Check for updates

Textile Research Journal

Original article

Textile Research Journal
0(00) 1–13
© The Author(s) 2020
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/0040517520928604
journals.sagepub.com/home/trj
SAGE

Mobile-Unet: An efficient convolutional neural network for fabric defect detection

Junfeng Jing[1], Zhen Wang[1], Matthias Rätsch[1,2] and Huanhuan Zhang[1]

Abstract

Deep learning–based fabric defect detection methods have been widely investigated to improve production efficiency and product quality. Although deep learning–based methods have proved to be powerful tools for classification and segmentation, some key issues remain to be addressed when applied to real applications. Firstly, the actual fabric production conditions of factories necessitate higher real-time performance of methods. Moreover, fabric defects as abnormal samples are very rare compared with normal samples, which results in data imbalance. It makes model training based on deep learning challenging. To solve these problems, an extremely efficient convolutional neural network, Mobile-Unet, is proposed to achieve the end-to-end defect segmentation. The median frequency balancing loss function is used to overcome the challenge of sample imbalance. Additionally, Mobile-Unet introduces depth-wise separable convolution, which dramatically reduces the complexity cost and model size of the network. It comprises two parts: encoder and decoder. The MobileNetV2 feature extractor is used as the encoder, and then five deconvolution layers are added as the decoder. Finally, the softmax layer is used to generate the segmentation mask. The performance of the proposed model has been evaluated by public fabric datasets and self-built fabric datasets. In comparison with other methods, the experimental results demonstrate that segmentation accuracy and detection speed in the proposed method achieve state-of-the-art performance.

Keywords

fabric defect, deep learning, Mobile-Unet, efficient convolutional neural network

Fabric is fundamental for many consumable products in daily life, such as bedsheets, quilt covers, and clothes, and it is also widely utilized in industry, such as in air suits, filter cloth, and so on.[1] The quality of products is critical, as it represents the reputation of enterprises. If the products produced by an enterprise are defective, it will cause not only a waste of raw materials, but also a substantial loss of customers. In addition, for some medical and space fabrics, the quality of fabrics may be related to life safety and spacecraft stabilizers. In order to ensure the quality of textiles, fabric inspection constitutes an essential process.

Currently, most fabric defect inspections are conducted visually by human workers, which usually leads to high labor costs and demand for skilled labor.[2] However, human errors are not uncommon and can severely affect production quality. According to feedback from textile factories, the accuracy of manual detection is only approximately 70%. In recent years, fabric defect detection based on computer vision has become a popular research topic. Automated visual inspection (AVI) of fabric applies computer vision techniques that not only improve efficiency and save costs, but also ensure the stability of product quality. In addition, with the development of Industry 4.0 and the information revolution occurring in factories, AVI systems can guide the improvement of the production process and control production costs.

[1]College of Electronics and Information, Xi'an Polytechnic University, China
[2]College of Engineering, Reutlingen University, Germany

Corresponding author:
Junfeng Jing, Xi'an Polytechnic University, No. 19 Jinhua South Road, Xi'an, Shaanxi 710048, China.
Email: jingjunfeng0718@sina.com

A variety of machine vision techniques have been studied and utilized for fabric defect detection. These defect recognition methods are mainly divided into four types: model-based methods, filter-based methods, learning-based methods, and feature-based methods.[3] For example, Jing et al.[4] used a Gabor filter to extract texture features, and employed golden image subtraction to segment the defects. Satisfactory results have been achieved in plain fabric, but pattern fabric defects cannot be accurately segmented. Due to the repeated texture structure of fabric, Li et al.[5] proposed a fabric defect detection method based on the low-rank representation method. However, due to the use of eigen-value decomposition on block image matrix for dimensionality reduction, the time complexity of the method is significant. A novel fabric defect detection method, which is based on the saliency metric for color dissimilarity and positional aggregation, was also proposed.[6] The algorithm has a desirable effect on the detection of patterned fabric, but due to the large amount of similarity calculation involved, the time complexity needs to be further improved. Zhang et al.[7] applied L0 gradient minimization to remove the background texture of fabric, and then used C-means clustering to segment the defects. Although this approach can detect fabric defects with diverse textures, numerous parameters of the method need to be adjusted manually. Kang et al.[8] presented a fabric defect segmentation method by basic image and the Elo-rating algorithm. This method can detect fabrics with small period texture well, but it cannot segment fabrics with large period texture accurately. Guan et al.[9] proposed a fabric defect detection method based on visual saliency, but it cannot be applied to red-green-blue color fabric defect detection. Typically, in the above methods, features must be hand-crafted to suit the particular task.

In recent years, convolutional neural networks (CNNs) have been widely utilized in image classification, object detection, and image segmentation.[10] It has been proven that, compared with traditional image processing methods, CNN can automatically learn useful features from data without requiring complex hand-designed features.[11] With the success of deep learning methods in various application areas,[12] these techniques are being applied to fabric defect detection tasks as well. As shown in Figure 1, the task of defect detection is divided into four levels, from coarse to fine: defect classification, defect location, defect segmentation, and defect semantic segmentation. Defect classification aims to identify the categories of defects contained in a fabric image. Defect location is used to detect the location and type of defects in a fabric image. Defect segmentation aims to determine whether each pixel in a fabric image is defective. Defect semantic segmentation is used not only to distinguish whether each pixel has defects, but also to identify the category of defects.

For defect classification, Li et al.[13] developed a compact CNN architecture for the detection of a few common fabric defects. The proposed network achieved superior performance in terms of detection accuracy with a much smaller model size, compared with mainstream CNN architectures. Similarly, because AlexNet possesses excellent feature extraction ability, a modified AlexNet[14] was proposed to classify the defects of yarn-dyed fabric. Considering that CNN cannot handle the small sample size problem very well in the classification task, Wei et al.[15] combined CNN and compressed sensing for defect classification. To augment the detection speed, a defect location algorithm based on the YOLO model[16] was proposed. Experiments demonstrated that the network achieves high detection accuracy for five kinds of fabric defects. Li et al.[17] proposed a defect location network based on surface defect detection, and provided a novel idea for defect location in actual industrial scenarios. The models proposed in the above research focus on the task of defect classification and location.

