Journal of Scientific & Industrial Research
Vol. 82, January 2023, pp. 128-134
DOI: 10.56042/jsir.v82i1.69928

MobileNetV2-based Transfer Learning Model with Edge Computing for Automatic Fabric Defect Detection

Lakshmi Ramani Burra[1]*, Karuna A[2], Srinivasarao Tumma[3], Krishna Marlapalli[4] & Praveen Tumuluru[5]

[*1]Computer Science and Engineering, Prasad V. Potluri Siddhartha Institute of Technology, Vijayawada 520 007, Andhra Pradesh, India

[2]University College of Engineering Kakinada (A), Jawaharlal Nehru Technological University Kakinada 533 003, Andhra Pradesh, India

[3]Computer Science and Engineering, Seshadri Rao Gudlavalleru Engineering College, Gudlavalleru 521 356, Andhra Pradesh, India

[4]Computer Science and Engineering, Sir C R Reddy College of Engineering, Eluru 534 007, Andhra Pradesh, India

[5]Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram 522 302, Andhra Pradesh, India

Received 7 June 2022; revised 22 September 2022; accepted 20 October 2022

In textile manufacturing, fabric defect detection is an essential quality control step and a challenging task. Earlier, manual efforts were applied to detect defects in fabric production. Human exhaustion, time consumption, and lack of concentration are the main problems in the manual defect detection process. Machine vision systems based on deep learning play a vital role in the Industrial Internet of things (IIoT) and fully automated production processes. Deep learning centered on Convolution Neural Network (CNN) models have been commonly used in fabric defect detection, but most of these models require high computing resources. This work presents a lightweight MobileNetV2-based Transfer Learning model to assist defect detection with low power consumption, low latency, easy upgrade, more efficiency, and an automatic visual inspection system with edge computing. Firstly, different image transformation techniques were performed as data augmentation on four fabric datasets for the model's adaptability in various fabrics. Secondly, fine-tuning hyperparameters of the MobileNetV2 with transfer learning gives a lightweight, adaptable and scalable model that suits the resource-constrained edge device. Finally, deploy the trained model to the NVIDIA Jetson Nano-kit edge device to make its detection faster. We assessed the model based on its accuracy, sensitivity rate, specificity rate, and F1 measure. The numerical simulation reveals that the model accuracy is 96.52%, precision is 96.52%, recall is 96.75%, and F1-Score is 96.52%.

Keywords: Deep learning, Edge devices, Industrial IoT, Modeling, MobileNetV2

Introduction

Industry 4.0 (I4.0) involves changing factories from legacy systems to innovative mechanisms and intelligent machines, enabling digital factories, and eventually establishing an associated place of work and business ecosystem. The I4.0 intends to create intelligent, networked secure value chains by digitizing critical functional operations. By switching corporate processes and business models to I4.0, industries can save money and add new significance to the value chain. Most industrial manufacturing process companies desire intelligent and sustainable production systems to reduce product defects with more benefits. This process entails identifying the problem and designing a solution to prevent it from recurring.

In the transition to I4.0, the textile manufacturing industry needs to develop its approach to transform the manufacturing process. Various studies on Fabric Defect Detection (FDD) have been carried out to enhance detection efficiency. Statistical, spectrum, model, and learning-based analysis are the most commonly used methods.[1-3] The Statistical approach focuses on some statistical information about pixels, and the structural approach uses the primary essential qualities of texture's primitive elements. Spectral approaches employ structural and frequency data to assess the existence of fabric defects. Model-based approaches create a picture and retain the data reflecting its complete texture, whereas learning-based approaches use labeled data to identify errors. However, the traditional approaches, on the other hand, rely on human vision analysis, shown in Fig. 1(a), which is time-consuming and slow productivity.

