



Stacked Autoencoder Method for Fabric Defect Detection

Abdulkadir SEKER, Ahmet Gürkan YUKSEK

Cumhuriyet University, Department of Computer Engineering, TR-58140, Sivas, Turkey

Received: 31.03.2017; Accepted: 17.04.2017

Abstract. The fabric defect detection has crucial importance in terms of sectoral quality. As fabric defection stage, accordingly the growing market volume and production capacity, detection via human vision has caused largely time-wasting and success rate decreasing until 60%. Due to a fabric has unique texture, there is necessity for it to work on separately from other images types while extracting its features. Features are vital material of computer vision especially classification problems. Hence, extracting right features is the most significant stage of error detection. This purpose in mind on this study, deep learning which distinguishes with multi-layer architectures and reveals high achievement on image and speech procession recent years by self-feature extraction is applied to fabric defect detection. Stacked autoencoder -a deep learning method- that aimed to represent input data via compression or decompression is tried to detect defect of fabrics and it gained acceptable success. The principal aim of this study is to increase achievement of feature extraction by tuning up the input value and hyper parameters autoencoder. Thanks to the fine tuning of hyper-parameters of deep model, we have 96% success rate on our own dataset.

Keywords: Deep learning, fabric defect detection, autoencoder, hyper parameter

Kumaş Hatası Tespiti için Yığılanmış Oto-kodlayıcı Yöntemi

Özet. Kumaş hatası tespiti sektörel kalite açısından önem arz etmektedir. Bu hataların tespitinde, gelişen pazar hacmi ve üretim kapasitelerinin büyüklüğü sebebiyle insan görüşü ile tespit, büyük oranda zaman kaybına ve hata tespit oranının %60 seviyelerine kadar düşmesine sebep olmaktadır. Bu bağlamda daha yüksek başarımlar elde edebilmek için görüntü işleme alanında bir çok yöntem denenmiştir. Kumaşın kendine has bir dokusunun olması sebebiyle, öznelikleri çıkarılırken diğer görüntü türlerinden ayrı olarak incelenmesi gerekmektedir. Öznelikler bilgisayarlı görmede özellikle sınıflandırma problemlerinde ham madde olmaktadır. Bu yüzden doğru öznelikleri çıkarmak, hata tespitinde en önemli aşamadır. Bu amaç doğrultusunda, çoklu-katman mimarisi ve kendi özneliklerini çıkararak son yıllarda görüntü ve ses işleme alanında büyük başarılar getirmesi ile öne çıkan derin öğrenme kumaş hatası tespitine uygulanmıştır. Giriş verisini sıkıştırma ya da genişletme ile temsil eden yığılı oto-kodlayıcılar -bir derin öğrenme yöntemi- kumaş hatası tespitinde denenmiş ve kabul edilebilir başarımlar elde edilmiştir. Çalışmanın asıl amacı oto kodlayıcının hiper parametreleri ve giriş değeri ile oynamalar yaparak öznelik çıkarımı başarısını artırmaktır. Derin modelin hiper parametrelerin ince ayarıyla, kendi veri setimizde %96'lık bir başarı oranı elde ettik.

Anahtar Kelimeler: Derin öğrenme, kumaş hatası tespiti, oto-kodlayıcı, hiper parametre

1. INTRODUCTION

As a traditional inspection, fabric defect detection is a serial work that employees detect errors on the illuminated inspection table at higher speed of 8-10 meters per minute from unrolling the fabric rolls. This inspection type is either exhausting or low-accuracy job. Besides, looking same direction

* Corresponding author. Email address: aseker@cumhuriyet.edu.tr

continuously and following the roll are caused by Alzheimer disease. While employee detecting defect after one hour, his efficiency and error hit ratio decreases. For these reasons automatic fabric defect detection systems are started to develop. Most of computer vision methods are tried on defect detection task. Thanks to hardware development and programming opportunity based on GPU, number of hidden layer (HL) size can increase in neural networks model, it calls as deep learning. It is a new research area on machine learning. It reveals many successful works with consisting multi hidden layer architectures. Deep networks applied on most of application area in signal and speech processing and just after image processing tasks. Due to constructs a deep network, first rule is having a large dataset for extracting distinctive features.

