






A Hybrid Machine Learning Approach to Fabric Defect Detection and Classification

Swash Sami Mohammed  and Hülya Gökalp Clarke  

Ondokuz Mayıs University, 55270 Samsun, Turkey
hulya.gokalp@omu.edu.tr

Abstract. This paper proposes a novel approach to detect and classify defects in fabric. Fabric defects may cause a roll of fabric to be graded as second or worse in quality checks, and may affect sales. Traditionally, fabrics are inspected by skilled workers at an inspection platform, which is a long and tiring process, and error-prone. Automated detection of defects in fabric has the potential to eliminate human errors and improve accuracy; therefore it has been an area of research over the last decade. This paper proposes a novel model to detect and classify defects in fabric by training and evaluating our model using the AITEX data set. In the proposed model, the images of fabrics are first fed into U-Net, which is a convolutional neural network (CNN), to determine whether the fabric is defect-free or not. VGG16 and random forest are then used to classify the defects in the fabrics. The training settings of the model were chosen as initial learning rate = 0.001, $\beta_1 = 0.9$ and $\beta_2 = .999$. The proposed approach achieved accuracy of 99.3% to detect defects, with high accuracy to classify sloughed filling (100%), broken pick (97%), broken yarn (80%) and fuzzy ball (74%), but low for nep (12.5%) and cut selvage (0%).

Keywords: Fabric defects · Convolutional Neural Network · U-Net · CNN · VGG16 · Random forest · Accuracy · Recall · Precision · F1 score

1 Introduction

In the textile industry, reports on fabric defects are used as a measure of quality of a fabric roll. Such reports also serve the purpose of eliminating cause of defects, hence improve production quality. The conventional approach is human inspection to determine the location, type and size of the defects. However, this approach is prone to human errors, optical illusions and has the risk of missing small defects. It also suffers from low accuracy rate, poor consistency among inspectors, and poor efficiency. Therefore, computer enabled automated defect detection based on image processing emerged as a promising approach to improve efficiency and accuracy, and has been an active research area over the last decade. Camera systems mounted on textile production machines together with advanced data processing and machine learning technology can potentially improve accuracy of fabric defect detection and classification. However, deployment of

such systems requires significant work on development and performance improvement as compared to the current practice.

A flaw on the manufactured fabric is known as a fabric defect. Defects may be caused by textile machine/process malfunction or oils from the machinery or faulty yarns or improper mixing of dye stuff in the dye solution. Fabric defects can be classified as minor, major and critical defects. Major defects may affect the sale of the product, and critical defects may result in the entire roll to be rated as second or worse in the quality check process. There are more than 60 defect are given in the document on Standard Test Methods for Visually Inspecting and Grading Fabrics [1]. The fabric defect can be classified into seven main classes, named as yarn, weaving, isolated, pattern, wet processing, raising and milling defects. Some of these are due to surface colour change, and others are due to local texture irregularity.

It is very important to detect these fabric defects because of the financial loss concerns that may arise from faulty production [2]. Considering the fact that 85% of the faults in the clothing industry are caused by fabric faults; if the companies producing woven fabrics use an effective quality control system to produce the desired quality fabric and sell it in this way, they will have adopted the most accurate production system [3]. Traditionally, fabric defect detection is carried out by a skilled worker who tries to detect defects in 2–2.5 m length fabric rolls that are passed at a speed of 8–10 m per minute through an illuminated inspection platform. The employee who detects the error stops the machine and marks the area containing the defect and records it. At the end of this process, acceptance/rejection decision is performed according to the total defect rate in the roll. Defect detection is a long and tiring process. Defect inspectors who constantly look at the same point can audit for a maximum of 1 h due to eyestrain. In traditional systems, the error rate is still around 60%. In addition, the risk of Alzheimer's disease has increased in workers who do this type of work. For these reasons, automated/computerized fabric defect detection systems have been researched.

Automated real-time inspection system is usually composed of camera, light, im-age capturing and processing unit. Defect detection techniques vary from conventional image processing techniques to machine learning techniques. In [4], algorithms developed for detecting fabric defects were divided into three groups, namely statistical [5–7], spectral [8–10] and model-based approaches [11–13].