Deep Learning (DL) algorithms have improved significantly over the years due to substantial growth in the computational power of Graphics Processing Units (GPUs). The DL algorithms have been helpful

*Author for Correspondence
E-mail: ramanimythili@gmail.com

for detection and classification problems, but there are still certain limitations in their application in specific industries.[4,5] Fabric defects are defined in the textile business as any distortion in the fabric that leads the consumer to reject the product. An automatic FDD system is desired to reach the industry's necessities, such as low latency response, power usage, storage, operational efficiency, and real-time defect detection on the production line.[6] This work provides an FDD method with Edge Computing (EC), i.e., an automatic system analysis shown in Fig. 1(b), to achieve the desired requirements.

Related Work

Increased automation, production flexibility, quick response to consumer needs, and enhanced quality are significant ways to increase the textile industry's profitability. The related work of FDD in this section is based on traditional, model-based, and learning-based approaches, which have been extensively used in current years and have attained results in different industrial areas.

A unique plan, Fabric Defects Analysis System (FDAS), has been proposed for defect classification in woven textiles based on visually measurable defects and does not require prior knowledge.[7] The approach central spatial frequency spectrum is presented to increase the analysis process efficiency and detect structural defects in fabrics.[8] A successful automated fabric inspection system in fabrics is proposed for multi-class defect detection and fabric classification, with the information of geometrics and texture to capture visual attributes.[9] The study proposes using the Curvelet Transform (CT) and Gray Level Coevent Matrices (GLCM) to distinguish essential edges from the noise and find latent texture defects.[10] This is a novel method for detecting faults in fabric images that combines CT and GLCM.[11] These methods provide a valid descriptive basis of fault textures from diverse images, and in addition, the algorithms are very resistant.

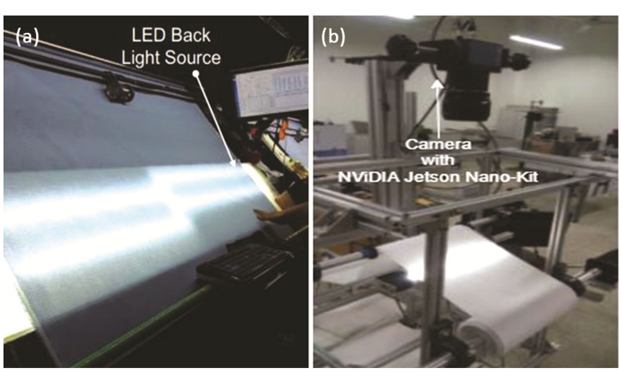

Fig. 1 — Defect detection system (a) Human vision analysis (b) Automatic system analysis

Several components of the manufacturing lifecycle, including model, plan, assessment, making, process, and maintenance, have been thoroughly examined using learning-based methodologies.[12] DST-PCA is a new feature extraction method for detecting knitting fabric faults such as holes, gouts, needle damages, and press-off.[13] The features are retrieved using the Discrete Shearlet Transform (DST) and then optimized for a three-layer ANN using Principal Component Analysis (PCA). A visual saliency–based defect identification technique was described to detect fabric flaws in textured and non-textured images. The approach extracts histogram features from Context-Aware (CA) saliency maps, subsequently input into an SVM for categorization.[14] The FDD technique MSCDAE, based on multi-scale convolutional denoising auto-encoder networks, uses the residual reconstruction maps provided by the CDAE networks to highlight problematic regions, increasing the model's robustness.[15]

Smart manufacturing relies on computational intelligence to provide compact visions for better decision-making during manufacturing cycles and product quality checks.[16] Because of their ability to automatically extract characteristics from raw data and recognize them, CNNs have recently grown in popularity and play a key role in intelligent manufacturing for image analysis.[17,18] The viability of utilizing Deep Convolutional Neural Network (DCNN) models like VGGNet, DetectNet, and GoogLeNetto to detect flaws in fabric has been established with the highest F1-score (> 0.95).[19–25]

Using a fabric fault detection approach based on the CenterNet, the feature map is extracted using a modified ResNet50 network with three separate convolutional layers as the object as a point with classification information, center offset, and the box size is determined. The Soft-NMS is used after prediction to improve detection accuracy.[22] An unsupervised learning strategy is established on the Self-Feature Comparison (SFC) to detect and segment fabric texture images accurately locate anomalies. The SFD and the Self-feature Reconstruction Module (SRM) modules are part of the SFC architecture. Faults were found using a collective anomaly score centered on feature reconstruction and distillation.[23]