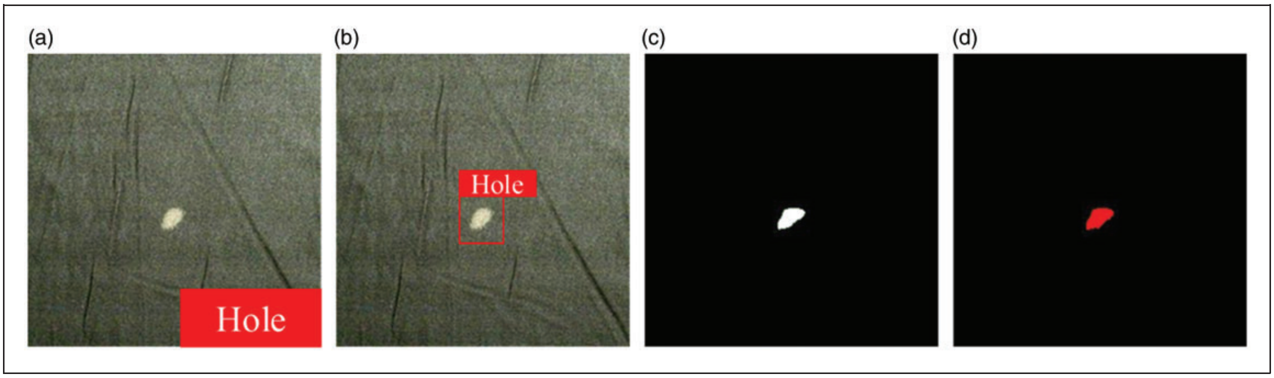

Figure 1. Four levels of defect detection: (a) defect classification; (b) defect location; (c) defect segmentation; (d) defect semantic segmentation.

The output of these models is the location of the defect class label or the defect relative to the image.

However, compared with defect classification and location detection models, defect recognition at the pixel level is often more useful and important in attempting to provide defect contours and separate them from background textures. Li et al.[18] proposed a pixel-level damage detection model for concrete structures based on the fully CNN (FCN). Miao et al.[19] presented an improved U-net model for defect segmentation, which integrates Resnet block and SE block to enhance its ability to segment small defects. Concerning fabric inspections, Tao et al.[20] designed a cascaded autoencoder for segmentation and classification of defects. However, because the detection requires two steps, the detection time cannot meet real-time requirements.

Hu et al.[21] proposed a fabric defect segmentation model based on the generative adversarial network, which has high defect segmentation accuracy for twill and plain fabrics, but cannot determine the defect type. Mei et al.[22] proposed a multi-scale convolution denoising network for fabric segmentation, but the performance of irregularly printed fabric detection needs to be further improved. Li et al.[23] proposed a method of pattern fabric defect detection. Even if the negative sample is not enough, it can obtain satisfactory detection accuracy. However, it is difficult to integrate this method into the automatic defect detection system because it cannot be detected in real time.

Many deep learning models, such as FCN[24] and SegNet,[25] have been proposed for image segmentation. In these models, U-net[26] has attracted wide attention due to its excellent segmentation performance. U-net was initially developed for biomedical image segmentation. The architecture is widely used in image segmentation for two reasons: it performs well on a small dataset and is trained end-to-end.[27] Despite its high accuracy, the detection approach consumes too much time. To balance real-time application and better feature extraction, we replaced the encoding part of U-net with MobileNetV2,[28] which is a classification network proposed by Google for mobile devices and real-time visual applications. This network can reduce the number of parameters without a loss of accuracy. Previous studies have shown that MobileNetV2 can achieve the same accuracy as VGG16 in ImageNet-1000 classification tasks, but the parameters are only 3% of VGG16. We named this hybrid model, Mobile-Unet, because it achieves the best of both models by using the U-net architecture as its base.

Deep learning constitutes an efficient and powerful tool, but the problem of class imbalance frequently occurs in practical applications, which makes it challenging for the CNN model to extract valuable features, or it may fall into over-fitting.[29] This problem is widespread in practice, including fraud detection,[30] anomaly detection,[31] facial recognition,[32] etc. Methods for handling a class imbalance in deep learning can be grouped into two categories: hard sampling methods and soft sampling methods.[29] Hard sampling methods solve the problem of class imbalance by down-sampling positive samples or down-sampling negative samples. Jian et al.[33] proposed a sampling-based method to solve the problem of sample imbalance in mobile phone screen defect detection. Soft sampling methods mostly construct a weighted loss function that allows the minority samples to contribute more to the loss. Unlike hard sampling methods, in soft sampling methods, no sample is discarded, and the whole dataset is utilized for updating the parameters. Because the cost of defect data collection is high and the number of samples is small, in our model, we use a weighted frequency loss function to solve the problem of foreground-background class imbalance.

In summary, our proposed method of fabric defect detection makes the following contributions:

1. Different from traditional algorithms for defect classification or defect location, Mobile-Unet is a novel end-to-end defect segmentation network that achieves pixel-level fabric defect classification.
2. Compared with extant methods, Mobile-Unet has fewer model parameters and can significantly shorten detection time. It is also more suitable for online automated detection.
3. In order to solve the imbalance between defective and non-defective samples, a cross-entropy loss function weighted with median frequency balancing is used to improve the convergence speed and detection accuracy of the model.

Proposed model

In this section, the proposed network will be described in detail. The model is an efficient end-to-end framework for defect segmentation, which combines the pre-training stage and the semantic segmentation stage. The relationship between the two stages is illustrated in Figure 2. The pre-training stage (stage 1 in Figure 2) comprises training MobileNetV2 for defect classification, and the input of the network is the defect image and the label of the whole image. The semantic segmentation stage (stage 2 in Figure 2) is the segmentation stage, in which we transfer the convolution layer of the first stage as the encoder part, and then add five deconvolution layers as the decoder. The transfer of these parameters is to improve the expression ability of the segmented network on small samples. If there is more than one defect in an image, the classification result of MobileNetV2 is the category of the defect with the