In previous studies, many methods such as wavelet transform, Gabor transform, co-occurrence matrix and shallow neural networks implemented in defect (fabric or other task) detection. In our study, we apply a deep network method called as stacked autoencoder on this issue. Autoencoder is a neural network model which try to repeat input value in its output. The input data is compressed or expanded in HL, then it is expanded or compressed again in output layer. The more output is similar to input; the better features will be extracted from autoencoder. Network will not learn from pairs of training data-its labels. It will learn the features of own data. For this reason, HL is indeed a feature detector. Stacked autoencoder construct as increase the number of hidden layer. It means input value will compress more than one. The essential aim of our study is to increase achievement of feature extraction by tuning up the input value and hyper parameters of stacked autoencoder.

2. FABRIC DEFECT AND DETECTION

The term defect means an imperfection that harms worth or utility. Other definition is absence of something necessary for completeness or perfection. As general terms, defect is a fault that spoils the material. Fabric is produced textile fibers which consist of materials such as cotton, wool, nylon, polyester, or a mix of these elements [1]. A fabric defect is any failure or irregularity in the fabric that unacceptably large defect by the consumer [2]. Fabrics are separated into several classes according to raw materials or producing methods. For this reason, defects are different according to each types.

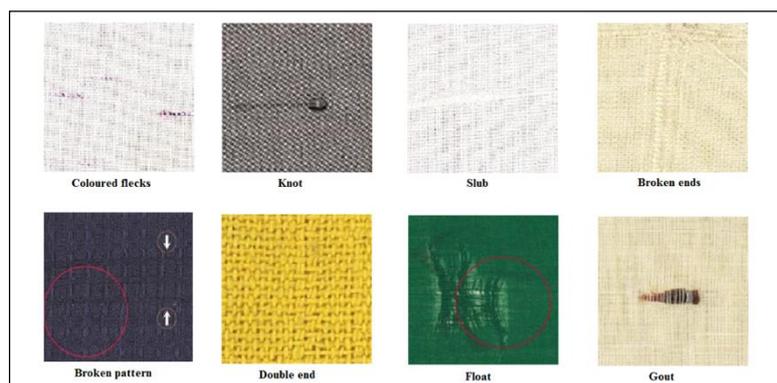


Figure 1. Some of defects on woven fabrics

Woven fabric has defects such as knots, colored flecks, slubs. On knitted fabric, there are defects such as bariness, bunching up, stripes [3].

Stacked Autoencoder Method for Fabric Defect Detection

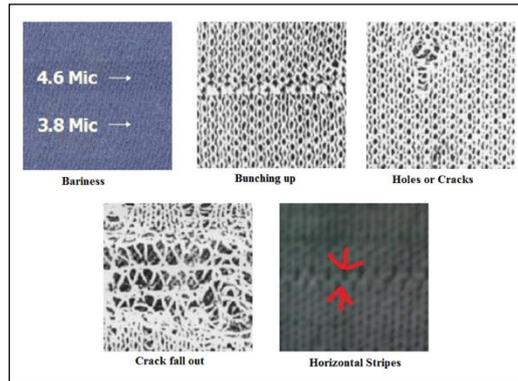


Figure 2. Some defects on knitted fabrics.

Inspection is a part of detection and fixing errors and it is visual examination of a fabric. Fabric defect detection is a crucial stage of quality control in the textile sector. Woven fabric quality based on yarn properties, after that weaving constructions and mechanism [4]. As long as consumers or manufacturers expect more quality from fabrics, they have become more conscious of poor quality issues. It is crucial to avoid defects for avoid rejection of fabrics from customers. Contained defect fabrics' prices are decrease to until 45%-65% [5]. Hence presence of fabric defect, it caused to reduce prices, even in sometimes, it creates a high decrease in value to the manufacturer. Due to reduce value loss because of presence defects in fabric, manufacturer try to avoid defects by taking precautions if it is possible. Detecting fabric can carry out in two ways. First choice is in the weaving process (or its stages) can be checking for the presence of defects, it is called as process inspection. The second way is the product inspection. It detects for the defects on the manufactured fabric. Nowadays, manufacturers focused on second way. At the output of weaving step, fabric produces as two-meter-width. It passes on inspection table at the speed of 0.3-0.5 meters per minute [6]. The slow speed of manufactured fabric is insufficient to keep a human inspector occupied. Also, the relatively hostile working environment near the weaving machines is not suitable for human inspection. Therefore, product inspection is not performed at the same time with the production stage.