Even if deep learning methods are more efficient than standard methods, the deployment of their systems frequently needs a significant amount of processing power. When computational resources are uncommon, the system's detection performance undergoes significantly. Most contemporary DL algorithms are paired with cloud computing to address this issue. However, data jams induced by the transmission of large amounts of data (such as images and videos) would significantly impact production efficiency in this cloud-centric approach.[24] The DL and edge computing are combined to solve the challenge. Open platform edge computing integrates data processing, storage, system, and application core activities while bringing computational and storage capabilities closer to customers or data sources.[25]

An edge computing approach has been proposed to detect fabric defects in the manufacturing industry with a minimum response, power, and easy upgradeability. The approach response time is 2.5 times greater than the cloud computing-based detection method.[26] Data improvement methodologies are proposed, along with the cross-entropy loss functions, to increase the model capacity for forecasting. To detect the defects in a real-time scenario, a modified CNN model with edge computing achieves good energy efficiency, a low response time, and high scalability.[27] The latency and power consumption issues were addressed better with the scalable and lightweight MobileNetV2 algorithm and NVIDIA Jetson TX2.

Materials and Methods

Data Set

The proposed method was evaluated on four fabric defect datasets, i.e., Aliyun-FD-10500[28], TILDA textile[29] (https://lmb.informatik.uni-freiburg.de/resources/datasets/tilda.en.html), DHU-FD-500, and DHU-FD-1000.[30] The Aliyun-FD-10500 is gathered from the publicly available fabric defect classification, i.e., TianChi Competition, TILDA, fashioned by a workspace on texture analysis of Deutsche Forschungsgemeinschaft Germany. The DHU fabric defect datasets are gathered from Donghua University (DHU), and each dataset has different classes of fabric defects.[28–30] DHU has 10 class labels like normal, broken pick, slub, double flat, sundries, broken-end, mis-pick, felter, oil-stains, and drawback with 500 and 1000 different images of size 224 × 224, as shown in Fig. 2.

The Tilda dataset has 3200 images of 768 × 512 pixels with eight class labels such as oil stains, broken end, holes, missing weft, slack end, ripped, cut selvage, kink, and normal. An instance of defects and texture base is depicted in Fig. 3.

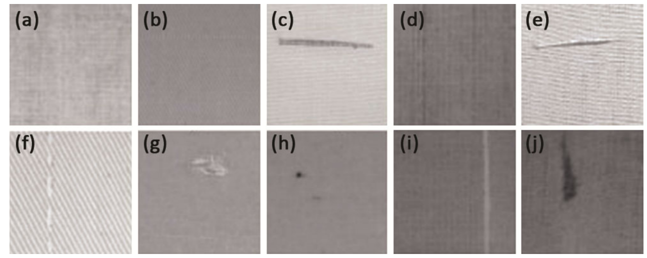

Fig. 2 — DHU-FD-500/1000 defective sample images: (a) Normal, (b) Mispick, (c) Broken picks, (d) Double flat, (e) Slub, (f) Felter, (g) Draw-back, (h) Sundries, (i) Broken end, (j) Oil stains

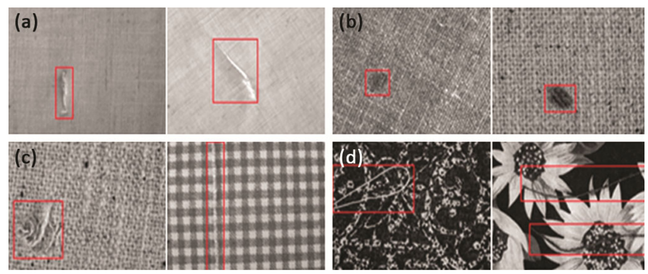

Fig. 3 — Defective images of the TILDA dataset (a) hole, (b) oil strain, (c) slackened, (d) dark wire

In Aliyun-FD-10500, the images are 10500 with seven class labels, i.e., stain, broken end, hole, felter, crack, broken picks, and normal 224 × 224 image size. The dataset description is shown in Table 1.