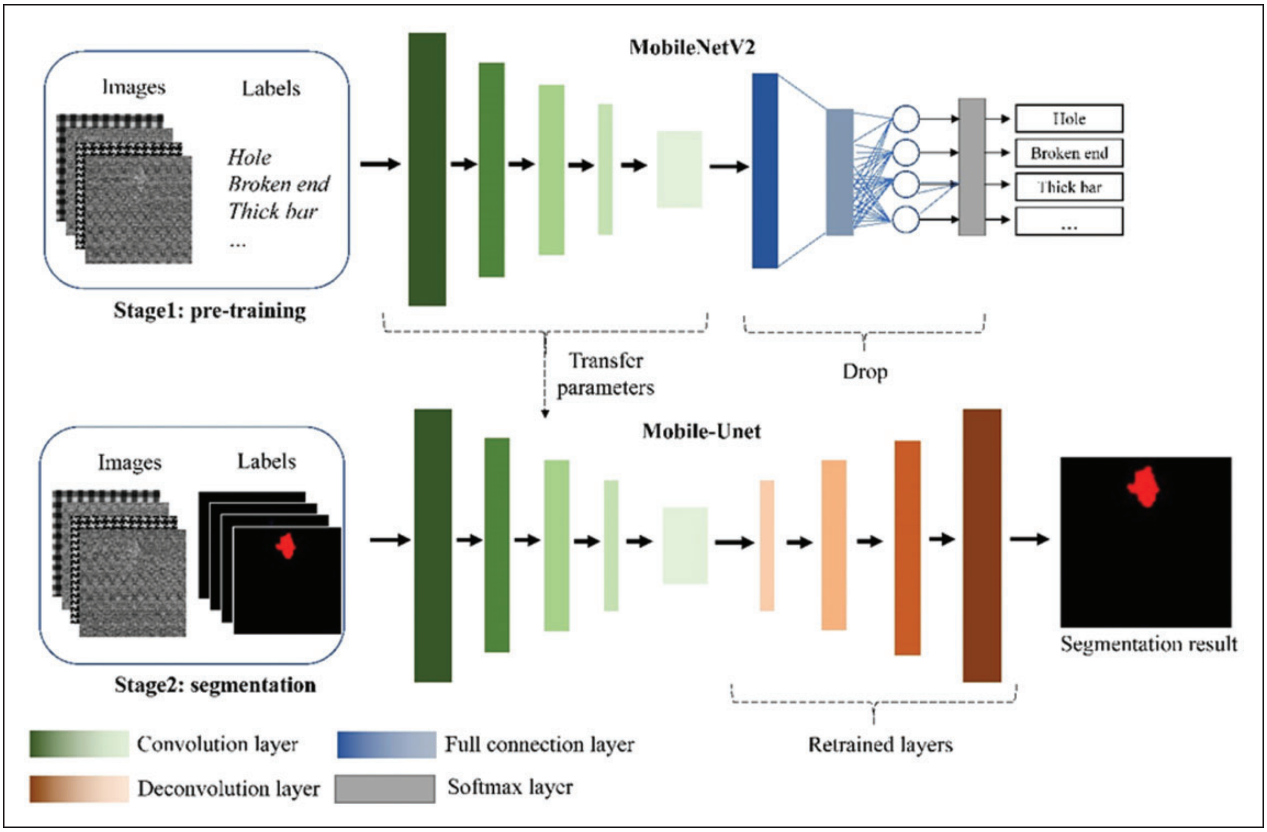

Figure 2. Process of our method.

largest area. The training dataset in the segmentation stage is the original images and the pixel-level manually marked images.

Pre-training stage

In the pre-training stage, whole fabric images are classified into their specific categories by MobileNetV2. The basic concept of MobileNetV2 is to use depth-wise separable convolutions to build a lightweight CNN. Convolution is an essential and fundamental mathematical operation in deep learning models. Numerous kinds of convolution exist,[28] one of which is depth-wise separable convolution. Depth-wise separable convolution is widely employed in real-time tasks due to two reasons: lesser parameters need to be adjusted compared with the classical convolution, which reduces possible over-fitting; and they are computationally cheaper because of fewer computations, which is more suitable for real-time vision applications. The operation models of standard convolution and depth-wise convolution are shown in Figure 3.

Depth-wise convolution constitutes two kinds of convolution blocks in MobileNetV2, called inverted residual blocks, whose structure is presented in Figure 4. One is a residual block with the stride of 1. The other is a block with a stride of 2 for downsizing. There are three layers for both types of blocks. The first layer is 1 × 1 convolution with the rectified linear unit (ReLU) 6 (ReLU6) activation function. The second layer is 3 × 3 depth-wise convolution. The third layer is also 1 × 1 convolution, but without a non-linear activation function. It is asserted that if ReLU is used again, the deep networks only have the power of a linear classifier on the non-zero volume part of the output domain.

In the pre-training stage, MobileNetV2 completes the classification of the whole image. Table 1 shows the overall architecture of MobileNetV2, where t is the expansion factor (the ratio of the number of channels in the middle layer to the number of channels in the input layer); k is the convolution kernel size; n is the block repeating number; s is the stride; *Conv* indicates the convolutional layer; *BN* indicates the batch normalization layer; N indicates the number of classes; and *C1* indicates the fully connected layer. The *D1–D5* layers are used to construct the encoder part of the segmentation model.

Semantic segmentation stage

The U-net model consists of two parts: a down-sampling part and an up-sampling part. The model fuses

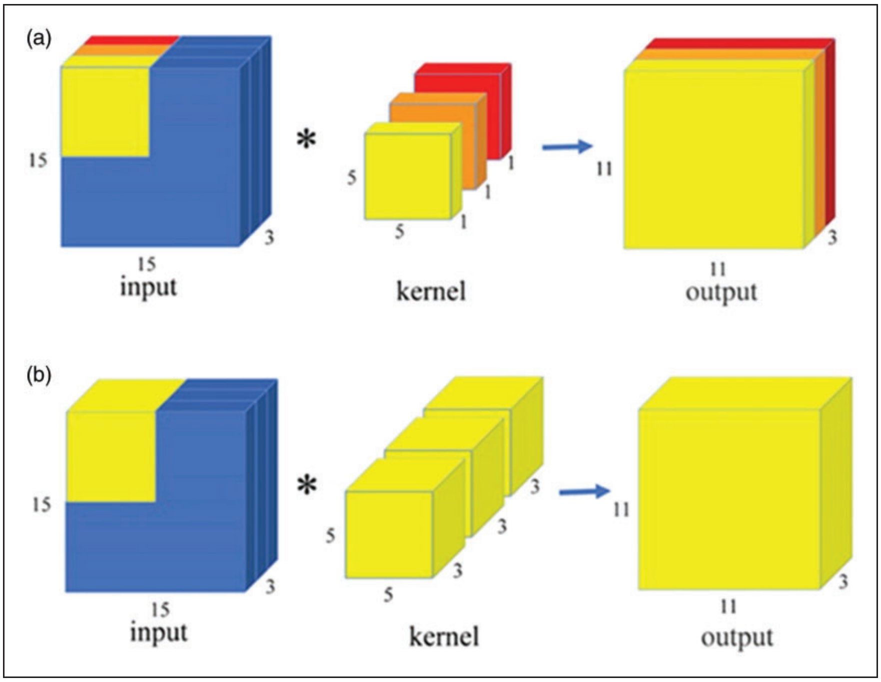

Figure 3. Two types of convolution using three kernels to transform a 15 × 15 × 3 image into an 11 × 11 × 3 image: (a) depth-wise convolution; (b) standard convolution.

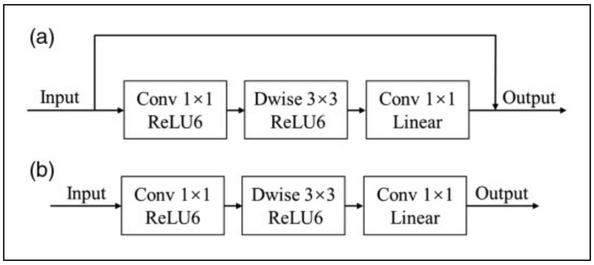

Figure 4. Inverted residual blocks: (a) stride = 2 block; (b) stride = 1 block.

high-level features and low-level features by a shortcut connection between the two parts, which improves the ability to segment image details. The down-sampling part is replaced by the feature extraction layers of pre-training MobileNetV2. The up-sampling part comprises five deconvolution layers and four inverted residual blocks, so that the input and output dimensions of Mobile-Unet are identical. The Mobile-Unet architecture is shown in Figure 5. Table 2 lists the detailed operations for each layer, where *ConvTranspose#* denotes the deconvolution layer; *Inverted Residual#* denotes inverted residual blocks; k, t, and s represent convolution kernels, expansion factors, and stride, respectively; the input of the model is the original fabric image; the output is the pixel segmentation results; and N represents the number of classes.