Figure 3. Traditional fabric defect inspection table.

The traditional inspection for fabric defect detection is fulfilled via human eyes. Its procedure is unrolling the fabric rolls that is brought from the weaving machines or supplier firms on the illuminated inspection table at higher speed of 8-10 meters per minute. When the inspector (human) catch sight of a defect on the moving roll, he stops the motor which gets the fabric roll moved. After signs the defect with a sticker and records its location, type and size, starts the motor again. In this way, inspector completes to scanning whole roll and calculates total defect rate. If the rate of defect in a roll higher than 10% (the rejection rate up to customer and importance of fabric usage), this fabric roll rejected. Because of there are several defect types, missing defect rate also increases. Even with highly trained quality control staffs, only about 70% of the defects being detected [7]. Because of these disadvantages, the researchers try to find automatic detection systems. The automated fabric inspection system has lower cost because of reduction in personnel and it provides higher accuracy of defect detection.

3. DEEP LEARNING ALGORITHMS

Deep learning (deep machine learning, or deep structured learning, or hierarchical learning, or sometimes DL) is a machine learning methods based on a set of algorithms that attempt to model high-level abstractions in data by using multiple processing layers (multi HL in neural nets) with complex structures or otherwise, composed of multiple non-linear transformations [8]. Success of whole machine learning methods increase with the right choice of features. Because that, many of algorithms need preprocessing about the representation of data. However, current machine learning algorithms have weakness about features. Researcher want to get rid of this dependency and they think that with Artificial Intelligence, learning representations of the data easier extract to useful information when building classifiers or other predictors. The idea of deep learning is relative tightly with representation learning. Deep learning methods goal is yielding more abstract – and ultimately more useful– representations [9].

As the speed of GPUs increased rapidly, it was soon possible to train deep networks such as convolutional networks without the help of pretraining as demonstrated by Ciresan and colleagues in 2011 and 2012 who won character recognition, traffic sign, and medical imaging competitions with their convolutional network architecture [10]. Krizhevsky, Sutskever, and Hinton used a similar architecture in 2012 that also features rectified linear activation functions and dropout for regularization [11]. They received outstanding results in the ILSVRC-2012 ImageNet competition [12], which marked the abandonment of feature engineering and the adoption of feature learning in the form of deep learning. Google, Facebook, and Microsoft noticed this trend and made major acquisitions of deep learning startups and research teams between 2012 and 2014. From here, the researches accelerate rapidly in deep learning.

Deep learning algorithms applied on several areas of computer science. Charissa Ann Ronao and Sung-Bae Cho propose a deep convolutional neural network for Human Activity recognition (HAR) using smartphone sensors and their algorithms achieved an almost perfect classification on moving activities [13]. In another study about NLP, stacked autoencoder applied on unsupervised extractive summarization of email with excellent performance. Summaries are highly informative and semantically similar to human abstracts [14]. On the robotic area, Ian Lenz, Honglak Lee and Ashutosh Saxena are using deep learning methods in order to solve the problem of detecting robotic grasps in an RGB-D view of a scene containing objects [15]. Bioinformatics area, Deep learning using for recognizing disease from DNA sequences [16] or predicting protein secondary structure [17]. Another state-of-art study is on largescale video classification with Convolutional Neural Networks (CNN) using a new dataset of 1 million YouTube videos belonging to 487 classes [18]. As is seen, during last decade, deep learning algorithms have been applied to almost every field.

3.1 Restricted (Deep) Boltzmann Machines

Restricted Boltzmann Machine (RBM) is a model developed from Boltzmann machines. Neurons aren't fully connected as Boltzmann machines. A restricted Boltzmann machine with no hidden-to-hidden and no visible-to-visible connections [19].

3.2 Autoencoders

An autoencoder is typically a feedforward neural network which goals to learn the best features from a compressed representation of input data [20].

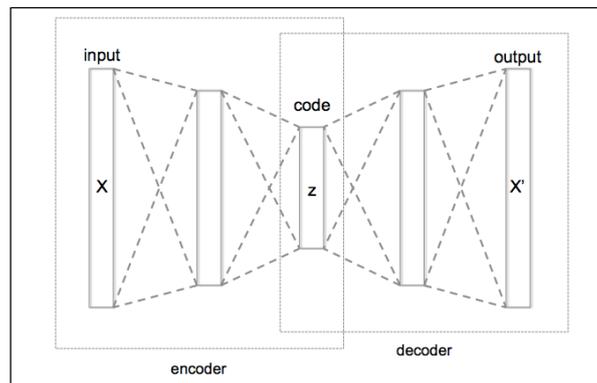


Figure 4. Schematic structure of an autoencoder.