Proposed Method

With the features of the DL model and EC platform, an automatic FDD system meets the requirements of an industry. In this, we proposed the framework of an automated and intelligent model with edge computing for FDD, shown in Fig. 4 which achieves specific characteristics with more benefits. The computerized system has improved operational efficiency and defect detection with the real-time quick response production line. The detection system must have the scale feature for the manufacturing line to be easily upgraded for future efficiency. Also, the production line should have low power usage for lower production costs. We use the MobileNetV2 architecture and an edge computing detection system to respond to the specific requirements above. MobileNetV2 presents a highly successful mobile-oriented model that may be utilized on the cross-establishment for various visual recognition applications. MobileNetV2 offers a highly productive

Table 1 — Data-set Description

the first one is TILDA which is created by a workshop on texture analysis of Deutsche Forschungsgemeinschaft German

the first one is TILDA which is created by a workshop on texture analysis of Deutsche Forschungsgemeinschaft Germany TILDA which is created by a workshop on texture analysis of Deutsche Forschungsgemeinschaft Germany

Dataset Name	Samples Size	Number of Class Labels	Data Source
Aliyun-FD-10500	10500	7 (stain, broken end, hole, felter, crack, broken picks, and normal)	TianChi competition
TILDA textile	3200	8(oil stains, broken end, holes, missing weft, slack end, ripped, cut selvage, kink and normal)	TechnischeUniversitt Hamburg
DHU-FD-500	500	10 (normal, broken picks, slub, double flat, sundries, broken-end, mispick, felter, oil-stains and drawback)	Donghua University (DHU)
DHU-FD-1000	1000	10 (normal, broken picks, slub, double flat, sundries, broken-end, mispick, felter, oil-stains and drawback)	Donghua University (DHU)

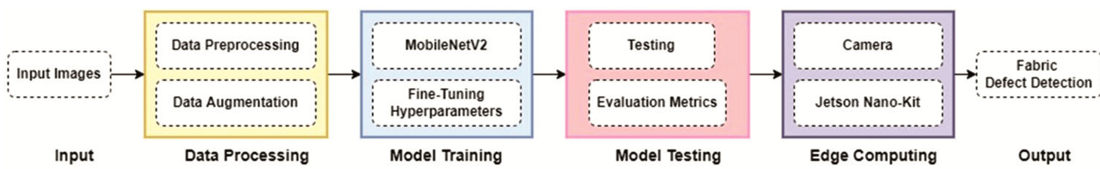

Fig. 4 — The framework of the proposed model

mobile-oriented model that can serve as the basis for various optical recognition applications.

The Edge device installation with the DL model can effectively reduce the production line power usage, system response time, and the industry's production investment. A lightweight network model, MobileNetV2, extracts the features with liner bottlenecks and inverted residuals for better detection speed. Similarly, the edge device computes the tasks from cloud to edge, improves the detection speed, and preserves data privacy in the manufacturing production.

Data Preprocessing and Augmentation

Poor lighting, excessive noise, and broad rotation angle commonly influence fabric data collected in complex industries. They can significantly affect the recognition accuracy of a model trained with the data. The dataset must be handled with caution to obtain a robust model. Appropriate preprocessing processes are required to improve prediction accuracy and reduce the total execution delay. As shown in Fig. 5, a range of feature selection methods (i.e., flipping, rotating, scaling, brightness conversion, contrast adjustment, and mosaic) can be coupled to maximize the number of training images. In addition, when the created images closely resemble their natural versions, these data augmentation techniques can increase the input image diversity, enhancing the trained model.