Improved loss function

The difference between fabric defect detection and other applications is that non-defective fabric samples are very easy to collect, while defective samples are scarce and difficult to collect, and this amount of gap results in sample imbalance. The number of defective samples is low, and the area of defect is small. Figure 6 shows the imbalance in the number of defects and background pixels. An ordinary loss function may cause the network to fall into local optimum. In order to solve the sample imbalance problem, an improved cross-entropy loss with median frequency balancing weights (CE-MFB)[34] is used. The class loss is weighted by the ratio of the median class frequency in the training set and the actual class frequency. The improved cross-entropy function is given by

$$Loss = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{c} w_c l_c^{(n)} log\left(p_c^{(n)}\right) \tag{1}$$

$$w_c = \frac{median\left(\left\{f_c|c \in C\right\}\right)}{f_c} \tag{2}$$

where N is the number of pixels in a training sample; w_c is the weight for class c; f_c is the frequency of pixels in class c; $p_c^{(n)}$ is the softmax probability of pixel n being in class c; $l_c^{(n)}$ corresponds to the label of sample n for class

Table 1. The structural configuration of MobileNetV2

Layers	Input size	Operation	t	k	n	s	Output size
Input image	–	–	–	–	–	–	(3,256,256)
D1	(3,256,256)	Conv + BN + ReLU6	–	3	1	–	(16,128,128)
		Inverted residual	1	–	1	1	
D2	(16,128,128)	Inverted residual	6	–	2	2	(24,64,64)
D3	(24,64,64)	Inverted residual	6	–	3	2	(32,32,32)
D4	(32,32,32)	Inverted residual	6	–	4	2	(96,16,16)
		Inverted residual	6	–	3	1	
D5	(96,16,16)	Inverted residual	6	–	3	2	(128,8,8)
		Inverted residual	6	–	1	1	
		Inverted residual	6	3	1	–	
C1	1280	Linear	–	–	–	–	N

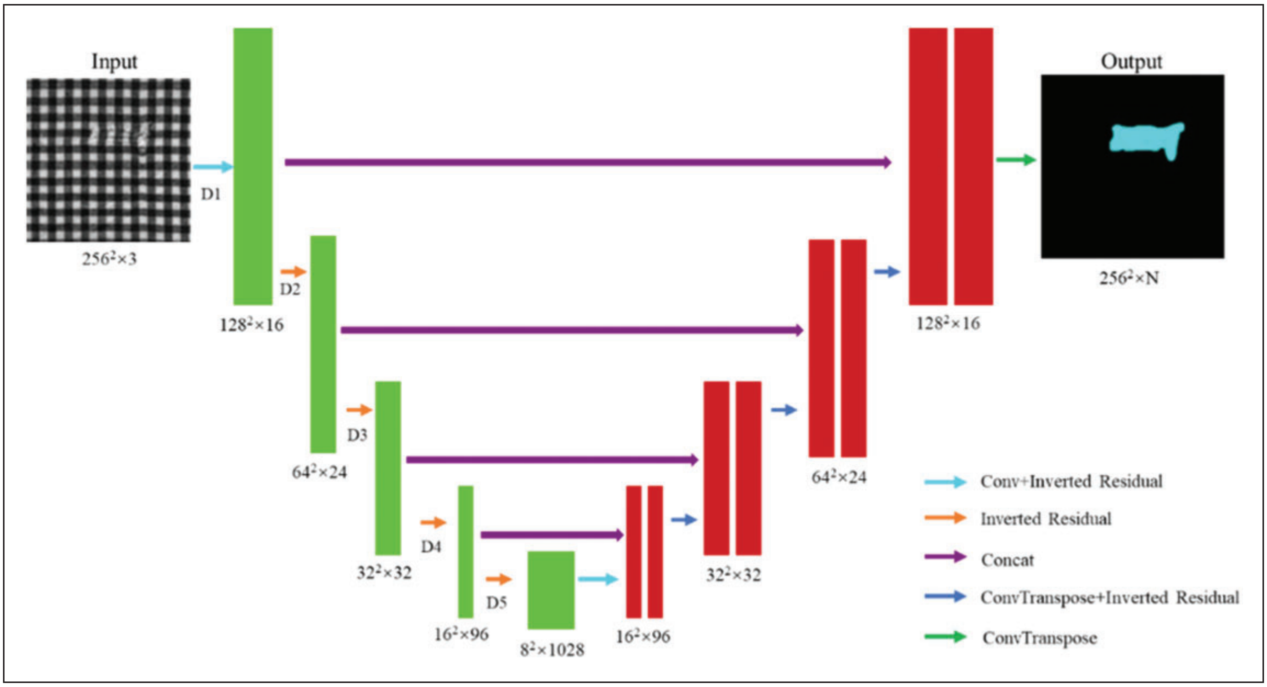

Figure 5. Architecture of Mobile-Unet.

c when the label is given in one-hot encoding; and C is the set of all classes.

Experiments and discussion

In this section, a series of experiments are conducted to evaluate the performance of the proposed Mobile-Unet. All experiments are performed on a server with an Intel Core i7 7700K processor, which has a main frequency of 4.2 GHz, 128 GB memory, and a GeForce1080Ti graphics processing unit (GPU). The software part used the Windows 10 operating system and the Pytorch deep learning framework. We use Adam optimizer to update the Mobile-Unet and initialized the weight of each layer using a Gaussian distribution with a zero mean and a standard deviation of 0.001. The initial learning rate was set to 0.001. The momentum was 0.9 and the weight decay was 5×10^{-5}. The batch size is set to 2, with a total of 400 iterations.