Another definition is roughly, autoencoder is a neural network try to output own input of network. Therefore, the number of nodes must be same input and output. When the number of node in hidden layer less than input, autoencoder represent a compressed input data. However, another use is to enlarge the given input's representation. Say that the training data consists of “25-r”¹ grayscale images and it means the value of each pixel is one input layer neuron (i.e., the input layer will have 625 neurons). According to definitions above, the output layer would have the same number of units (625) as the input layer. The schematic structure of an autoencoder as seen in Figure 4. The input(x) is compressed (encoded) to z in hidden layer. Then, it is decompressed (decode) to x in output layer. The weights of nodes are tuned with each iteration until the input values are obtained on output layer. In this case, the input value is represented by a z-dimensional feature vector (dimension reduction).

3.3 Deep Belief Networks

Deep Belief Networks (DBN) is a neural network created by stacking of RBMs. DBNs are graphical models that learn to a deep hierarchical representation of the training data [21].

3.4 Convolutional Neural Networks

Convolutional Neural Networks (CNN) is consisted of one or more convolutional layers (often with subsampling step) and fully connected layer that comes after convolutional ones. The CNNs develop for using advantages of the 2D structured input image or speech signals [22]. This is attained with local connections and tied weights followed by some form of pooling which results in translation constant

¹ The “r” symbol is used for resolution. For example; “25-r” represents for “25x25 images”

features. Besides, CNN is trained easier compare to fully connected networks have the same number of hidden layer with itself.

4. MATERIALS and METHODS

4.1 Dataset

Initially, defective fabric samples were obtained from mattress ticking manufactory. The samples were woven fabrics, and they had white color and plain fabric. Photographs of these samples were taken with a DSLR camera from different angles. We setup an experiment environment. Firstly, fabrics were taken photos from opposite direction and 20 cm distance (Figure 5). But these photos had no distinctive texture, the knots weren't seen sufficiently.



Figure 5 The first (left) and second (right) environment of taking photos.

All fabrics were ironed and mounted to carton with punches, then we took again photos from above of fabric and closer than first try (Figure 5). While we were taking photos, we used manual setting due to catch the best images. We tuned three setting; aperture, shutter speed and ISO. Aperture means that “the hole in the lens” and it is needed to focus. As long as hole is larger, more light gets in. We set to 16f for aperture. Shutter speed indicates the time of shutter opened, and we set it to 1 second. The photographs are our source fabrics; they are abbreviated as SF in this paper. After the elimination, according to quality and specificity, there are totally 10 unique images of defective and perfect (two of them perfect) fabric pieces (Figure 6).



Figure 6. Some of the sample defective fabrics.

Because of the number of samples are inadequate for training, it has planned to retrieve small patches from selected images. Patches size are 100-r, 50 50-r and 25-r.

4.2 Dataset Parameters

First parameter of dataset is the patches that extracted from SF are varied in different size. When deep model setting up, input size depends on patches' size. After the several test with different size of patches, it has been decided to optimum patch size.

The second parameter is resizing value of SF. As seen in Figure 7, if resizing isn't done on the SF, the 25-r samples has no distinctive information about texture. Therefore, it has tried with several resize parameters while creating dataset task. The same size patches which created by different resize parameters are includes variable-density information. There are patches of a 100-r sample taken from same region of SF with resize parameter respectively 0.2-0.4-0.8-1(no resize) in Figure 7.

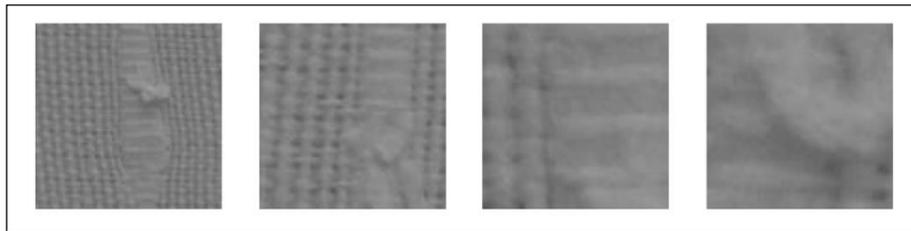


Figure 7. Patches are obtained from same region with different resize parameters.