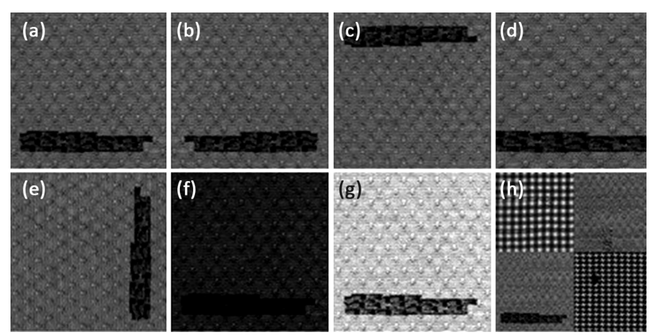

Fig. 5—Data Augmentation Techniques: (a) Original, (b) Horizontal Flip, (c) Vertical Flip, (d) Scale up, (e) Rotate, (f) Gamma Contrast, (g) Change brightness, (h) Mosaic

We enhanced the training datasets with brightness increase, blurring, flipping, random cropping, and noise to improve the model robustness in the real-world manufacturing environment. We tested the model on four fabric datasets for detection, where the flip transformation uses to increase the data size and refine the quality of the model. The approach random crop simulated the sample obtained from several camera views. Brightness increase is a lighting effect used to generate a range of different lighting effects, and finally, the model's durability improved by adding noise and blur.

MobileNetV2 Architecture

MobileNets are low-power, low-latency models that meet various resource constraints of different use cases. It significantly improves performance of the state-of-the-art model, a wide range of functions and performance metrics. It is also a powerful feature extractor for detecting and segmenting objects.

MobileNetV2 uses depth-wise separable convolution, resulting in a lower computational cost than standard convolutions with only a minor loss in accuracy. A bottleneck depth-separable convolution with residuals is the fundamental building block of MobileNetV2. It is optimized for resource-constrained applications such as edge devices, reducing the number of operations and memory requirements while maintaining accuracy. The architecture includes 32 filters of the first convolution layer, subsequent with the residual bottleneck layers of 19. The MobileNetV2 is designed using inverted residuals and linear bottleneck units.[31] For non-linearity, ReLU6 is used, and considering its robustness when used with 3 × 3 kernel size, low-precision computation, dropout, and batch normalization during training. It gets a low-dimensional compressed input for representation and enlarges to a high-dimensional before filtering it with a depth-wise convolution.

Recommender Systems, Natural language processing, video and image analysis, and other applications use deep learning. The DL models must meet strict throughput and latency limitations in safety-critical applications like automotive. Also, with more deep learning applications in production, the demand for performance and accuracy has led to huge model sizes and complexity. Initially, the model was trained on the workspace and shifted to the NVIDIA Jetson TX2 edge device. To optimize and accelerate the model, we use TensorRT to improve the detection speed and find the system delayed response time to ensure that the proposed approach is feasible. TensorRT is a DL deployment tool used in various circumstances. It supports all main frameworks and quickly analyses large amounts of data with advanced optimizations, reduced precision, and efficient memory utilization.

Transfer Learning

Transfer learning is a method for performing tasks like image classification, and natural language processing using feature representations from a model that has already been trained. Transfer learning is frequently used when the dataset is too large to train a model from end to end. The pre-trained models are continuously trained on massive datasets, a standard benchmark in computer vision. Numerous computer vision applications can use the weights that the model generates. These models can be used to directly forecast outcomes for novel tasks or as a component in training a new model. When pre-trained models are used in a new model, the training time and generalization error are reduced.

Fine-tuning is a stage in transfer learning used to improve the model's performance. When validating the model to obtain the final outputs during fine-tuning, the parameters of a trained model are fine-tuned and tailored to fit precisely. To retrain the model or a subset, use a low learning rate and avoid overfitting.

Results and Discussion

This section will conduct experiments to see how well the proposed method works. All tests were performed on a DELL Power Edge R740 Server equipped with an Intel Xeon Gold 6226R-2.9G processor, 128 GB of RAM, and an NVIDIA Quadro RTX8000 GPU with 48GB of GDDR6 memory. The Implementation was done using the Ubuntu operating system and the DL PyTorch framework. The Jetson Nano-kit, the most power-efficient embedded and fastest AI computer device, was used for all edge research.

