance between two points is defined as the number of pixels between the two centers. In such a way, no assumptions about the spacing between yarns is necessary. This simple algorithm works already well for the majority of float-points, but not perfectly due to some irregularities in the fabrics or a noisy image segmentation. To correct small errors, minor corrections are carried out:

- *Remove non-plausible connections*
  Since yarns are straight lines, two float-points cannot share the same lower (or upper) neighbor. If this is observed, both connections are cut.
- *Patch missing links*
  If the majority of links are correct, the median distance between two float-points can be obtained. The median distance can then be used to estimate the location of un-found neighbors. The estimated location of the neighbor is searched within a radius of one third of the median distance, and if a floating point found in this area, it is considered as the neighbor.

The same procedure is repeated for the weft float-points, but with a tube in the horizontal direction.

D. Defect detection

Once all yarns are tracked through the image, several measurements for every float-point can be extracted. In particular, the size of the float-point area and the distances to the upper and lower (resp. right and left) neighbor are known. These parameters can be used for an automatic defect detection. For a defect-free fabric, the parameters should remain roughly the same for every float-point. In fact, defective float-points can now be determined via anomaly detection algorithms. For the anomaly detection, first the Minimum Covariance Determinant (MCD) algorithm is used to estimate location and scatter of the data. This algorithm can robustly estimate these, even in the case of some strong deviating points by choosing a subset of samples (e.g. 80%) with the smallest covariance determinant. The exact procedure is explained in [18].

For every float-point, the Mahalanobis distance to the center of the data can then be calculated with:

$$MD(x) = \sqrt{(x-\mu)^T S^{-1}(x-\mu)}, \tag{2}$$

where μ is the centre and S the covariance computed by the MCD algorithm.

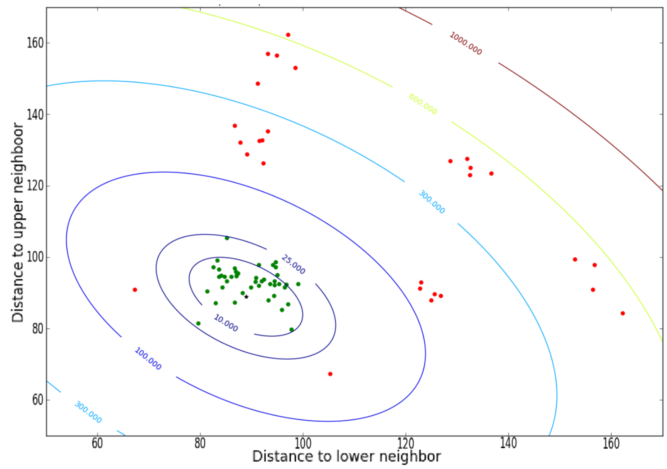


Fig. 3. A plot of the distances of each float-point in an image with a defect to its neighbors. In the defect detection algorithm, a third dimension, the size of the float-point, is also used but can not be depicted here. Points which are marked as defective are colored in red, whereas defect-free floating points are marked in green. Readers of the digital version are invited to zoom in for a better view.

Finally, the Mahalanobis distance can be used to classify single float-points as defect-free or defective via a threshold.

E. Analysis of the Results

For a binary classification, such as defective and non-defective images of fabrics, sensitivity and specificity are the classical statistical measures and were thus applied to the output of the defect detection algorithm:

True Positives (TP)	Correctly identified as defective
False Positives (FP)	Incorrectly identified as defective
False Negatives (FN)	Incorrectly identified as defect-free
True Negatives (TN)	Correctly identified as defect-free
Sensitivity (TPR)	Probability of correct defect identification
Specificity (TNR)	Probability of correct defect-free identification
Accuracy (ACC)	Proportion of correctly classified data

TP, FP, FN and TN can be read directly from the output. Sensitivity, specificity and accuracy need to be calculated according to the following equations:

$$TPR = \frac{TP}{TP + FN} \tag{3}$$

$$TNR = \frac{TN}{TN + FP} \tag{4}$$

$$ACC = \frac{TP + TN}{TP + FP + FN + TN} \tag{5}$$

To detect even small defects and to be less susceptible to noise, the number of faulty float-points is calculated for each of the four quadrants (with an overlap of 100px) of an image, and then given as a percentage of all points in this quadrant. The maximum percentage of faulty points of the four quadrants is then used for classification. If it was higher than a predefined threshold, the image was marked as defective.

Two parameters can be set for the defect detection algorithm: one threshold for the minimum Mahalanobis distance for each float-point to be marked as defective, and one threshold for the minimum percentage of float-points needed for classification as defective. Initially, the threshold 1 was set to 30 and the threshold 2 to 1.5, but they were adjusted by hand for each fabric according to the amount of noise in the segmentation result and the size of the defects.

IV. Dataset

For this work, nine different plain weave fabrics were at hand. All fabrics had at least one defect which occurred during production. To obtain a broader variety of defects, some small local defects such as holes were added by hand.

