Journal of Scientific & Industrial Research
Vol. 82, April 2023, pp. 418-425
DOI: 10.56042/jsir.v82i04.72390

Lightweight CNN Models for Product Defect Detection with Edge Computing in Manufacturing Industries

Janakiramaiah Bonam[1]*, Sai Sudheer Kondapalli[2], Narasimha Prasad L V[3] & Krishna Marlapalli[4]

[1]Prasad V. Potluri Siddhartha Institute of Technology, Vijayawada, 520 007, Andhra Pradesh, India
[2]Tejas Networks, Bengaluru, 560 100, Karnataka, India
[3]Computer Science and Engineering, Institute of Aeronautical Engineering, Hyderabad, 500 043, Telangana, India
[4]Computer Science and Engineering, Sir C R Reddy College of Engineering, Eluru, 534 007, Andhra Pradesh, India

Received 07 July 2022; revised 22 September 2022; accepted 07 October 2022

Detecting product defects is one of the manufacturing industry's most essential processes in quality control. Human visual inspection for product defects is the traditional method employed in the industry. Nevertheless, it can be laborious, prone to human mistakes, and unreliable. Deep Learning-based Convolution Neural Networks (CNN) has been extensively used in fully automating product defect detection systems. However, real-time edge devices installed at the manufacturing site generally have limited computing capability and cannot run different CNN models. A lightweight CNN model is adopted in this scenario to find a balance between defect detection, model training time, memory consumption, computing time and efficiency. This work provides lightweight CNN models with transfer learning for product defect detection on fabric, surface, and casting datasets. We deployed the trained model to the NVIDIA Jetson Nano-kit edge device for detection speed with better simulation results in terms of accuracy, sensitivity rate, specificity rate, and F1 measure in the workplace context of the Manufacturing Industries.

Keywords: Convolution neural network, Deep learning, Edge devices, Lightweight model

Introduction

The quality of manufactured goods is readily impacted during the industrial production process because of the shortcomings and constraints of working conditions, current technology, and other factors. Traditional industrial techniques are being replaced by new strategies based on artificial intelligence to make decisions on their own, work independently, and also continuous learning. Delivering products with high quality is most important in manufacturing industries. The most prominent issue affecting the rate is the product's surface defects. So, product surface defect detection is required to guarantee the qualification ratio and reliable quality. As the Internet of Things has grown in popularity, artificial intelligence and computer vision have penetrated our daily lives.[1] Across various industries, efforts are being made to reduce labour-intensive and dangerous old procedures. With the advent of Industry 4.0, numerous businesses began to concentrate on automating their everyday tasks, deploying everything from test automation services to product development. Automated technology enhanced the production floor and assisted employees in avoiding monotonous or laborious activities.

Edge computing is a distributed computing framework that relocates services, memory, and processing power to the location where data are generated. Deep Learning (DL) has recently been very successful in machine learning across many application domains, such as computer vision, medical diagnosis, agriculture/horticulture, natural language processing, and big data analysis.[2–4] The DL depends on high-performance computing platforms with much storage space for the data required to train such models. Deep learning often uses cloud computing to handle the large amount of computing it needs. When cloud resources are used, data must be moved from its source on the network edge, like smart phones and Internet of Things (IoT) sensors, to a central location in the cloud. Moving the data from the source to the cloud could be a solution, but it has some problems, like latency, security, and the ability to grow.[5] Machine-to-machine communication protocols allow smart edge devices worldwide to communicate with each other and share information all the time. Through machine-to-machine

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

communication protocols, smart edge devices throughout the globe communicate and exchange data continuously. Massive volumes of data are generated by a wide variety of similar sensors and endpoints; these data can be handled in real-time and utilized to construct deep convoluted learning models. End devices are considered edges in this context, and these are any gadgets, wearables or mobile devices that collect data closest to the user. A few of the layers of processing and storage that could be present at the edge are the device, client premises, and edge servers.