Considering small datasets, we applied different image transformation techniques, such as data augmentation, to improve model generalization accuracy. The suggested method uses MobileNetV2 as the base model and applies transfer learning with fine-tuning hyperparameters. Each dataset is distributed into training and testing with a proportion of 90% and 10%, respectively. All four data-set images are resized into 224 × 224 and fed the input of the MobileNet. The image scale transformation is done randomly from 100% to 150%. Rotation with a random of 0 to 45 degrees, flip with 25% and shear with a random of 0 to 30°, shown in Table 2. The dataset is batch normalized with a batch size of 32 images. The Table 3 depicts the parameters used for experimentation to fine-tune the model, assessing

Table 2 — Image transformations used for data-set augmentation

Parameters	Value
Scale	Random between 100 and 150%
Rotation	Random between 0 and 45^0
Horizontal flip	Random 25%
Vertical flip	Random 25%
Shear	Random between 0 and 30°

Table 3 — Training and Testing phase parameters

Parameters	Value
Batch Size	32
Learning rate	0.001
Loss	Cross Entropy
Optimizer	SGD
Activation function	ReLU6
Epochs	50
Alpha	1.0

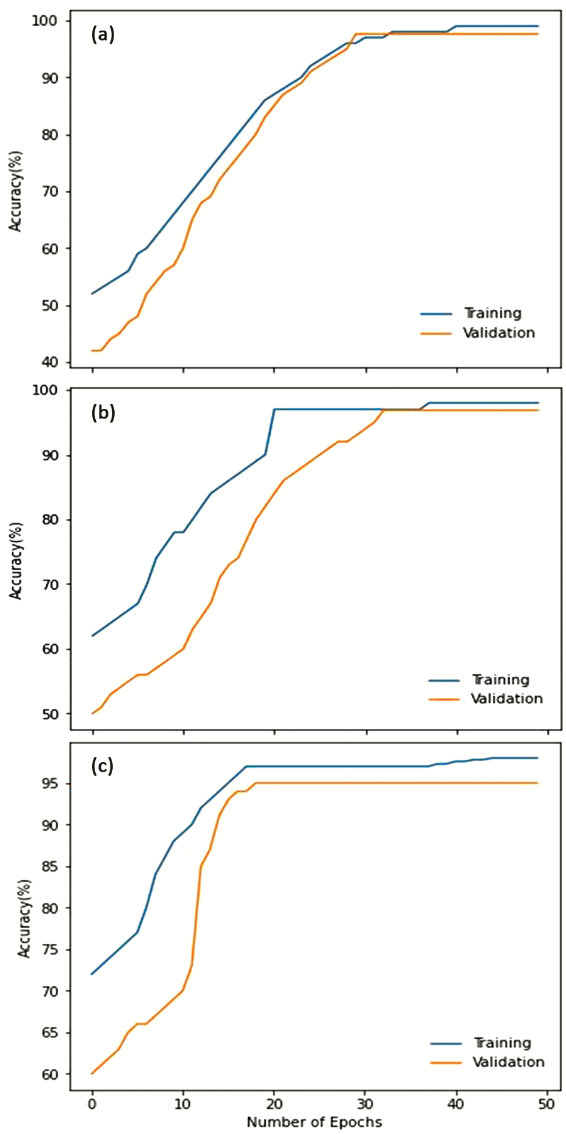

Fig. 6 — Data-set Model accuracy: (a) DHU-FD, (b) Aliyun-FD, (c) TILDA textile

different learning rates from 0.01 to 0.0001 and batch sizes from 16 to 64 and 25 to 100 epochs. After extensive experiments, the number of epochs is 50, 0.001 is the learning rate, and 32 batch size was determined.

Table 4 — Classification Report

Data-set	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Aliyun-FD-10500	96.87	96.87	97.04	96.87
TILDA textile	94.94	94.94	95.39	94.94
DHU-FD-500 and DHU-FD-1000	97.76	97.76	97.82	97.76
Average	96.52	96.52	96.75	96.52