Plain weave is the most commonly used weave type. Each weft goes under one warp and then over one warp. The next warp yarn does the same, but in reverse order, resulting in a criss-cross pattern. Other weave types, such as twill and satin weaves, where the warp yarn passes under one weft, and then over at least two wefts, were not analyzed for this work.

For this work, a total of 1431 plain weave images from 9 different fabrics with 89 defects were obtained. In this context, one images denotes a concatenated front-light and back-light image (see Section III-A). The images were taken from a close distance with a 50mm lens. In at least one image of every fabric, the weft-float and warp-float regions were hand-labelled for the training of the FCN.

TABLE I
Overview over plain weave images taken and analyzed during this work

Fabric	#Images	#Images with defects	#Annotated images
P1	212	13	4
P2	160	15	6
P3	160	15	4
P4	159	14	2
P5	139	12	3
P6	156	11	2
P7	170	4	1
P8	147	2	1
P9	128	3	1
Total:	1431	89	24

V. Evaluation and Results

Our new fabric defect detection pipeline was tested for all three different network architectures explained in Section III-B and for all fabrics detailed in Table I. Detailed results can be found in Tables II, III and IV. The fabric number in the tables indicates the fabric on which they were tested. All neural networks tested were trained without the fabric on which they are tested. Hence, the tables demonstrate the ability of the applied neural networks to successfully analyze other fabrics.

Overall, the fully convolutional architecture (network B) as well as the bigger U-net architecture (network C) produce

TABLE II
Classification results of Network A

Fa.	Measurements						
	TP	FP	TN	FN	TPR	TNR	ACC
P1	10	65	134	3	0.77	0.67	0.68
P2	5	22	123	10	0.33	0.85	0.80
P3	14	0	145	1	0.93	1.00	0.99
P4	8	2	143	6	0.57	0.99	0.95
P5	4	7	125	3	0.57	0.95	0.93
P6	5	14	131	6	0.45	0.90	0.87
P7	2	8	158	2	0.50	0.95	0.94
P8	1	0	145	1	0.50	1.00	0.99
P9	3	4	121	0	1.00	0.97	0.97
TL	52	122	1225	32	0.62	0.91	0.89

TABLE III
Classification results of Network B

Fa.	Measurements						
	TP	FP	TN	FN	TPR	TNR	ACC
P1	13	2	197	0	1.00	0.99	0.99
P2	5	0	145	10	0.33	1.00	0.94
P3	15	1	144	0	1.00	0.99	0.99
P4	13	5	140	1	0.93	0.97	0.96
P5	5	0	132	2	0.71	1.00	0.99
P6	4	35	110	7	0.36	0.76	0.73
P7	3	8	158	1	0.75	0.95	0.95
P8	2	0	145	0	1.00	1.00	1.00
P9	2	2	123	1	0.67	0.98	0.98
TL	62	53	1294	22	0.74	0.96	0.95

TABLE IV
Classification results of Network C

Fa.	Measurements						
	TP	FP	TN	FN	TPR	TNR	ACC
P1	12	1	198	1	0.92	0.99	0.99
P2	9	1	144	6	0.60	0.99	0.96
P3	15	0	145	0	1.00	1.00	1.00
P4	7	0	145	7	0.50	1.00	0.96
P5	3	0	132	4	0.43	1.00	0.97
P6	2	0	145	9	0.18	1.00	0.94
P7	3	6	160	1	0.75	0.96	0.96
P8	2	0	145	0	1.00	1.00	1.00
P9	3	2	123	0	1.00	0.98	0.98
TL	56	10	1336	28	0.67	0.99	0.97

superb results, while the smaller U-net architecture (network A) performs mediocre by comparison. Network C best in overall accuracy (0.96) and TNR (0.99), network B achieves the best TPR (0.74). Only 11 false positives were detected using network C, by far the best result and a very important measurement. In industrial use, every detected defect would necessitate a human supervisor to check the fabric. Since this is time-consuming, a high FP rate is unacceptable.

With the two well performing network architectures, the fabrics P1, P3, P8 and P9 achieve outstanding results. Perfect results are achieved on fabric P3 and P8, all defects are correctly detected and no false alarms are given for both well performing networks.

In Figure 4, an original test image of fabric P8 is displayed together with the detected warp and weft float-points and the tracked warp yarns. Float-points, which were labeled as defective are marked with red color, defect-free float-points

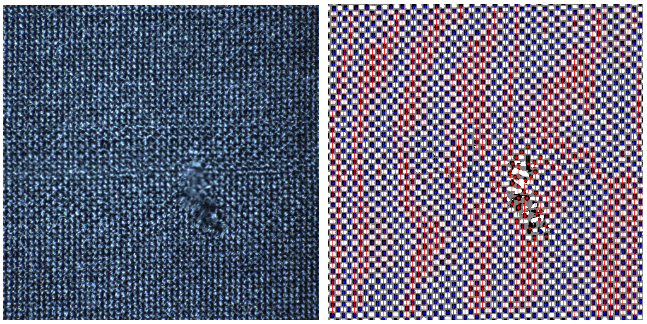


Fig. 4. Exemplary defect detection result: Fabric P8, image 180

are marked in blue. Weft yarns are not displayed for better visibility, they show comparable classification results. As demonstrated for this specific fabric, as well as on one of the other three fabrics with outstanding results, the performance of the algorithm seems already good enough for industrial use.