Recently, machine learning approaches have attracted a lot of attention and been applied in detection of fabric defects, as well as in many different fields. For example, neural network and deep learning network approaches were used in [14, 15], while the convolutional deep learning approach was used in [16–18]. In traditional convolutional deep learning models, a large dataset is necessary for successful performance. However, for some classification problems, the amount of data that can be collected is limited in practice, leading to a small dataset. It has been shown that, with small datasets, U-Net gives better results than classical models in pixel-based image segmentation problems.

In this paper, we are proposing a hybrid classification model which employs U-Net to determine whether image contains any defects. If a defect is detected, then defect classification is carried out by using a hybrid model made up of VGG16 and Random Forest. We used the AITEX dataset [19] for training and validation purposed. The results

show that the proposed model resulted in 99.8% accuracy for detecting defects which is better than those presented in the literature.

2 Dataset and Experimental Work

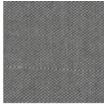


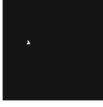


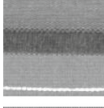

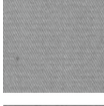

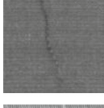



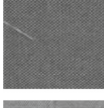





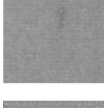

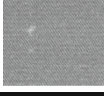

We used the AITEX fabric defects dataset [19] for training and testing our model. The original AITEX dataset consists of 245 images of 4096×256 pixels captured from seven different fabric structures. The fabrics in the dataset are mainly plain. There are 140 defect-free images, i.e. 20 images for each of fabric structure type. The remaining 105 images contain 12 different types of fabric defects which commonly appear in the textile industry. After data enhancement and decomposition of the original dataset into smaller 256×256 pixels images, we obtained a data set of 165 defected samples of 256×256 pixels. Then we carried out augmentation on the 165 samples (90° clockwise rotation, 180° clockwise rotation and 270° clockwise rotation), we obtained a set of 660 defected samples. Table 1 shows the numbers of occurrences of the defect types before and after the augmentation. Table 2 shows samples of images for different defect types and corresponding numbers of occurrences after augmentation. We used 592 images corresponding to the seven most common defect types as our data set. This dataset was randomly divided into training set (85%) and validation set (15%).

Table 1. Defective sample counts before and after data augmentation.

Defect type	Original sample count	Count after augmentation
Broken pick	55	220
Fuzzy ball	38	152
Nep	14	56
Sloughed filling	14	56
Cut selvage	9	36
Broken end	9	36
Broken yarn	9	36
Weft curling	6	24
Crease	5	20
Warp ball	4	16
Knots	1	4
Contamination	1	4

Our tests were conducted on a computer with a processor of ‘Intel Core i7-11800H’, a graphics card of ‘RTX 3070 8 GB’ and operating system of Windows10. Our source code was developed in Python programming environment.

Table 2. Enhanced AITEX fabric defects dataset.

Defect Name	Sample count	Defected sample	Grayscale mask
Broken pick	220		
Fuzzy ball	152		
Nep	56		
Sloughed filling	56		
Cut selvage	36		
Broken end	36		
Broken yarn	36		
Weft curling	24		
Crease	20		
Warp Ball	16		
Knots	4		
Contamination	4		

3 Defect Detection and Classification

A two-stage classification approach is used in the proposed model. Images of fabrics are first applied to a customized U-Net to determine whether the fabric is defected or not. Images of defected fabrics are then put through a classification process where VGG16 is used for extracting features and a random forest is used to determine defect type. Following subsections describe U-Net, VGG16 and random forest methods used.

3.1 Defect Detection Using a Customized U-Net

U-Net, a type of Convolutional Neural Networks (CNN) approach, was first developed in 2015 by Olaf Ronneberger, Phillip Fischer, and Thomas Brox to perform better segmentation on biomedical images [20]. Different from traditional convolutional deep learning models which require a large dataset for successful performance, U-Net performs well with small datasets. Since data sets available for fabric defects are small, U-Net was used for defect detection in this work.