Deep learning can enable edge devices to integrate and interpret unstructured types of information (audio, visual, and text) and take corrective measures, even though it comes at the expense of greater power and performance metrics. Because deep learning resources require a lot of computation time and resources to process the information streams generated by these sensors and edge devices, such information can scale quite rapidly. Hence it is vitally important to run the algorithms at the edge. Using applications and services that operate on the edge already benefits many applications. These applications can be much enhanced using DL techniques.

Compared to the growing work on deep learning, lightweight models for resource-constrained devices using edge computing have more benefits related to shared communication and computational resources. The main contributions of this work are as follows:

1) Exploring the different techniques for product defect detection in the manufacturing industry.
2) To investigate various lightweight models for product fault detection that can be used on devices with limited resources.
3) To improve system response time, conserve energy by sending fewer data to the cloud, and enhance network robustness, we are investigating the history and features of edge computing edge devices. The architecture of lightweight CNN models with edge devices for manufacturing industrial defect detection applications is shown in Fig. 1.

The proposed framework starts with considering the dataset for defect detection in the manufacturing products of industries. The trained model's accuracy depends on the dataset images' quality. Hence, data pre-processing is applied to the input images. After the pre-processing, the dataset is divided into training and testing data. The training data is input to the Lightweight CNN model with fewer parameters, and the trained CNN model is evaluated with a test dataset. If the model achieves good accuracy, then the weights and biases are considered for model conversion and optimization for deploying on the edge devices. The edge devices can detect and classify the defects in the manufactured products in the industries.

Product Defect Detection Systems in Manufacturing Industries

Quality control is a critical manufacturing element; companies use the best technologies to discover problems. It is essential in many manufacturing processes, notably mass production. Scratches, abrasions, pits, and internal holes are defects during manufacturing products in complex industrial processes because of defects in the machine production equipment, design and adverse working conditions.[6–7] Due to continuous use, products may rust and wear out quickly. In addition to raising prices for enterprises, shortening the usable lives of produced goods, and wasting a considerable amount of resources, these defects seriously affect individuals and their safety. Manual fault diagnosis is excelled by

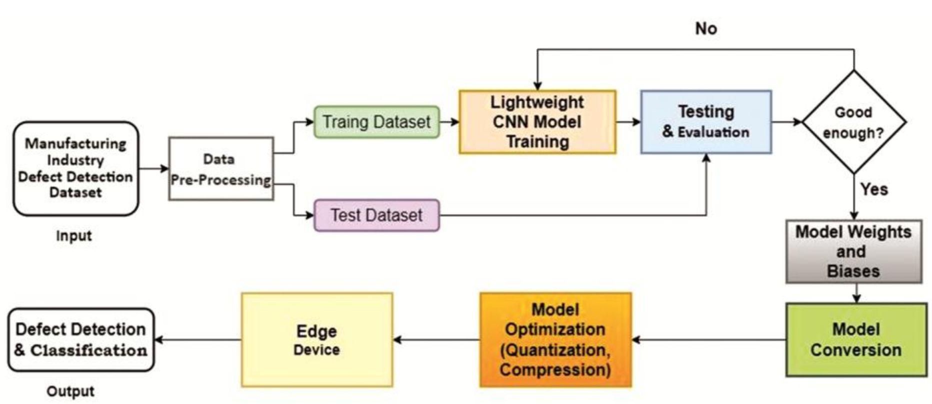

Fig. 1 —Architecture for lightweight CNN Model with edge device

automatic fault detection technology. In addition to adapting to an inappropriate environment, it functions long-term with great accuracy and efficiency. Therefore, determining faults is a vital ability industry should have to raise the quality of manufactured goods without compromising output. Defect-detection technology research can lower production costs, boost productivity and product quality, and create a solid framework for the systematic transformation of the industrial sector.

The use of technology for defect detection is a popular topic in both academics and the industry. Researchers have yet to categorize the primary detection methods, a summary of technology for the applications of defect detection, the prevailing equipment, and other forecasts. Product defect types include surface, casting, textile, and natural defects, which can be detected with detection techniques.