Evaluation Metrics

Calculating Recall, Precision, Accuracy, and F1-Score based on the Confusion matrix is a generic objective technique to approve the algorithm performance for classification. The TP, and TN are used to describe truly defective and defect-less, respectively. Similarly, FP and FN provide description of false positive and false negative. The metric precision is computed as $\frac{TP}{FP+TP}$ while the corresponding Recall is calculated as $\frac{TP}{FN+TP}$. Further, the other significant metric is the accuracy which is given by

$$\frac{TP+TN}{TN+FP+TP+FN}$$

Similarly, the F1-Score is a harmonic mean between precision and recall and is given as

$$\frac{2\ (\text{precision} \times \text{reacall})}{\text{precision} + \text{recall}}$$

The first experiment uses DHU-FD-500 and DHU-FD-1000 data-sets, giving a test accuracy of 97.76%. The training accuracy reaches 98.21%, as shown in Fig. 6(a). The second experiment used the Aliyun-FD-10500 dataset, producing the test accuracy of the model at 96.87% and the training accuracy at 98.25%, as depicted in Fig. 6(b). The third experiment uses the TILDA textile data-set to generate 95.6% training accuracy and 94.94% testing accuracy, as shown in Fig. 6(c). Complete experiment results and the classification report are shown in Table 4.

Conclusions

With edge computing, this work provides a lightweight MobileNetV2-based Transfer Learning model to detect defect in fabrics with low power consumption, low latency, better efficiency, easy upgrading, and an automatic visual inspection system. The image transformation techniques were performed as data augmentation on four fabric datasets for the model's adaptability in various fabric materials. Transfer learning is used to fine-tune the MobileNetV2 hyperparameters resulting in a lightweight, versatile, and scalable model, making it ideal for resource-constrained edge devices. Finally, to make detection faster, the learned model to the NVIDIA Jetson Nano-kit edge device is deployed. To conclude, the trained network model is both quick and accurate when it comes to detection and improves fabric production efficiency in textile industries.

References

1 Najafabadi F S & Pourghassem H, Corner defect detection based on dot product in ceramic tile images, in *2011 IEEE 7th Int Colloq Signal Process Appl* (UniversitiTeknologiMARA) 2011, 293–297.

2 Elham H, Farhadi F & Tajeripour F, Fabric defect detection using auto-correlation function, *Int J Comput Theory*, **5(1)** (2013) 114.

3 Ghazvini M, Monadjemi S A, Movahhedinia N & Jamshidi K, Defect detection of tiles using 2D-wavelet transform and statistical features, *World Acad Sci Eng Technol*, **49** (2009) 901–904.

4 Janakiramaiah B, Kalyani G & Jayalakshmi A, Automatic alert generation in a surveillance system for smart city environment using deep learning algorithm, *Evol Intell*, **14(2)** (2021) 635–642.

5 Zhou X, Li Y & Liang W, CNN-RNN based intelligent recommendation for online medical pre-diagnosis support, *IEEE/ACM Trans Comput Biol Bioinform* **18(3)** (2020) 912–921.

6 Rasheed A, Zafar B, Rasheed A, Ali N, Sajid M, Dar S H, Habib U, Shehryar T & Mahmood M T, Fabric defect detection using computer vision techniques: a comprehensive review, *Math Probl Eng* (2020). https://doi.org/10.1155/2020/8189403

7 Srinivasan K, Dastoor P H, Radhakrishnaiah P & Jayaraman S, FDAS: a knowledge-based framework for analysis of defects in woven textile structures, *J Text Inst*, **83(3)** (1992) 431–448.

8 Chan C H & Pang G K, Fabric defect detection by Fourier analysis, *IEEE Trans Ind Appl*, **36**(5) (2000) 1267–1276.

9 Selvi S S T & Nasira G M, An effective automatic fabric defect detection system using digital image processing, *J Environ Nanotechnol*, **6(1)** (2017) 79–85.

10 Sadaghiyanfam S, Using gray-level-co-occurrence matrix and wavelet transform for textural fabric defect detection: A comparison study, in *2018 Elect Electrn Comput Sci Biomedi Eng Meeting* (EBBT) (IEEE) 2018, 1–5.

11 Anandan P & Sabeenian R S, Fabric defect detection using discrete curvelet transform, *Procedia Comput Sci*, **133** (2018) 1056–1065.