The architectural structure of the proposed U-Net is shown in Fig. 1. The number of filters in the original U-Net has been reduced from 64,128, 256, 512, 1024 to 16, 32, 64, 128, 256, respectively. The reduction in the number of filters speeds up the training process and reduces the computational load. As seen in Fig. 1, in the encoder side or in the feature extraction side of the U-Net, different numbers and sizes of features are extracted along the convolutional neural network model, and the pooling process of their height and width dimensions is carried out using the pooling layer.

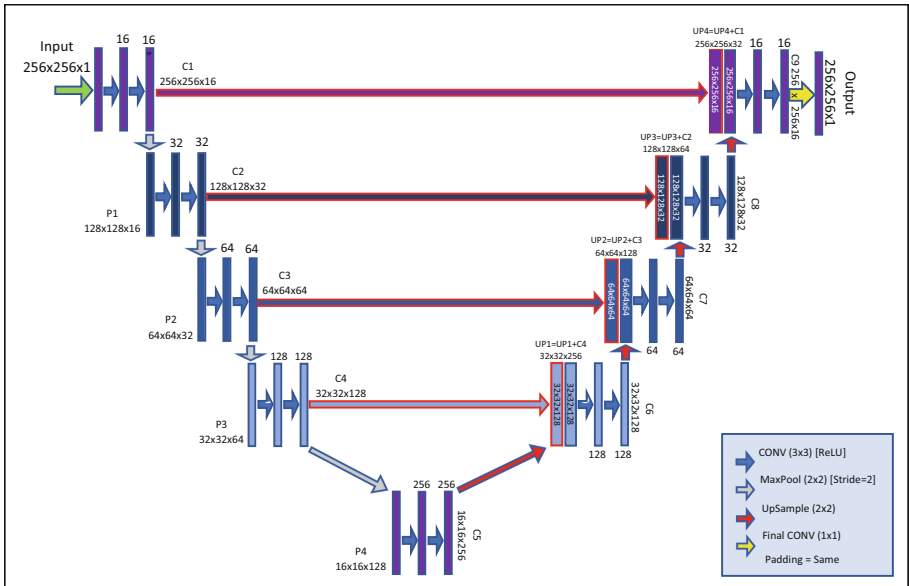


Fig. 1. Architectural structure of the proposed U-Net.

The second half of the model serves to increase the size, hence the resolution of the image at the output of the model in accordance with the input image. In order to reduce complexity and computation time, each convolution output in the Encoder part is combined with the corresponding upsampling layers.

Loss Function. The number of fabric defects is usually small and the total area affected by the defects is quite small compared to the defect-free part of the fabric. Using an ordinary loss function such as standard cross entropy may cause the model to fall into its local optimum points. To overcome this imbalance problem, cross entropy and Dice binary loss function are used together. Dice function is a frequently used performance criterion to evaluate success in biomedical images, and will be used in this work. The Cross-entropy (C) and Dice (D) loss functions are expressed as follows;

$$C(p, q) = - \sum_{x \in X} p(x) \log q(x) \quad (1)$$

$$D(p, q) = \frac{2 \sum_i^N p_i q_i}{\sum_i^N p_i^2 + \sum_i^N q_i^2} \quad (2)$$

where $p(x)$ and $q(x)$ in (1) represents distribution functions of predicted and actual sets, the values p_i and q_i values in (2) are the pixel values of prediction and ground truth, respectively.

Training Evaluation Metrics. In traditional binary segmentation problems, accuracy metric is typically used. The Accuracy metric is defined as follows

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

where (TP) represents true positive, while (TN), (FP) and (FN) stands for true negative, false positive and false negative respectively.

Since defects affect only a small area of the fabric, the number of pixels in defective regions is very small compared to that in the defect-free region. Therefore, Recall (also known as sensitivity), precision and F1 Score are taken into account as well as the accuracy in the evaluation of the model. These evaluation criteria are expressed as,

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

$$F1score = 2 * \left(\frac{Precision * Recall}{Precision + Recall} \right) \quad (6)$$