4.3 Hyper-parameters

Machine learning algorithms has some parameters are called as “hyper-parameters”. There is difference between hyper-parameter and parameter. Hyper-parameters are set by user to optimization. Model learns parameters in time after some iterations. Hyper-parameters are often tuned by user or optimized by some “hyper-learner”.

In our study there are also several hyper-parameters. These parameters are iteration number, input size, input type, hidden layer (HL) number and node number of hidden layers. It is aimed to find the optimum hyper-parameters in some way.

- Iteration number : It means how many times network runs. The iteration which is needed to learn. It changes according to length of input data. The bigger ones, learn later.
- Input Size : Node number of input data. For example, the input size is 10000 for 100-r image. It is crucial parameter for learning.
- Input Type : It means; input has how much detail information. If resize parameter of SF is smaller, patches taken from SF have more detail information. On contrary, patches are more regular, this situation provides that defects are seemed more obviously.
- Hidden Layer : The hidden layer node number is last hyper-parameter of our model. It is important to represent input data more distinctive. We aimed to find an optimum proportion between input and hidden layer.

4.4 Implementation

It has been implemented with deep learning idea for detection of fabric defect. It is thought that this neural network that setting up with multi-hidden-layer can be give satisfying results on classification problem of fabric which has sophisticated texture. Due to train this types of neural network, one way is train only one layer at the same time. Accordingly, stacked autoencoder model has been used in this study.

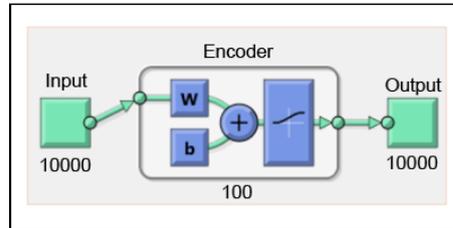


Figure 8. Basic model of autoencoder with 100 nodes in hidden layer.

While autoencoder training is carried out, there are two choices that compression or expansion. Input is compressed in the hidden layer, consequently it is expanded to obtain same thing with input. In the output of hidden layer, it will be gained compressed or abstract information from input as seen Figure 9 or 11. This information will use as features that representation of input data.

An example is given below, autoencoder will be test on synthetic digit images. The synthetic images have been produced by applying random affine transformations to digit images have different fonts. When the test results on 5000-sample-dataset in which each sample is 28-r with, supposed features can see easily. Figure 9 shows that distinguishing features are extracted from autoencoder.

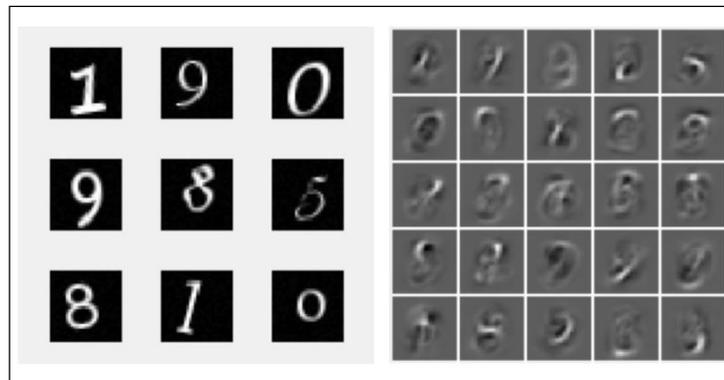


Figure 9 Synthetic digit samples (left), some of the features are obtained from autoencoder hidden layer's output (right).

The first issue of this stage is input size of autoencoder. Because the more input size is chosen correctly, the more distinctive information will be extracted. It is started with 100-r samples; it means input size has 10000 nodes. When output of first HL visualized, it is supposed to extract distinguishing parts which can be use as feature. The HL size in the first train, it is chosen as 100. It means 10000-node-input is tried to represent with 100-node-output. As a result of several experiment with different iteration number, there is no sufficient results. When hidden layer's output is visualized, thing like noise has been seen (Figure 10). We increase the number of iterations (100, 200, 400 and 800), but all features are just like obtained ones on Figure 10.