Unfortunately, not all fabrics show such excellent results. The worst performance can be observed on fabric eight. For every network architecture, more defects were missed than detected. The reason for this under-performance is to be sought in an unsatisfactory image segmentation result. Here, the image segmentation step could not grasp the structure of the fabric.

VI. Discussion

The presented framework is capable of robustly tracking the location of all yarns throughout the image, and allows detection of defects in plain weave fabrics. For each fabric in the dataset, the algorithm was trained only on other fabrics. By this procedure, the generalization ability of the algorithm to unknown fabrics is demonstrated. For two fabrics, very promising results were obtained: all defects were detected and no false alarm was given. Accuracies of more than 96% were obtained for most other fabrics. For these achieved accuracies, and especially for the good segmentation results, the combination of using both front- and back-light images was crucial. Using both types of lighting at the same time provides clear advantages for defect detection. However, an exhaustive quantitative testing, comparing this dual-lighting approach to front-light only and back-light only, still needs to be performed.

The most promising results were obtained for fabrics with similar fabrics in the training set. In contrast, the weave defect detection did not work well on all fabrics. In effect, the blind yarn detection, and thus the following defect detection, failed on one fabric, fabric P6. This fabric had a texture disparate from all other fabrics in the dataset and was furthermore intended for a entirely different use case.

A total of 1431 images from nine different fabrics with a total of 89 mistakes were analyzed. While this is enough to demonstrate the functionality of the presented algorithm, the ability to generalize to new fabrics would further improve with a higher number of different fabrics. Thus, for a use case in industry, more ground-truth labels of the semantic segmentation of more divers fabrics are necessary.

VII. Conclusion

We have presented a novel approach for defect detection in textiles. The framework is shown to work well on plain weave fabrics. For the first time, yarns can be localized and tracked in unseen fabrics without tedious parameter settings. Subsequently, the weave defects can be easily detected. For this innovative approach, no defects were necessary for training. Without algorithmic changes, the application of this framework to other textiles such as twill and satin weave fabrics appears feasible.

References

[1] E. Pace *et al.*, "Operator-free fabric web inspection system," Oct. 8 2002, US Patent 6,462,820.

[2] S. C. Kim and T. J. Kang, "Texture classification and segmentation using wavelet packet frame and gaussian mixture model," *Pattern Recognition*, vol. 40, pp. 1207–1221, 2006.

[3] J. L. Raheja, S. Kumar, and A. Chaudhary, "Fabric defect detection based on gclm and gabor filter: A comparison," *Optik - International Journal for Light and Electron Optics*, 2013.

[4] D. Schneider, "On-loom fabric defect detection - state-of-the-art and beyond," Ph.D. dissertation, RWTH Aachen University, 2015.

[5] A. Kumar, "Computer-vision-based fabric defect detection: A survey," *IEE Transactions on Industrial Electronics*, vol. 55, pp. 348–363, 2008.

[6] P. Mahajan, S. Kolhe, and P. Patil, "A review of automatic fabric defect detection techniques," *Advances in Computational Research*, vol. 1, 1 2009.

[7] H. Y. Ngan, G. K. Pang, and N. H. Yung, "Automated fabric defect detection - a review," *Image and Vision Computing*, vol. 29, pp. 442 – 458, 2011.

[8] K. Hanbay, M. F. Talu, and Ö. F. Özgüven, "Fabric defect detection systems and methods—a systematic literature review," *Optik - International Journal for Light and Electron Optics*, vol. 127, no. 24, pp. 11 960 – 11 973, 2016.

[9] E. J. Wood, "Applying fourier and associated transform to pattern characterization in textiles," *Textile Research Journal*, vol. 60, pp. 212–220, 1990.

[10] C.-H. Chan and G. K. H. Pang, "Fabric defect detection by fourier analysis," *IEEE Transactions on Industry Applications*, vol. 36, no. 5, pp. 1267–1276, Sep 2000.

[11] F. Tajeripour, E. Kabir, and A. Sheikhi, "Fabric defect detection using modified local binary patterns," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, no. 1, 2007.

[12] J. L. Raheja, B. Ajay, and A. Chaudhary, "Real time fabric defect detection system on an embedded dsp platform," *Optik - International Journal for Light and Electron Optics*, vol. 124, no. 21, pp. 5280 – 5284, 2013.

[13] J. Masci, U. Meier, D. Ciresan, J. Schmidhuber, and G. Fricout, "Steel defect classification with max-pooling convolutional neural networks," in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, June 2012, pp. 1–6.

[14] 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. PP, no. 99, pp. 1–9, 2016.

[15] O. Ronneberger, P.Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, ser. LNCS, vol. 9351. Springer, 2015, pp. 234–241.

[16] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *CoRR*, vol. abs/1605.06211, 2016.

[17] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015.

[18] M. Hubert and M. Debruyne, "Minimum covariance determinant," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 1, pp. 36–43, 2010.