Surface Defect Detection System

Surface defects degrade the performance of industrial products. Manufacturers have invested significant effort in examining product quality and looking for surface defects. Machine vision-based techniques for surface defect detection have gained prominence in recent years due to their capacity to overcome various limitations of physical detection, poor real-time performance, low accuracy, and high labour intensity. To a certain extent, the primary research discusses surface defect detection systems.

The defect detection technique for ceramic tile surfaces can be achieved using deep learning. It is demonstrated in the literature that a Mask Region-based Convolutional Neural Network (Mask R-CNN) can be used to construct a pixel-level instance segmentation approach.[8] A dynamic bottle inspection system has been modelled using an Artificial Neural Network (ANN) trained by the differential evaluation (DEA), Back Propagation (BP), and Support Vector Machine (SVM) methods to categorize the defect images.[9] Similarly, a classifier employing the characteristics of image patches extracted from a trained Deep Neural Network (DNN) is also considered in the existing work available in the literature.[10] Further, a high-resolution Automated Optical Inspection (AOI) system is suggested for parallel computing.[11]

Casting Defect Detection System

Casting defects damaging the end product's quality are unavoidable during the process.[12] If defects were not detectable, critical mechanical components would fail, so to avoid errors, each component must be thoroughly inspected. Early defect detection aids in the early discovery of defective items throughout the production process thus saving time and money. As a result, many academics have investigated defect detection-related applications and technologies to provide references for defect detection technology applications and studies.

Using the Maximum Between-Class Variance, the authors built a defect identification method based on original image thresholding.[13] In, a system for automatic defect detection in aluminium castings was proposed based on a two-step analysis: radioscopic image recognition and tracking. In work, the Phase-Only Fourier Transform (POFT) is used to detect saliency, significantly improving vulnerable regions.[14] In the work, the authors employed the anchor box initialize clustering technique to construct an enhanced You Only Look Once (YOLOv3) algorithm.[15] An in-depth convolution neural network developed, centre-peripheral difference calculation approach based on selective attention mechanism.[16] Many cutting-edge object analyzers are utilized to localize casting defects when the feature extraction layer is detached from the object identification architecture.

Fabric Defect Detection System

Fabric fault diagnosis has long been a focus of computer science and technology study. Fabric defects come in various forms, produced mainly by process issues and machine failures. Defects will reduce the end product's quality, resulting in a significant waste of all types of resources.[17] This section describes and categorizes fabric defect-detecting technologies in a broader context.

Fabric Defects Analysis System (FDAS) has been proposed as a novel method for defect categorization in woven textiles based on visually detectable faults that do not require prior information.[18] The approach central spatial frequency spectrum is presented to improve the efficiency of the analysis process and detect structural defects in fabrics.[19] A successful automated fabric inspection system for multi-class defect detection and fabric categorization is proposed using geometrics and texture information to capture visual features. Fabric analysis is done using digital images. With the image capture device, the recognition system acquires digital fabric images and transfers them for processing to determine if the fabric is defect-free, or defective.[20] A unique method for finding defects in fabric images incorporates the Curvelet Transforms (CT) and the Gray-Level Co-occurrence Matrix (GLCM).[21] These methods give a reliable descriptive basis for fault textures derived from various images.

DST-PCA is a novel feature extraction method for detecting faults in knitting fabrics.[22] The features are obtained using the Discrete Shearlet Transform (DST), 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 patterned and non-patterned images. Histogram features are derived using Context-Aware (CA) saliency maps, which are subsequently fed into an SVM for classification.[23] The Multi-Scale Convolutional Denoising Auto-Encoder (MSCDAE) is a technique for fabric defect detection centred on networks. The method reveals problematic regions using the residual reconstruction maps created by the CDAE networks, increasing the model's robustness.[24] A method for detecting defects of fabric is based on the centrenet, a neural network with high accuracy and detection speed.[25] The SFC architecture includes the Self-Feature Distillation (SFD) and Self-feature Reconstruction Module (SRM).[26] A modified Resnet-50 network is used to extract the feature map, and three independent convolutional layers are used to determine the item as a point with categorization information, centre offset, and box size.