Establishing datasets

The performance evaluation of the proposed method (Mobile-Unet) mainly involves two fabric image benchmark databases, Fabric Images (FI)[35] and Yarn-dyed Fabric Images (YFI), in which all images are manually

Table 2. Definition and operation of each layer of Mobile-Unet

Layer	Input	Output	Output size	k	s	t
Input image	–	–	(3,256,256)	–	–	–
D1	Input image	X1	(16,128,128)	–	–	–
D2	X1	X2	(24,64,64)	–	–	–
D3	X2	X3	(32,32,32)	–	–	–
D4	X3	X4	(96,16,16)	–	–	–
D5	X4	X5	(1280,8,8)	–	–	–
ConvTranspose1	X5	L1	(96,16,16)	4	–	2
Inverted Residual1	L1 + X4	L2	(96,16,16)	–	6	1
ConvTranspose2	L2	L3	(32,32,32)	4	–	2
Inverted Residual2	L3 + X3	L4	(32,32,32)	–	6	1
ConvTranspose3	L4	L5	(24,64,64)	4	–	2
Inverted Residual3	L5 + X2	L6	(24,64,64)	–	6	1
ConvTranspose4	L6	L7	(16,128,128)	4	–	2
Inverted Residual4	L7 + X1	L8	(16,128,128)	–	6	1
ConvTranspose5 + SoftMax	L8	Output	(N,256,256)	4	–	2

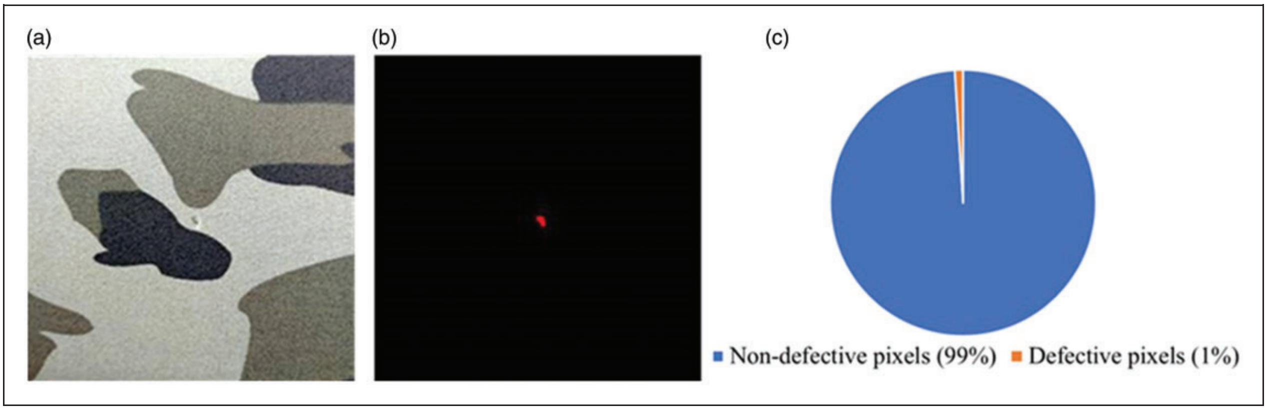

Figure 6. Imbalanced number of defects and background pixels: (a) original images; (b) ground truth; (c) comparison of the number of defective and non-defective pixels.

annotated segmentations. The FI database contains 106 red-green-blue images of size 256 × 256 from the star-, dot-, and box-patterned fabric databases provided by the Industrial Automation Research Laboratory of the Department of Electrical and Electronic Engineering at Hong Kong University. There are six defect types for each fabric pattern: broken end, hole, knot, netting multiple, thick bar, and thin bar. The YFI data are collected in a Chinese textile factory. The dataset contains 1340 images, including plain, textured, and complex printed fabrics. There are four defect types: drop stitch, blot, broken filament, and hole. In order to make the model learn more invariant image features and prevent over-fitting, data augmentation techniques such as cropping, padding, and horizontal flipping are used.

All images are strictly labeled by hand, and each pixel is labeled by one-hot encoding in order to conveniently visualize different colors representing different defects. Table 3 provides the details of each database. Some representative images and their corresponding labels are shown in Figures 7 and 8. All images have a size of 256 × 256 pixels. We shuffled all images, and then divided them into two parts: 70% as the training set and 30% as the test set. The training set is used for the model training, and the test set is used for evaluating the model performance.

Evaluation metrics

There are many metrics used to evaluate the performance of defect detection methods, such as Intersection

Table 3. Details of the datasets

Dataset	Class	Training set	Test set	Total	Color
Yarn-dyed Fabric Images	Drop stitch	250	50	300	Blue
	Blot	200	50	250	Green
	Broken filament	240	50	290	Red
	Hole	200	50	250	Yellow
	Non-defective	200	50	250	Black
Fabric Images	Broken end	100	20	120	Cyan
	Hole	100	20	120	Magenta
	Knot	100	20	120	Red
	Netting multiple	100	20	120	Green
	Thick bar	100	20	120	Yellow
	Thin bar	100	20	120	Blue
	Non-defective	100	20	120	Black

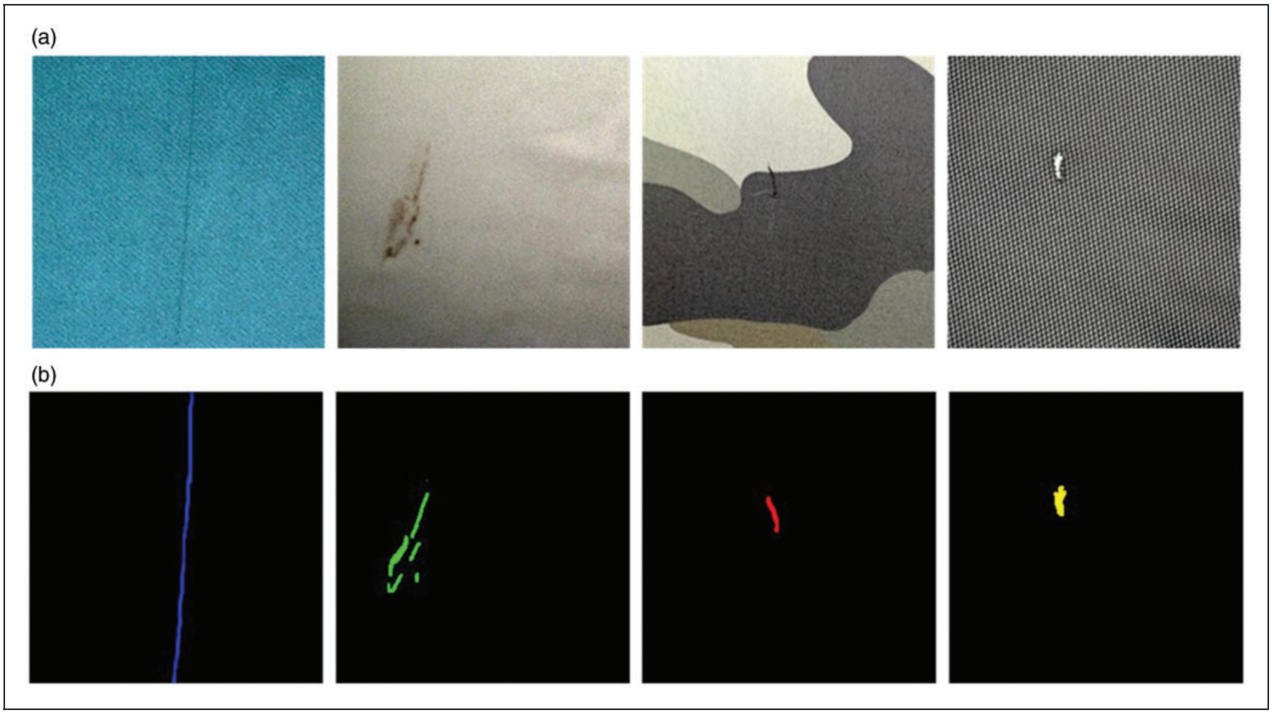

Figure 7. Some representative samples in the Fabric Images database: (a) original images; (b) ground truth.