Stacked Autoencoder Method for Fabric Defect Detection

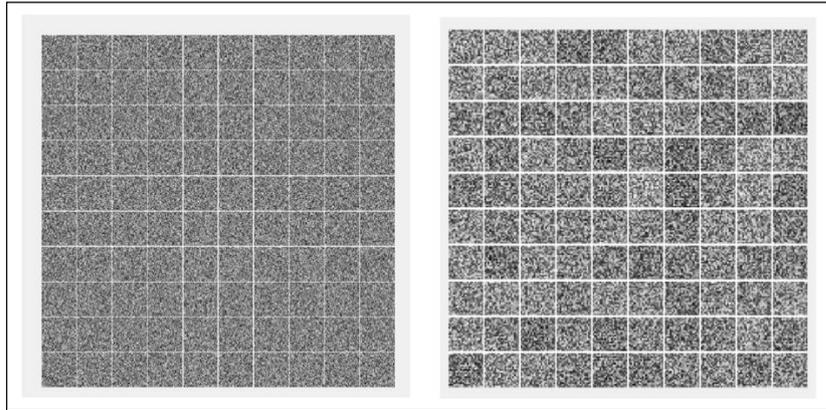


Figure 10. Obtained features from first hidden layer.

Afterwards, in order that decide to correct input layer size, other test has been done with 50-r and 25-r samples. According to output of first hidden layer, the most distinctive features have extracted 25-r samples (Figure 11). The tests also have been done different iteration numbers, but similar results are found.

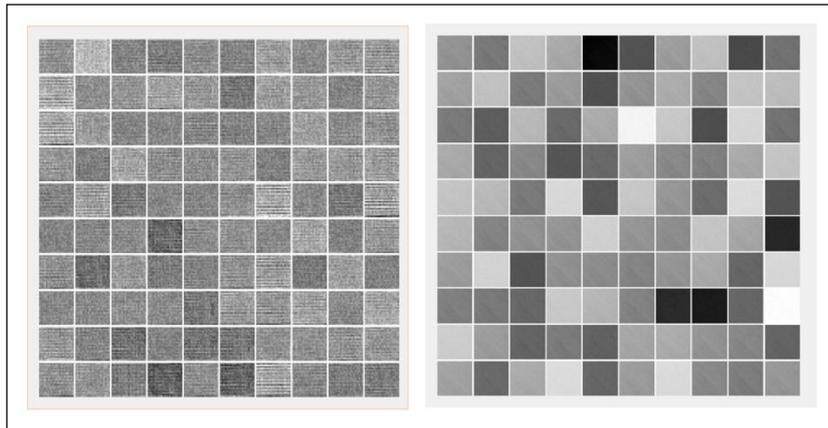


Figure 11. Visualized features of 1.hidden layer's output for 50-r and 25-r (right).

After decide to 25-r sample as an input, it is needed to find optimum resolution of image samples which enter network after labelled can called as pre-input. For this reason, datasets are prepared with 100-r, 200-r, 300-r 400-r samples and resize parameter 1, 0.8, 0.6, 0.4, 0.2 by a script that generated them automatically via 4-days-work. Thus, 20 group of samples can be input for the autoencoder. As analyze results of all data sets, we decide to optimum one. After all test, we have some opinions according to all results are analyzed;

- The optimum iteration number is 400.
- The best distinctive resize parameter is 0.2 or 0.4
- 100-r and 200-r samples give more valuable results.
- 300-r and 400-r samples is losing information while compressing to 25-r.

Due to the outputs of a network to be interpretable, it is highly desirable for those outputs to lie between zero and one and to sum to one. The aim of the softmax activation function is to enforce these constraints

on the outputs. Train a softmax layer to classify the 50-dimensional feature vectors. Unlike the autoencoders, you train the softmax layer in a supervised fashion using labels for the training data. The model has ready to classify hereby; it will be trained a final layer to classify these 50-dimensional vectors into two fabric classes. After including the final softmax layer, the setting up deep model has been completed. The model has two encoder layers and one softmax layer (Figure 12).