12 Bertolini M, Mezzogori D, Neroni M & Zammori F, Machine Learning for industrial applications: A comprehensive literature review, *Expert Syst Appl*, **175** (2021) 114820. https://doi.org/10.1016/j.eswa.2021.114820

13 Jolliffe I T & Cadima J, Principal component analysis: a review and recent developments, *Philos Trans Royal Soc A: Math Phy Eng Sci*, **374**(2065) (2016) 20150202.

14 Li M, Wan S, Deng Z & Wang Y, Fabric defect detection based on saliency histogram features, *Comput Intell*, **35(3)** (2019) 517–534.

15 Mei S, Wang Y & Wen G, Automatic fabric defect detection with a multi-scale convolutional denoising autoencoder network model, *Sensors* **18(4)** (2018)1064.

16 Zhang W, Jia M P, Zhu L & Yan X A, Comprehensive overview on computational intelligence techniques for machinery condition monitoring and fault diagnosis, *Chinese J Mech Eng*, **30(4)** (2017) 782–795.

17 Khan A, Sohail A, Zahoora U & Qureshi A S, A survey of the recent architectures of deep convolutional neural networks. *Artif Intell Rev* **53(8)** (2020) 5455–5516.

18 Alzubaidi L, Zhang J, Humaidi A J, Al-Dujaili A, Duan Y, Al-Shamma O & Farhan L, Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions, *J Big Data*, **8(1)** (2021) 1–74.

19 Beljadid A, Tannouche A & Balouki A, Application of deep learning for the detection of default in fabric texture, in *2020 IEEE 6th Int Conf Optim Appl*, (IEEE) 2020, 1–5.

20 Janakiramaiah B, Kalyani G, Karuna A, Prasad L V & Krishna M, Military object detection in defense using multi-level capsule networks, *Soft Comput* (2021) 1–5. https://doi.org/10.1007/s00500-021-05912-0

21 Janakiramaiah B, Kalyani G, Prasad L V, Karuna A & Krishna M, Intelligent system for leaf disease detection using capsule networks for horticulture, *J Intell Fuzzy Syst*, **41(6)** (2021) 6697–6713.

22 He Y, Huang X Y & Tay F E H, Fabric defect detection based on object as point, in *CS & IT Conf Proc* ONEDU, **11(6)** (2021) 99–107. DOI: 10.5121/csit.2021.110608

23 Peng Z, Gong X, Wei B, Xu X & Meng S, Automatic unsupervised fabric defect detection based on self-feature comparison, *Electronics*, **10(21)** (2021) 2652.

24 Li E, Zeng L, Zhou Z & Chen X, Edge AI: On-demand accelerating deep neural network inference via edge computing, *IEEE Trans Wirel Commun*, **19(1)** (2019) 447–457.

25 Shi W, Sun H, Cao J, Zhang Q & Liu W, Edge computing-an emerging computing model for the internet of everything era, *J Comput Res Dev*, **54(5)** (2017) 907–924,

26 Song S, Jing J, Huang Y & Shi M, EfficientDet for fabric defect detection based on edge computing, *J Eng Fibers Fabr*, **16** (2021) 15589250211008346.

27 Zhu Z, Han G, Jia G & Shu L, Modified densenet for automatic fabric defect detection with edge computing for minimizing latency, *IEEE Internet Things J*, **7(10)** (2020) 9623–9636.

28 Şeker A, Peker K A, Yüksek A G & Delibaş E, Fabric defect detection using deep learning, in *2016 24th Signal Process Commu Appl Conf*, (SIU) (IEEE) 2016, 1437–1440 D. F. Germany, "Tilda Textile Texture Database." [http://lmb.informatik.uni-freiburg.de/resources/datasets/tilda.en.html. Version 1.0, 1996 (access)

29 Zhao Y, Hao K, He H, Tang X, & Wei B, A visual long-short-term memory based integrated CNN model for fabric defect image classification, *Neurocomputing*, **380** (2020) 259–270.

30 Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L C, MobileNetV2: Inverted Residuals and Linear Bottlenecks, in *Proc IEEE Conf Comput Vis Pattern Recognit*, **18(22)** (2018) 4510–4520.