over Union (IoU), Recall, Precision and F1-Measure.[22] These evaluation metrics are defined as follows

$$IoU = \frac{TP}{FN + TP + FP} \qquad (3)$$

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

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

$$F1 - Measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \qquad (6)$$

with reference to Figure 9, where the green rectangle represents the predicted result, and the blue rectangle represents the ground truth. The ground truth is obtained from manual marking by annotating software. According to combinations of ground truth and prediction of the result, each pixel is divided into four types: true positive (TP), false positive (FP), true

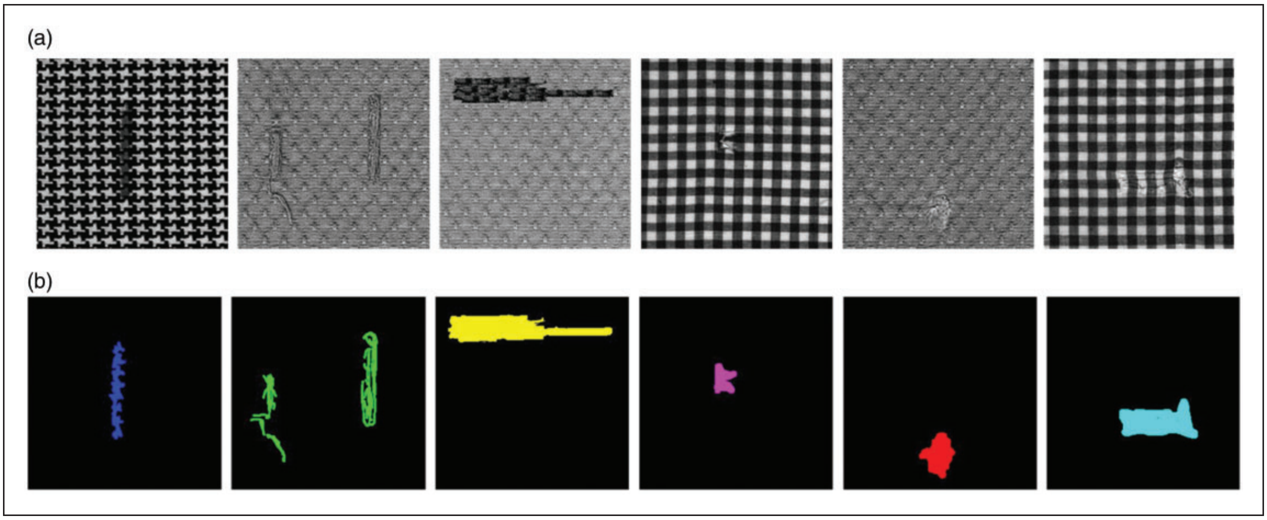

Figure 8. Some representative samples in the Yarn-dyed Fabric Images database: (a) original images; (b) ground truth.

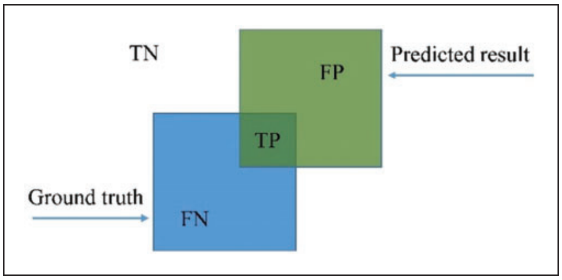

Figure 9. Relational diagram of true positive (TP), false positive (FP), true negative (TN), and false negative (FN).

negative (TN), and false negative (FN). For an excellent detection model, the predicted results are highly similar to the ground truth.

Qualitative analysis

To verify the performance of the proposed Mobile-Unet model, in this section, the detection performance of Mobile-Unet is compared with those of several extant deep learning–based image segmentation models, including FCN,[24] U-net,[36] SegNet,[25] and traditional defect detection method LGM-FC[7] and hybrid method PTIP.[37]

All of the compared methods are evaluated using the YFI and FI datasets listed in Table 3. Some of the inspection results of these methods are shown in Figure 10. Figure 10(a) to (d) are detection results in the FI dataset, and Figure 10(e) to (h) are detection results in the YFI dataset. Among these methods, FCN, SegNet, and U-net are end-to-end deep learning models to segment defects and classify defective pixels (different colors represent different defects). As shown, although FCN, SegNet, and U-net can segment defects, there are some errors in pixel classification. LGM-FC is a traditional defect segmentation method that first removes fabric image texture and then uses fuzzy C-means to segment defects. The LGM-FC obtains good performance on simple textures, such as samples in Figure 10(a), (d), (e), and (h). However, it cannot complete defect segmentation when the texture is large, as shown in Figure 10(b), (c), and (g). PTIP is a hybrid method of the traditional method and the deep learning method. It first divides the fabric image into periodic patches, and then classifies the patches using CNN to complete the segmentation. It can be seen that when the texture is regular, it can be segmented accurately, as shown in Figure 10(a) to (e). Because of the excessive dependence on the periodic texture of the fabric, however, PTIP is unable to detect the defects of non-textured fabrics, as shown in Figure 10(f) and (g). Notably, all of the results of PTIP exhibit a square segmentation region. In contrast, the proposed Mobile-Unet obtains pixel-level defect segmentation. These experimental results demonstrate that, compared with the five methods, the proposed Mobile-Unet retains the detailed information of defects more completely, and always achieves good performance on all types of defects and textures.

Quantitative analysis

Recall, Precision, F1-Measure, and IoU are utilized to quantitatively analyze the performance of model segmentation. The experimental results of YFI and FI datasets are reported in Table 4. It can be seen that, compared with other methods, Mobile-Unet is superior to other methods in both F1-Measure and IoU metrics. In addition, it can be seen that the IoU metric of the