An overview of product defect detection systems in manufacturing industries is described above with various features, reduced human intervention, and optimized quality. These systems are not limited to deep learning features. There is a lot of scope in using deep neural networks with edge computing which attains more accuracy with low latency and high efficiency.

Lightweight Network Models

Deep learning is an enthralling subfield of machine learning. Massive amounts of data are used to train machines to perform tasks previously performed by humans. Some of the most intriguing and complex tasks involve figuring out how perception works, identifying what's in an image, and helping self-driving automobiles explore and interact with their environment. It represents cutting-edge of computer vision and speech recognition technology. Its computational complexity has reduced its adoption in various fields, including computer vision and language processing.

The DNN design space is vast and empirically investigated. It is highly beneficial in computer vision and image classification. In recent years, networks with a specific number of layers, kernels, and activation functions, among other factors, have been developed, focusing on CNN's. Deep CNN architectures are exceptionally computationally demanding. As a result, lightweight CNN architectures for edge computing are required. The lightweight CNN models are used to improve the intelligence of small devices with limited hardware resources.

The deep neural networks are all focused on accuracy. Less accurate networks are regarded as the worst and are therefore excluded from further examination. However, when measures such as cost, performance, energy consumption, and hardware size are considered, the precise networks are not always improved. Allowing an uncertain drop in accuracy can occasionally result in a significant reduction in the number of parameters and procedures in a model. The effort required to run extensive networks on mobile and end devices is substantial, and in these circumstances, for network accuracy, the direction has been trade-off the device resource deployment. As a result, numerous network models geared toward mobile and end devices have been developed. This section describes some of these models.

MobileNet

Constructing a lightweight deep neural network divides the convolution into depth-wise and point-wise. Furthermore, it offers two simple hyper parameters, namely the width multiplier and the resolution multiplier, allowing to development of tiny models with zero latency that match the design requirements of mobile end devices. MobileNet excels in terms of resource/accuracy trade-off. It only provides excellent accuracy with limited resources.

The lightning-fast MobileNet architecture is ideal for real-time applications. Compared to MobileNetV1, it is more efficient in terms of accuracy and speed. MobileNet was later improved to MobileNetV2, which included shortcut connections and in-between data encoding to minimize the sum of operations and weights.[27] The bottleneck residual block has replaced the primary building block.

SqueezeNet

SqueezeNet is a network that has comparable accuracy to AlexNet but with 50 fewer parameters.[28] More than a few policies were employed to accomplish the significant reduction in parameters:

(1) moderate the number of inputs of 3×3 filters; (2) replace 3×3 filters with 1×1 filters; and (3) down sample the network with pooling layers. SqueezeNet begins with a single convolution layer (conv1), followed by eight fire modules (fire2 to fire9), and then a final convolution layer (conv10). The Fire module is the SqueezeNet design's base, and it consists of two layers: a squeeze convolution layer with only (1×1) filters and an expanded layer with a combination of (1×1) and (3×3) convolution filters.

ShuffleNet

ShuffleNet is a lightweight CNN model with minimal computational resources for mobile devices. It comprises three levels of ShuffleNet units stacked on top of each other.[29] The ImageNet classification tasks are performed better than MobileNet. In order to reduce the computation costs while retaining accuracy, the ShuffleNet design employs two novel processes. Stride 2 is used as the first building piece in each stage. Each stage's output channels are the same, while the next ones are doubled. Point-wise group convolution and channel shuffle are two of these operations. The shuffle operation channel enables the partition of each group's channels into several subgroups, which are subsequently fed to each group in the following layer.

CondenseNet

CondenseNet uses grouped convolutions; before training, the output maps are separated into G groups of the same size.[30] During training, it learns which to group when the channels are randomly shuffled. Pruning is also integrated with another hyperparameter, C. 1/C parameters are pruned at the close of every condensing stage. Following the condensing stage, an additional optimization stage is conducted to remove less essential feature maps. However, it attains similar accuracy to ShuffleNet, with around half the parameters.