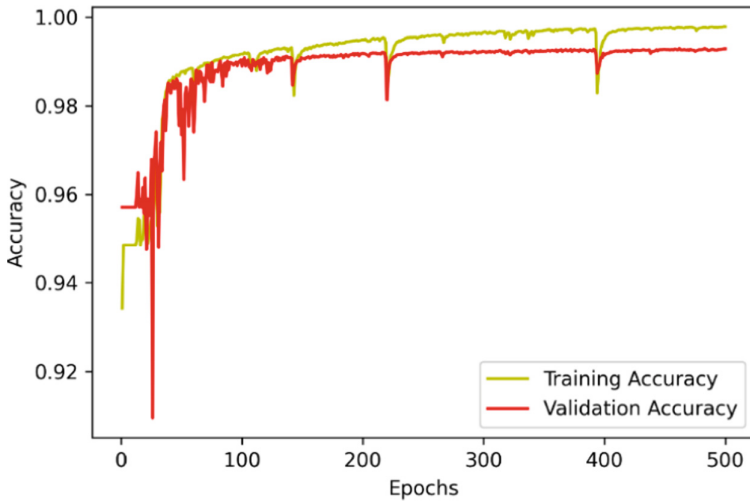
Training. We used the Adam optimizer with a mini-batch size of 16. The training settings of the model were chosen as follows: initial learning rate was set to 0.001, and exponential decay rates for the estimates of mean and variance were set to $\beta_1 = 0.9$ and $\beta_2 = 0.999$, respectively. We trained the model for 100 epochs. Table 3 lists the results

Table 3. Performance comparison of the proposed model

Model	PTIP [21]	U-Net	Mobile U-Net [17]	Improved U-Net [18]	Our U-Net
Accuracy (%)	85.6	89.3	93.5	98.3	99.2
Recall (%)	90.1	85.1	90.4	92.7	94.3

from our model and compares them with other models. Accuracy for our model is 99.2% and recall is 94.3%, both of which are superior as compared to other models.

Then, we trained the model for 500 epochs. Figures 2, 3, 4 and 5 present the results of accuracy, recall, F1-Score and precision for training and testing phases. As can be seen, the proposed model showed superior performance after being trained for 500 rounds. The accuracy was 99.8% for the training phase and 99.3% for the validation phase. Corresponding values were 98.5% and 90.2% for recall, 96.2% and 91.3% for F1, and 96.5% and 92.2% for precision.

**Fig. 2.** Accuracy performance of the proposed model.

The loss function performance figures of the proposed model for the training and validation phases are shown in Fig. 6. As can be seen in the figure, values of the loss functions were reduced to very small values, i.e. being 0.0095 for Training-Loss and 0.0529 for Validation-Loss.

We tested the trained model on numerous images from the validation data set. For this we randomly selected an image from the validation data set and applied it to the input of the model, and obtained the model output. We have seen that our model successfully detected the defects in all the images. Figure 7 shows two sample images of defective

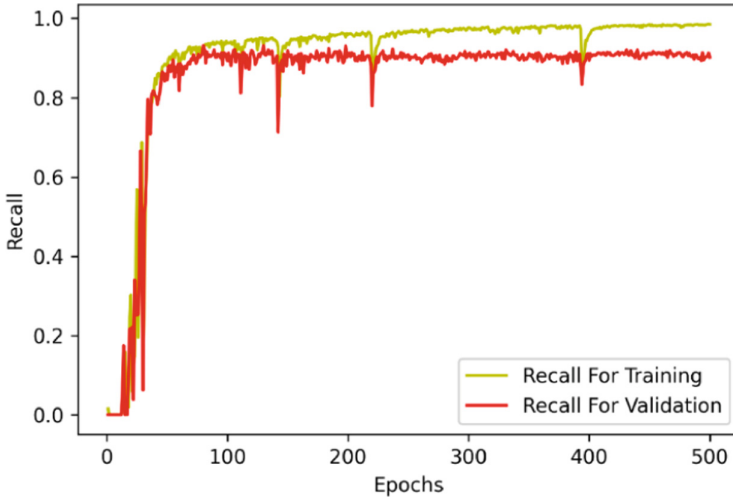


Fig. 3. Recall performance of the proposed model.