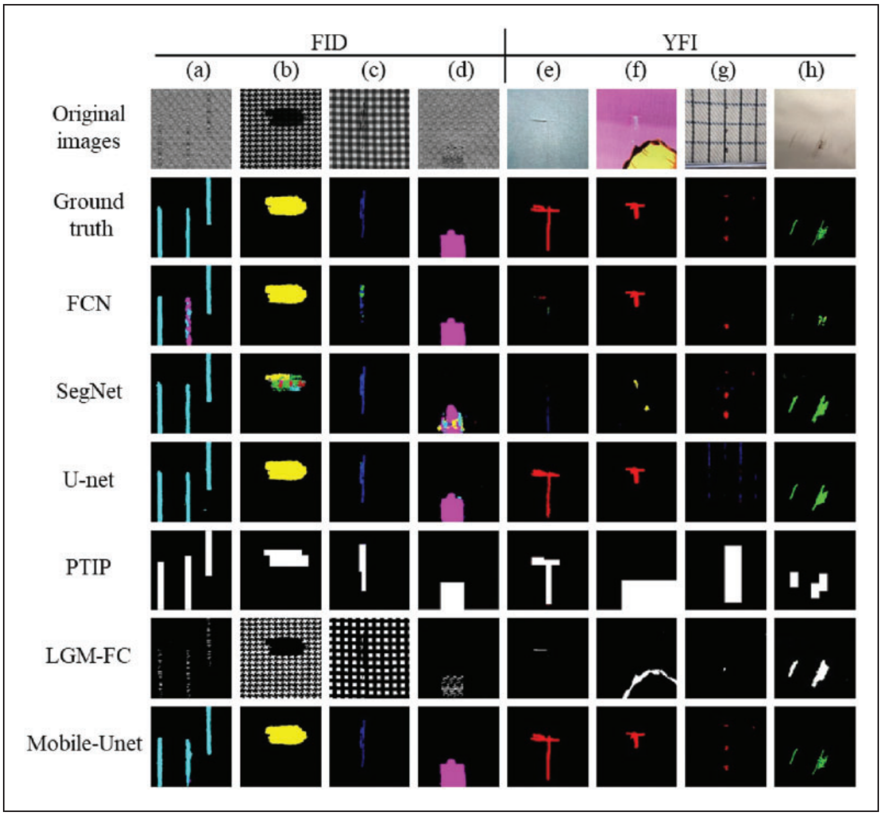

Figure 10. Comparative visual output of fabric image by different methods.

Table 4. Comparison of segmentation accuracy

Dataset	Metric	FCN	SegNet	U-net	PTIP	LGM-FC	Mobile-Unet
Yarn-dyed Fabric Images	Intersection over Union	0.89	0.74	0.89	0.72	0.70	0.92
	Recall	0.93	0.90	0.91	0.95	0.90	0.92
	Precision	0.95	0.80	0.96	0.75	0.76	0.99
	F1-Measure	0.94	0.85	0.93	0.83	0.82	0.95
Fabric Images	Intersection over Union	0.66	0.68	0.67	0.65	0.64	0.70
	Recall	0.78	0.70	0.75	0.76	0.77	0.80
	Precision	0.81	0.91	0.86	0.81	0.79	0.92
	F1-Measure	0.79	0.80	0.80	0.78	0.78	0.82

proposed method performs best on both YFI and FI datasets, which shows that, compared with other methods, the proposed method has the smallest miss detection and false detection. It is also worth noting that the two metrics are higher in the YFI dataset than in the FI dataset, because the number of FI dataset samples is less, and the types of defects are more than in the YFI dataset. In summary, the results demonstrate that Mobile-Unet can achieve state-of-the-art detection results.

Influence of the improved loss function

Figure 11 shows an increasing trend of model accuracy under different loss functions. With the training process of the model, it can be seen that the weight will eventually stabilize by using two different loss functions. However, the model with the CE-MFB loss function has shorter convergence time and higher segmentation accuracy. This experiment proves the effectiveness of the CE-MFB loss

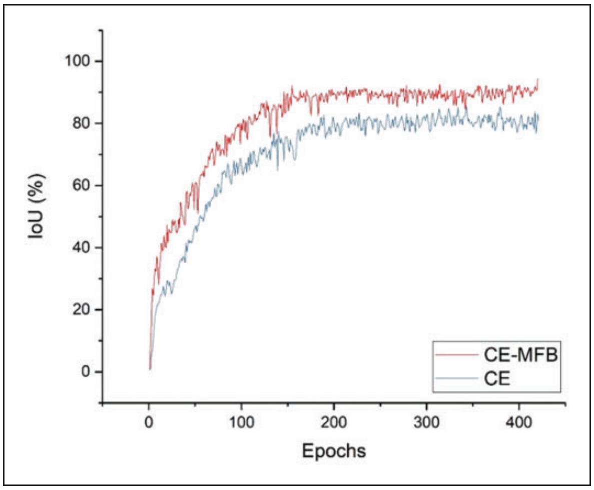

Figure 11. Intersection over Union (IoU) curves with different loss functions.
CE: cross-entropy; CE-MFB: cross-entropy loss with median frequency balancing weights.

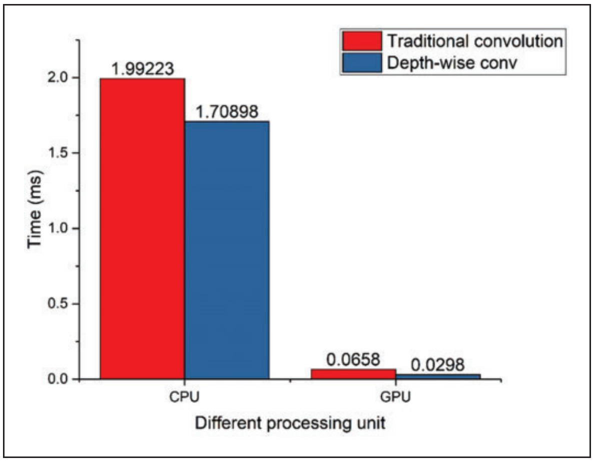

Figure 12. Performance comparison of convolution and depth-wise convolution under different processing units.
CPU: central processing unit; GPU: graphics processing unit.

function in dealing with the problem of sample imbalance.

Computational cost analysis

The proposed model is evaluated on an industrial computer, and the calculation cost is compared with the existing method. Mobile-Unet is a lightweight deep neural network constructed with depth-wise convolution. Figure 12 shows the performance comparison of traditional convolution and depth-wise convolution under different processing units (using three kernels to transform a $256 \times 256 \times 3$ image to a $128 \times 128 \times 9$ image). It reveals that the depth-wise convolution results are two times faster than the traditional convolution on the GPU platform: depth-wise convolution is 0.0298 ms, and the traditional convolution is 0.0658 ms. Similarly, on the central processing unit (CPU) platform, the separation convolution is also faster than the traditional convolution.

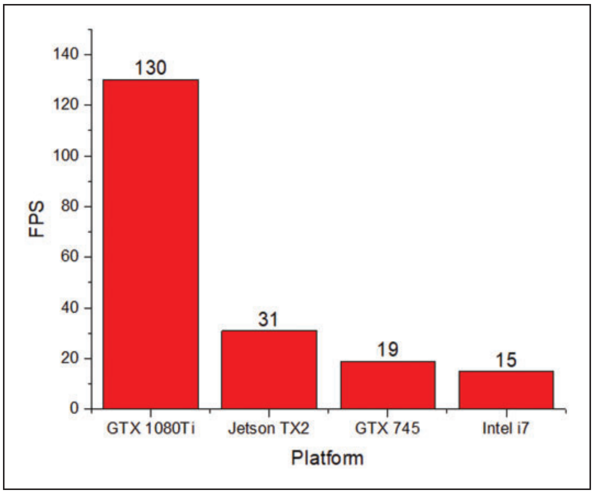

Figure 13. Detection time (frames per second) on different platforms.