PeleeNet

PelleNet is efficient embedded platform architecture comprising four rounds of feature extraction and a stem block; each stage has a stride two average pooling layer, except for the last layer.[31] The PeleeNet model is 66% more compact than the MobileNet model. It is 1.8x faster and achieves higher precision on ImageNet ILSVRC 2012 using NVIDIA Jetson T×2 than the two MobileNet variants.

NASNet

NASNet begins with an overall architecture and a set of non-predefined blocks rather than training a predetermined network.[32] The two types of blocks in the architecture are normal (maintains the feature maps size) and reduction (lowers the feature maps size by 4). A reinforcement learning search strategy is used to control the structure of the blocks.

To restrict the processes and parameters, every network experimented with several main module topologies. The lightweight CNN models described above are all categorized according to model complexity. The lightweight network models are approximately ten times faster and contain fewer parameters than Deep Neural Network models but are less accurate.

Edge Computing

The practice of physically bringing compute power closer to the source of data, which is typically an IoT device or sensor, is known as Edge Computing (EC). It is named after the method of delivering computing power to a device's or network's "edge". By processing data at the network's edge, edge computing decreases the need for big data to travel among servers, the cloud, and devices or edge locations. The intelligence at the edge domain can be achieved by model partitioning, simplification, and the IT industry's latest hardware and software solutions. Therefore, EC and Intelligent manufacturing with information-enabled operations provide vast opportunities to enhance business performance. Edge computing has the following benefits: Lower network latency, increased compute, storage and network capacity, network bandwidth expansion, an increase in overall system response time, Node-aware security and privacy, At-node fault tolerance and mitigation, and reduced the amount of data sent to the cloud to save energy.

Integrating computer vision and AI into the IoT and edge device prototypes is now more accessible to the improved capabilities of the Intel Neural Compute Stick. The NVIDIA Jetson Nano Developer Kit is a compact but powerful computer that simultaneously operates multiple neural networks for speech processing, image classification, segmentation, and object detection. It explores the major supporting hardware for edge intelligence, including specialized AI processors and standard parts for edge nodes. OpenMV aspires to be the "Arduino of Machine Vision" by creating Python-powered machine vision modules, low-cost and scalable. Machine vision algorithms are simple to run on the OpenMV Cam; to track, and detect faces, colours, and more in seconds and then control I/O pins in real time. The Thundercomm TurboX AI Kit is a high-performance embedded development device based on Qualcomm SDA845 processors. The TurboX AI Kit is designed to aid in developing on-device AI applications in robotics, augmented reality/virtual reality, intelligent cameras, automotive, smart retail, smart factory, smart home, and smart city.

Experiments and Discussion

The proposed framework runs on a DELL Power Edge R740 Server with an Intel Xeon Gold 6226R-2.9G processor, 128 GB of RAM, and an NVIDIA Quadro RTX8000 GPU - 48 GB of GDDR6 memory. The deep learning framework PyTorch and the Ubuntu operating system were used in the implementation. All edge research was conducted using the Jetson Nano-kit.

Dataset

The fabric defect dataset, the surface of ball screw drives dataset, and the submersible pump impeller dataset were used to assess the lightweight CNN Models. The aliyun-FD-10500 fabric defect detection dataset contains 10500 images divided into seven classes.[33] The surface of the Ball Screw Drives Dataset[34] can contain 21835 images classified as "P, N," where P denotes surface failures known as pitting (also known as pitting (s)) and N denotes no pitting (no surface failures). The total number of images in the submersible pump impeller casting defect detection dataset[35] is 7340 images with two classes. The datasets summary report is displayed in Table 1.

Data Augmentation

On three datasets, data augmentation was done to improve the model's robustness. Random transformations are applied to the image scale, ranging from 100 to 150 percent, shear from 0 to 30 degrees, flip from 0 to 30 degrees, and rotation from 0 to 45 degrees, as shown in Table 2.