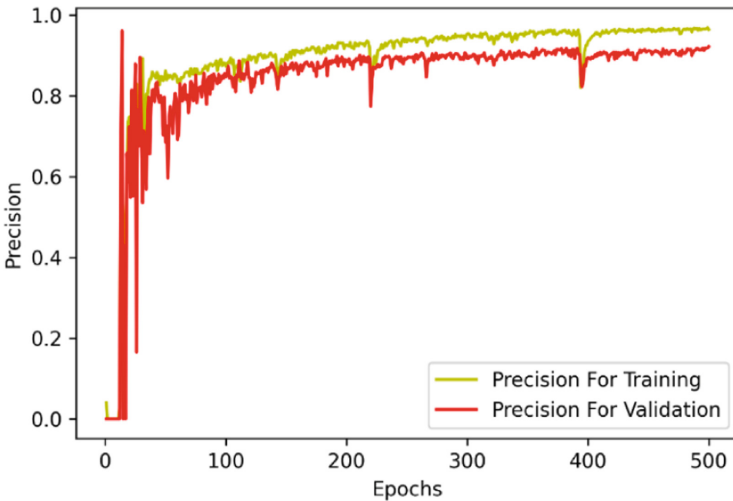


Fig. 4. Precision performance of the proposed model.

fabrics on the left column, and model output on the right, and shows that the defects were detected successfully by the model.

3.2 Hybrid Model for Defect Classification

We used a hybrid model for fabric defect classification in this work. Images of fabrics are first applied to the customized U-Net to determine whether the fabric is defected. Images of defected fabrics are then put through a classification process where VGG16 is used for extracting features and a random forest to determine defect type.

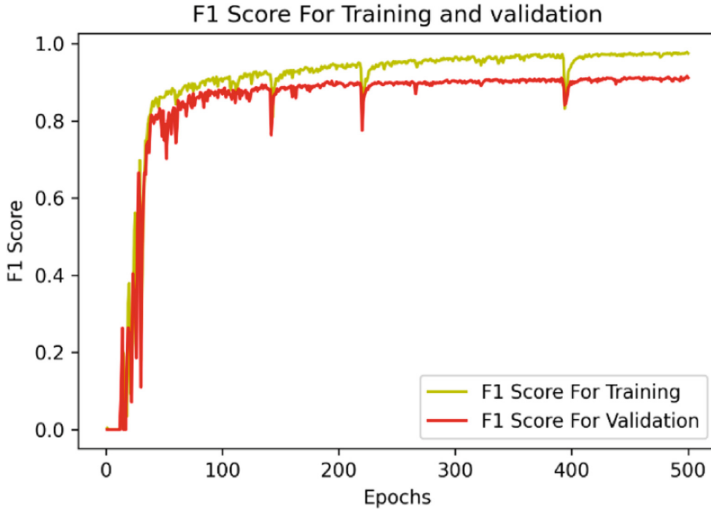


Fig. 5. F1-score performance of the proposed model.

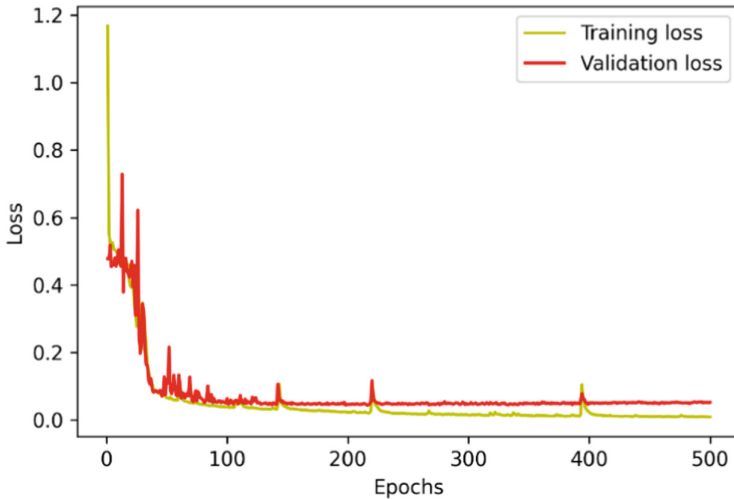


Fig. 6. Loss performance of the proposed model.