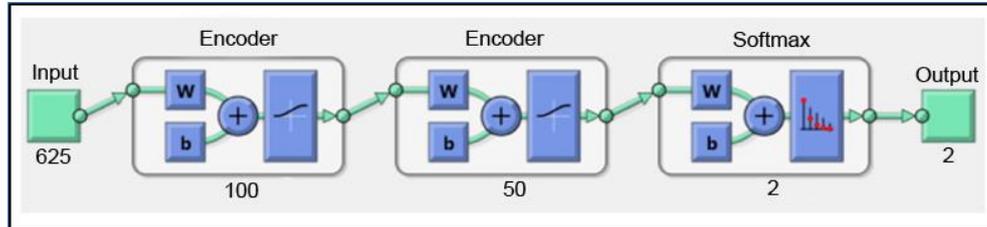


Figure 12. Stacked/Deep an autoencoder model.

5. RESULTS

We show the succession rate on the confusion matrix [23]. Confusion matrix shows the quality of learning. Rows indicate for predicted class, columns demonstrate what the actual class of samples is. The diagonal cells show number and percentage of the examples the trained model correctly predicted the classes. The other cells indicate where the classifier has made mistakes. The column on the far right of the matrix shows the accuracy per estimated class and the row at the bottom of the matrix shows the accuracy per actual class. The cell in the bottom right of the matrix demonstrated the overall accuracy.

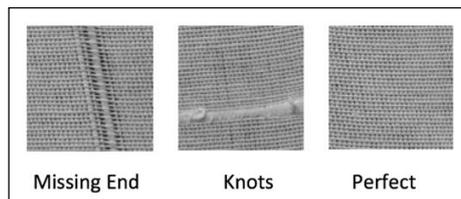


Figure 13. Classes of fabric patches in the dataset.

The first test is performed on 1400 samples which separated into three classes as missing-end, knots and perfect fabrics as seen in Figure 13. There are approximately 300 defective samples in this dataset. The success rates are for missing end 86%, knots 36%, perfect 96%. The total accuracy is 73%. Afterwards the bad results of first test, it was observed that the dataset isn't sufficient in order to recognize three distinctive classes. For this reason, we will work on only two classes as defected and perfect. Therefore, the dataset is generated with new photos, and the number of samples increase to 5000 (taken 500 patches from per SF). Training has done with these samples and test results is below confusion table.

Stacked Autoencoder Method for Fabric Defect Detection

Output Class	1	31 31.0%	3 3.0%	91.2% 8.8%
	2	9 9.0%	57 57.0%	86.4% 13.6%
		77.5% 22.5%	95.0% 5.0%	88.0% 12.0%
		1	2	Target Class

Figure 14. Confusion matrix with new extended dataset.

The test patches are generated from SF which model did not meet before. The total accuracy increase to acceptable level with 88% as seen in Figure 14. We check all test samples up and notice that the deep model doesn't recognize *missing end defect*. We extracted these samples and create the last dataset without samples have missing end defect. When the compress inputs which have missing end defect are observed, it is noticed that the compression process (100 to 25) caused to lose distinctive features of missing end. The last test was executed with this new dataset and results below are obtained as shown in confusion matrix (Figure 15). Final accuracy has reached to 96%.

Output Class	1	43 34.4%	3 2.4%	93.5% 6.5%
	2	2 1.6%	77 61.6%	97.5% 2.5%
		95.6% 4.4%	96.2% 3.7%	96.0% 4.0%
		1	2	Target Class

Figure 15. The last confusion matrix of the data set that does not contain the missing end samples.

6. CONCLUSION

We implemented a deep learning method –stacked autoencoder- for detection defects on woven fabrics. We found that how autoencoder works better with optimum parameters and explored deep autoencoder method, implement our deep model. Own fabric defect dataset has been created and decided parameters of dataset. In addition, we showed that the results of implementations (three-class and two-class model). According to these models, we realized that our dataset was not enough to recognize defect types, it can use only for anomaly (defect) detection. We decided to optimum input value resolution is 25-r. Due to manually labelled samples from 25-r, we generated dataset larger resolution patches. After labelling process has handled, we resized them to 25-r in order to model works better. According to our test

results, it was concluded that the optimum input value is 100-r and resize parameter of SF is 0.2. As hyper parameters of deep model, we tuned hidden layer size and iteration number. It has seen that 400 iteration number was sufficient to train. Deciding to hidden layer size, the error curves has been examined; it was observed that 100 hidden node gave the best results. In the results of two-class implementation with optimum parameters, we have reach 96% total accuracy on our own generated dataset.