Fine-Tune the Hyper Parameters

The parameters utilized for testing to fine-tune the model are shown in Table 3; they include batch sizes ranging from 16 to 64 and 25 to 100 epochs, as well as learning rates ranging from 0.01 to 0.0001. After considerable testing, the batch size, learning rate, and a number of epochs were found to be 0.001, 50, and 32, respectively.

Result Analysis

We examined the pre-trained model, applied the fine-tuning techniques, and performed the experiments on three datasets. The Aliyun-FD-10500 dataset used MobileNetV2; the model's test accuracy and training accuracy achieved are 96.87% and 98.25%. The surface of the Ball Screw Drives dataset with the CondenseNetV2 model achieves an accuracy of 98.08%, and the Casting Defect Detection dataset with the ShuffleNet V2 model achieves an accuracy of 99.58% classification results are shown in Table 4.

Table 1 — Datasets summary report

Dataset name	Dataset size	Number of classes	Application
Aliyun-FD-10500	10500	7 (stain, broken end, hole, felter, crack, broken picks, and normal)	Fabric Defect Detection
Surface of the Ball Screw Drives	21835	2(Surface failure, No Surface failure)	Surface Defect Detection
Casting Defect Detection	7340	2(OK, Defective)	Casting Defect Detection

Table 2 — Image transformations used for dataset augmentation

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

Table 3 — Training and testing phase parameters

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

Table 4 — Classification report

Dataset	Lightweight CNN Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Aliyun-FD-10500	MobileNetV2	96.87	96.87	97.04	96.87
Surface of the Ball Screw Drives	ShuffleNetV2	99	100	99	99.58
Casting Defect Detection	CondenseNetV2	98.08	98.5	98.5	98.5

Conclusions

Defect detection is one of the most crucial parts of the manufacturing industry to ascertain product quality while delivering to the customers. I4.0 concentrates on automated processes or devices for defect detection in the quality-checking phase of the products. The experimented lightweight CNN models for product defect detections with edge devices achieve highly accurate results in detecting the defects. Edge computing is one technology used to incorporate the models with intelligence on the hardware devices. Most edge devices will include machine intelligence since deep learning has made substantial advancements recently, enabling lightweight CNN models to run exact algorithms on hardware with constrained resources. Relating deep learning model services to the edge has allowed numerous product defect detection in industrial applications, including surface, casting, and fabric, to execute with real-time constraints. The developed model of this research work is tested with three different datasets for each type of surface, casting and fabric defect. The work can be extended by considering multiple types of defects in each category of the products in manufacturing industries.

References

1 Routray S K, Sharmila K P, Javali A, Ghosh A D & Sarangi S, An outlook of narrowband IoT for industry 4.0, *In 2020 2nd Int ConfInv Res Comput Appl* (IEEE) 2020, 923–926, **DOI:** 10.1109/ICIRCA48905.2020.9182803

2 Kalyani G, Janakiramaiah B, Prasad L V, Karuna A & Babu A M, Efficient crowd counting model using feature pyramid network and ResNeXt, *Soft Comput*, **25(15) (**2021) 10497–10507, https://doi.org/10.1007/s00500-021-05993-x

3 Subramanian M, Lv N P & VE S, Hyperparameter optimization for transfer learning of VGG16 for disease identification in corn leaves using Bayesian optimization, *Big Data*, **10(3)** (2022) 215–229, DOI: 10.1089/big.2021.0218.

4 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, DOI: 10.3233/JIFS-210593

5 Grigorescu S, Trasnea B, Cocias T & Macesanu G, A survey of deep learning techniques for autonomous driving, *J Field Robotics*, **37(3)** (2020) 362–386.

6 Konovalenko I, Maruschak P, Brevus V & Prentkovskis O, Recognition of scratches and abrasions on metal surfaces using a classifier based on a convolutional neural network, *Metals*, **11(4)** (2021) 549, https://doi.org/10.3390/met11040549

7 Yu J, Zhang D, Li H, Song C, Zhou X, Shen S, ... & Wang H, Detection of internal holes in additive manufactured Ti-6Al-4V part using laser ultrasonic testing, *Appl Sci*, **10(1)** (2020) 365.