VGG16. VGG16 is a convolutional neural network model proposed in an article by K. Simonyan and A. Zisserman from Oxford University [22]. In VGG architecture, the image is passed through a series of convolutional layers which followed by three Fully-Connected (FC) layers. Filters with a very small receptive field (3×3) are used in the convolutional layers. The convolution stride is fixed to 1 pixel; the spatial padding of convolutional layer input is used to preserve spatial resolution after convolution, i.e. the padding is 1-pixel for 3×3 convolutional layers. Max-pooling follows every stack of convolutional layers, and performs spatial pooling over a 2×2 pixel window, with

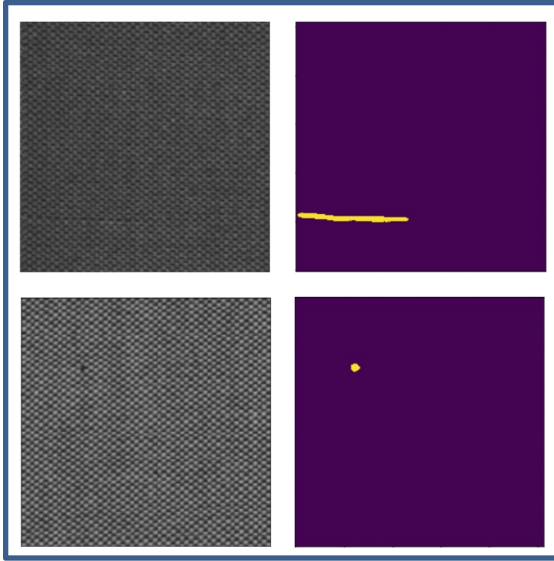


Fig. 7. Samples of defect detection test results of the proposed model.

stride 2. Three Fully-Connected (FC) layers follow a stack of convolutional layers with 4096, 4096 and 1000 channels. Each fully connected layer is followed by a nonlinear activation function, such as ReLU. The final layer is the soft-max layer which normalizes output real values from the last fully connected layer to target class probabilities. In our work, we used the VGG16 only for feature extraction, therefore the fully connected layer which used in original model for classification is removed (see Fig. 8), and the feature set of $8 \times 8 \times 512$ which obtained from the last convolution layer of VGG16 were fed to random forest for fabric defect classification.

Random Forest. Random Forest is a supervised learning algorithm. As the name suggests, it creates a forest in a random manner. The “forest” is a collection of decision trees that are mostly trained by the “bagging” method. With this method, hundreds of decision trees are created and each decision tree makes an individual prediction. The structure of a simple random forest consists of multiple decision trees as shown in Fig. 9 and combines them to get a more accurate and stable prediction. For example, in regression problems, average of the predictions of the decision trees is used. In classification problems, most voted one is chosen among the predictions. The biggest advantage of the random forest is that it does not cause an over-learning/overfitting problem that is often experienced in traditional machine learning methods.

3.3 Testing the Model

We tested our model by using images of fabrics with seven most common defect types (namely broken pick, fuzzy ball, nep, sloughed fill, cut selvage, broken end and broken

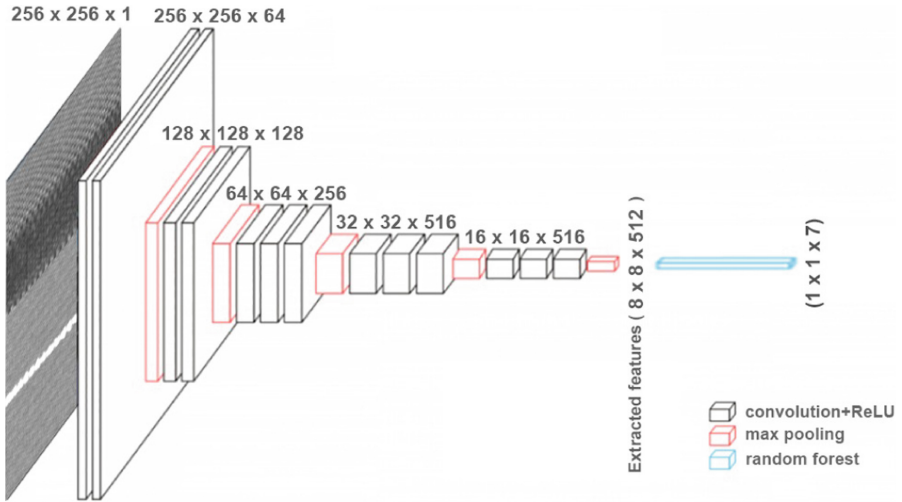


Fig. 8. Architectural structure of VGG16 used.