REFERENCES

- [1] H. Y. T. Ngan, G. K. H. Pang, and N. H. C. Yung, "Automated Fabric Defect Detection-A review," *Image Vis. Comput.*, vol. 29, no. 7, pp. 442–458, Jun. 2011.
- [2] K. Kaur, N. Gupta, and K. Adhikary, "An Automatic Method to Inspect Discontinuities in Textile," *IJCSET*, vol. 1, no. 8, pp. 496–498, 2011.
- [3] Hitesh Choudhary, "Fabric Defects in Woven and Knitted Fabric," 2012. [Online]. Available: <https://www.slideshare.net/hiteshhobbit/fabric-defects-11884107>. [Accessed: 20-Mar-2017].
- [4] Ö. Kisoğlu, "Kumaş Kalite Kontrol Sistemleri," *Pamukkale Üniversitesi Mühendislik Bilim. Derg.*, vol. 12, no. 2, pp. 233–241, 2006.
- [5] A. Kumar, "Computer Vision Based Fabric Defect Detection: A Survey," *IEEE Trans. Ind. Electron.*, vol. 55, no. 1, pp. 348–363, Jan. 2008.
- [6] K. V. N. Kumar and U. S. Ragupathy, "An Intelligent Scheme for Fault Detection in Textile Web Materials," *Int. J. Comput. Appl.*, vol. 46, no. 10, pp. 975–8887, 2012.
- [7] J. L. Dorrity, "Real-Time Fabric Defect Detection And Control in Weaving Processes," 1995.
- [8] Y. Bengio, "Learning Deep Architectures for AI," *Found. trends® Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.
- [9] Y. Bengio, A. Courville, and P. Vincent, "Representation Learning: A Review and New Perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, 2013.
- [10] D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber, "A committee of Neural Networks for Traffic Sign Classification," in *The 2011 International Joint Conference on Neural Networks*, 2011, pp. 1918–1921.
- [11] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving Neural Networks by Preventing Co-adaptation of Feature Detectors," *Neural Evol. Comput.*, Jul. 2012.
- [12] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
- [13] C. A. Ronao and S.-B. Cho, "Human Activity Recognition with Smartphone Sensors Using Deep Learning Neural Networks," *Expert Syst. Appl.*, vol. 59, pp. 235–244, Oct. 2016.
- [14] M. Yousefi-Azar and L. Hamey, "Text Summarization Using Unsupervised Deep Learning," *Expert Syst. Appl.*, vol. 68, pp. 93–105, Feb. 2017.
- [15] I. Lenz, H. Lee, and A. Saxena, "Deep Learning for Detecting Robotic Grasps," *Int. J. Rob. Res.*, vol. 34, pp. 705–724, 2015.
- [16] B. Alipanahi, A. DeLong, M. T. Weirauch, and B. J. Frey, "Predicting the Sequence Specificities of DNA and RNA Binding Proteins by Deep Learning," *Nat. Biotechnol.*, vol. 33, no. 8, pp. 831–838, Jul. 2015.
- [17] Y. Wang, H. Mao, and Z. Yi, "Protein Secondary Structure Prediction by Using Deep Learning Method," *Knowledge-Based Syst.*, vol. 118, pp. 115–123, Feb. 2017.
- [18] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale Video Classification with Convolutional Neural Networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.
- [19] R. Salakhutdinov and G. Hinton, "Deep Boltzmann Machines," in *International Conference on*

- Artificial Intelligence and Statistics*, 2009, pp. 3–11.
- [20] A. Krizhevsky and G. E. Hinton, “Using Very Deep Autoencoders for Content Based Image Retrieval,” in *European Symposium on Artificial Neural Networks*, 2011, pp. 489–494.
- [21] G. E. Hinton and R. R. Salakhutdinov, “Reducing the Dimensionality of Data with Neural Networks,” *Science (80-.)*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.
- [22] S.-C. B. Lo, H.-P. Chan, J.-S. Lin, H. Li, M. T. Freedman, and S. K. Mun, “Artificial Convolution Neural Network for Medical Image Pattern Recognition,” *Neural Networks*, vol. 8, no. 7–8, pp. 1201–1214, Jan. 1995.
- [23] M. J. Brusco and J. D. Cradit, “Graph Coloring, Minimum-diameter Partitioning, and the Analysis of Confusion Matrices,” *J. Math. Psychol.*, vol. 48, no. 5, pp. 301–309, Oct. 2004.