8 Pan Z, Yang J, Wang X E, Wang F, Azim I & Wang C, Image-based surface scratch detection on architectural glass panels using deep learning approach, *Constr Build Mater*, **282** (2021) 122717, https://doi.org/10.1016/j.conbuildmat.2021.122717

9 Sahoo S K, Sharma M M & Choudhury B B, A dynamic bottle inspection structure, In *Comput Intell Data Mining* (Springer) 2019, 873–884.

10 Ren R, Hung T & Tan K C, A generic deep-learning-based approach for automated surface inspection, *IEEE Trans Cybernetics*, **48(3) (**2017) 929–940, doi: 10.1109/TCYB.2017.2668395.

11 Jiang J, Cao P, Lu Z, Lou W & Yang Y, Surface defect detection for mobile phone back glass based on symmetric convolutional neural network deep learning, *Appl Sci*, **10(10)** (2020) 3621.

12 Hu C & Wang Y, An efficient convolutional neural network model based on object-level attention mechanism for casting defect detection on radiography images, *IEEE Trans Ind Electron*, **67(12)** (2020) 10922–10930.

13 Ng H F, Automatic thresholding for defect detection, *Patt Recogn Lett*, **27(14)** (2006) 1644–1649.

14 Bai X, Fang Y, Lin W, Wang L & Ju B F, Saliency-based defect detection in industrial images by using phase spectrum, *IEEE TransInd Inf*, **10(4)** (2014) 2135–2145.

15 Duan L, Yang K & Ruan L, Research on automatic recognition of casting defects based on deep learning, *IEEE Access*, **9** (2020) 12209–12216.

16 Lin J, Yao Y, Ma L & Wang Y, Detection of a casting defect tracked by deep convolution neural network, *Int J Adv Manuf Technol*, **97(1)** (2018) 573–581.

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

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

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

20 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.

21 Sadaghiyanfam S, Using gray-level-co-occurrence matrix and wavelet transform for textural fabric defect detection: A comparison study, In *2018 Electric Electron Comput Sci, BiomedEng' Meet* (IEEE) 2018, 1–5.

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

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

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

25 He Y, Huang X Y & Tay F E H, Fabric defect detection based on object as point, In *CS IT Conf Proc*, **6(11)** (2021).

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

27 Sandler M, Howard A, Zhu M, Zhmoginov A & Chen L C, Mobilenetv2: Inverted residuals and linear bottlenecks, In *Proc IEEE Conf Comput Vis Patt Recogn* (IEEE) 2018, 4510–4520.

28 Iandola F N, Han S, Moskewicz M W, Ashraf K, Dally W J & Keutzer K, SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size. *arXiv preprint arXiv:1602.07360* (2016).

29 Zhang X, Zhou X, Lin M. Sun J, ShuffleNet: An extremely efficient convolutional neural network for mobile devices, in *Proc 2018 IEEE/CVF Conf Comput Vis Patt Recogn* (IEEE) 2018, 6848–6856.

30 Huang G, Liu S, van der Maaten L & Weinberger K Q, CondenseNet: An efficient densenet using learned group convolutions, in *Proc 2018 IEEE/CVF Conf Comput Vis Patt Recogn* (IEEE) 2018, 2752–2761.

31 Wang R J, Li X & Ling C X, Pelee: A real-time object detection system on mobile devices, *Adv in Neural Infoproc Syst*, **31** (2018).

32 Qin X & Wang Z, Nasnet: A neuron attention stage-by-stage net for single image deraining, *arXiv preprint arXiv: 1912.03151* (2019).

33 Seker A, Peker K A, Yüksek A G & Delibas E, Fabric defect detection using deep learning, In *Proc the 24th Sig Proc Commut Appl Conf* (IEEE) 2016, 1437–1440.

34 Schlagenhauf T, *Ball Screw Drive Surface Defect Dataset for Classification* (2021)

35 Dabhi R, Casting product image data for quality inspection, (Kaggle.com) 2020.