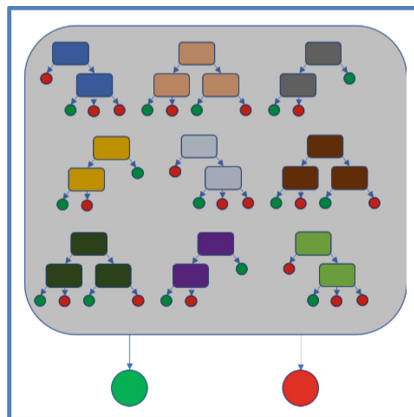


Fig. 9. Simple random forest.

yarn). We had 592 samples images for the seven most common defect types. The proposed model performed well on the limited data set of fabric defects; U-Net detected fabric defects with high accuracy (99.3%). The confusion matrix for classification performance of our model is given in Fig. 10. It can be seen that the proposed hybrid model classifies sloughed filling type with 100% accuracy, broken pick type with 97% accuracy, broken yarn type with 80% accuracy and fuzzy ball type with 74% accuracy. However, the model could classify nep type only with 12.5% accuracy, and confused this type with fuzzy ball and broken end types. The model failed to classify cut selvage type, and confused it with broken pick. High accuracy for the broken pick defect is attributed to the fact that this defect type is the most occurring type in the data set (220 out of 592 samples) and the area affected by defect is not small. The sloughed fill, although not so

common (56 out of 592 samples), due to the large areas affected by this type resulted in a very high classification accuracy. Performance was not good for the nep and cut selvage. The reason for this is thought to be that only a small area of the fabric is affected by these two types.

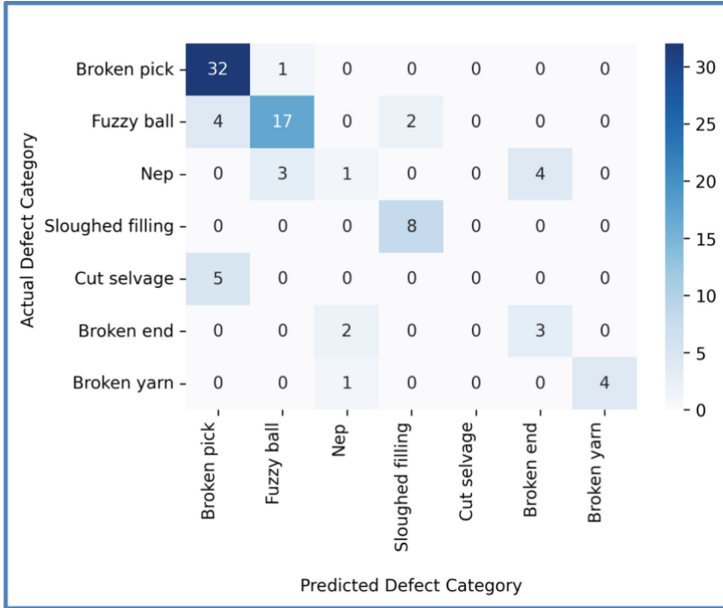


Fig. 10. Confusion matrix for classification results.

4 Conclusions

We proposed a hybrid detection and classification model for fabric defects. The model first detects defects by using a customized U-Net, and then carries out classification on defected fabrics by using VGG16 for feature extraction and random forest for classification. AITEX data set was used for training and testing purposes. We applied data enhancement, decomposition, augmentation on the data set, and obtained 660 images of 256×256 pixels for defected fabrics and used 592 of those corresponding to the seven most common defect types. The customised U-Net performed well on binary classification of defected and defect-free images, giving accuracy level of 99.3%. On the defect classification side of our model, we achieved high accuracy for defect types of broken pick, sloughed fill, broken yarn and fuzzy ball. On the other hand, exhibited poor classification performance for types of cut selvage and nep. Our results show that defects that affect large area on the fabric surface can be detected with high accuracy. Future research efforts can be directed towards improving the classification performance for defects with small size. The developed model will be evaluated by collecting data in real field trials at a company producing textile machinery in Turkey.

References

1. Standard Test Methods for Visually Inspecting and Grading Fabrics. D5430–13 (2017)
2. İzbudak, H., Alkan, A.: Denim fabric defect detection by using correlation method. In: National Conference on Electrical, Electronics and Computer Engineering, Bursa, Turkey (2010)
3. Dorrity, J.L., Vachtsevanos, G., Jasper, W.: Real-time fabric defect detection and control in weaving processes. Natl. Textile Center Ann. Report G94-2 (1996)
4. Hanbay, K., Talub, M.F., Özgüven, Ö.F.: Fabric defect detection systems and methods—a systematic literature review. *Optik* **127**, 11960–11973 (2016)
5. Zhang, Y.F., Bresee, R.R.: Fabric defect detection and classification using image analysis. *Text. Res. J.* **65**(1), 1–9 (1995)
6. Chetverikov, D., Hanbury, A.: Finding defects in texture using regularity and local orientation. *Pattern Recogn.* **35**(10), 2165–2180 (2002)
7. Jasper, W.J., Potlapalli, H.: Image analysis of mispicks in woven fabric. *Text. Res. J.* **65**(11), 683–692 (1995)
8. Tsai, D.M., Hsieh, C.Y.: Automated surface inspection for directional textures. *Image Vis. Comput.* **18**(1), 49–62 (1999)
9. Chan, C.H., Pang, G.K.: Fabric defect detection by Fourier analysis. *IEEE Trans. Ind. Appl.* **36**(5), 1267–1276 (2000)
10. Escofet, J., Navarro, R.F., GarciaVerela, M.S.M., Pladellorens, J.M.: Detection of local defects in textile webs using Gabor filters. *Opt. Eng.* **37**(8), 2297–2307 (1998)
11. Kumar, A.: Neural network based detection of local textile defects. *Pattern Recogn.* **36**(7), 1645–1659 (2003)
12. Cohen, F.S., Fan, Z., Attali, S.: Automated inspection of textile fabrics using textural models. *IEEE Trans. Pattern Anal. Mach. Intell.* **13**(08), 803–808 (1991)
13. Odemir, S., Baykut, A., Meylani, R., Erçil, A., Ertuzun, A.: Comparative evaluation of texture analysis algorithms for defect inspection of textile products. In: 14th International Conference on Pattern Recognition (Cat. No. 98EX170), vol. 2, pp. 1738–1740. IEEE, Brisbane, QLD, Australia (1998)
14. Campbell, J.G., Fraley, C., Murtagh, F., Raftery, A.E.: Linear flaw detection in woven textiles using model-based clustering. *Pattern Recogn. Lett.* **18**(14), 1539–1548 (1997)
15. Şeker, A., Peker, K.A., Yüksek, A.G., Delibaş, E.: Fabric defect detection using deep learning. In: 24th Signal Processing and Communication Application Conference (SIU), pp. 1437–1440. IEEE, Zonguldak, Turkey (2016)
16. Mei, S., Wang, Y., Wen, G.: Automatic fabric defect detection with a multi-scale convolutional denoising autoencoder network model. *Sensors* **18**(4), 1064 (2018)
17. Jing, J.F., Ma, H., Zhang, H.H.: Automatic fabric defect detection using a deep convolutional neural network. *Color. Technol.* **135**(3), 213–223 (2019)
18. Rong-qiang, L., Ming-hui, L., Jia-chen, S., Yi-bin, L.: Fabric defect detection method based on improved U-Net. *J. Phys. Conf. Ser.* **1948**(1), 012160 (2021)
19. Silvestre-Blanes, J., Albero, T., Miralles, I., PérezLlorens, R., Moreno, J.: A public fabric database for defect detection methods and results. *Autex Res. J.* **19**(4), 363–374 (2019)
20. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. Lecture Notes in Computer Science, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
21. Wang, Z., Jing, J.: Pixel-wise fabric defect detection by CNNs without labelled training data. *IEEE Access* **8**, 161317–161325 (2020)
22. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)