In addition, the detection time of Mobile-Unet is also evaluated on different hardware platforms, including two GPU platforms (GTX 1080Ti and GTX 745), a CPU platform (Intel i7) and an embedded platform (Jetson TX2) as shown in Figure 13. It can be seen that the proposed Mobile-Unet performs well in detection time on the GPU and CPU platforms. It also can reach 31 frames per second (FPS) even on the ARM core embedded platform.

The parameters of the deep learning model are updated through back-propagation. The number of parameters determines the consumption of computing resources and training time. Similarly, the model size reflects the size of memory occupied by the model. In general, for a deep learning model, the smaller the model size and the fewer the number of parameters, the faster the speed and the better suitability for real-time applications. As shown in Table 5, our method is a lighter weight model with a small number of parameters and model size. This performance is achieved using only 4.6 million parameters for Mobile-Unet, while U-net, SegNet, and FCN have more than four times as many parameters, with 31.1 million, 29.4 million, and 18.6 million parameters, respectively. Floating point operations (FLOPs) and multiply-adds (MAdds) are also standard metrics for evaluating deep learning models.[38] FLOPs and MAdds reflect the computing resources requirements of deep learning models. The smaller the FLOPs and MAdds are, the

Table 5. Comparison of model efficiency

	FCN	SegNet	U-net	PTIP	LGM-FC	Mobile-Unet
No. of parameters (millions)	18.6	29.4	31.10	7.15	–	4.6
Model size (millions)	71.1	112.4	118.5	27.31	–	22.98
No. of floating point operations (millions)	20.11	30.85	35.35	8.34	–	0.85
No. of multiply-adds (millions)	42.87	61.91	73.91	–	–	2.05
Detection time (ms)	8.76	1074	12.73	25.0	3322	7.05

more suitable they are for resource-constrained devices. In industrial applications, detecting a single image less than 33 ms, that is, detecting a speed higher than 30 FPS can be called real-time detection. Obviously, the detection time of our model is significantly shorter than other methods, only 7.05 ms, and the detection speed is as high as 140 FPS. In summary, compared with other models, Mobile-Unet has lower time complexity, which means that this model can save more computing resources and is more efficient.

Conclusion

Inspired by MobileNetV2 and U-net, an effective deep learning model Mobile-Unet for fabric defect segmentation is proposed. Mobile-Unet has been evaluated on YFI and FI datasets.

Compared with SegNet and U-net, Mobile-Unet required 10 times fewer FLOPs, has five times fewer parameters, and achieved high segmentation accuracy. In practical applications, the number of defective samples is small, and the data will face the problem of sample imbalance, which makes the network difficult to train. In order to solve this problem, the improved loss function is used to make the model easier to train. However, as a supervised method, its training requires a large amount of manually annotated data, and thus developing a weak supervised and unsupervised defect segmentation method constitutes the direction of our future research.

Declaration of conflicting interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Youth Innovation Team of Shaanxi Universities, and the Scientific Research Program Funded by the Shaanxi Provincial Education Department[Grant No. 19JC018] ,the National Natural Science Foundation of China [Grant No. 61902302] and the Graduate Scientific Innovation Fund for Xi'an Polytechnic University [Grant No. chx2020014].

ORCID iD

Junfeng Jing https://orcid.org/0000-0001-6646-3698

References

1. Yapi D, Allili MS and Baaziz N. Automatic fabric defect detection using learning-based local textural distributions in the contourlet domain. *IEEE Trans Autom Sci Eng* 2018; 15: 1014–1026.
2. Wu Y, Zhou J, Akankwasa NT, et al. Fabric texture representation using the stable learned discrete cosine transform dictionary. *Text Res J* 2019; 89: 294–310.
3. Mallik-Goswami B and Datta AK. Detecting defects in fabric with laser-based morphological image processing. *Text Res J* 2000; 70: 758–762.
4. Jing J, Yang P, Li P, et al. Supervised defect detection on textile fabrics via optimal Gabor filter. *J Ind Text* 2014; 44: 40–57.
5. Li P, Liang J, Shen X, et al. Textile fabric defect detection based on low-rank representation. *Multimed Tools Appl* 2019; 78: 99–124.
6. Zhang K, Yan Y, Li P, et al. Fabric defect detection using salience metric for color dissimilarity and positional aggregation. *IEEE Access* 2018; 6: 49170–49181.
7. Zhang H, Ma J, Jing J, et al. Fabric defect detection using L0 gradient minimization and fuzzy c-means. *Appl Sci* 2019; 9: 3506.
8. Kang X and Zhang E. A universal defect detection approach for various types of fabrics based on the Elo-rating algorithm of the integral image. *Text Res J* 2019; 89: 4766–4793.
9. Guan S and Shi H. Fabric defect detection based on the saliency map construction of target-driven feature. *J Text Inst* 2018; 109: 1133–1142.
10. Lu Y. Artificial intelligence: a survey on evolution, models, applications and future trends. *J Manag Anal* 2019; 6: 1–29.
11. Tabernik D, Šela S, Skvarč J, et al. Segmentation-based deep-learning approach for surface-defect detection. *J Intell Manuf* 2020; 31: 759–776.
12. Alguliyev RM, Aliguliyev RM and Abdullayeva FJ. Privacy-preserving deep learning algorithm for big personal data analysis. *J Ind Inf Integr* 2019; 15: 1–14.

13. Li Y, Zhang D and Lee DJ. Automatic fabric defect detection with a wide-and-compact network. *Neurocomputing* 2019; 329: 329–338.
14. Jing J, Dong A, Li P, et al. Yarn-dyed fabric defect classification based on convolutional neural network. *Opt Eng* 2017; 56: 093104.
15. Wei B, Hao K, Tang XS, et al. A new method using the convolutional neural network with compressive sensing for fabric defect classification based on small sample sizes. *Text Res J* 2019; 89: 3539–3555.
16. Zhang H, Zhang L, Li P, et al. Yarn-dyed fabric defect detection with YOLOV2 based on deep convolution neural networks. In: *IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS)*. Enshi, China: IEEE, 2018, pp.170–174.
17. Li Y, Huang H, Xie Q, et al. Research on a surface defect detection algorithm based on MobileNet-SSD. *Appl Sci* 2018; 8: 1678.
18. Li S, Zhao X and Zhou G. Automatic pixel-level multiple damage detection of concrete structure using fully convolutional network. *Comput Civ Infrastruct Eng* 2019; 34: 616–634.
19. Miao X, Wang J, Wang Z, et al. Automatic recognition of highway tunnel defects based on an improved U-net model. *IEEE Sens J* 2019; 1748: 1–1.
20. Tao X, Zhang D, Ma W, et al. Automatic metallic surface defect detection and recognition with convolutional neural networks. *Appl Sci* 2018; 8: 1575.
21. Hu G, Huang J, Wang Q, et al. Unsupervised fabric defect detection based on a deep convolutional generative adversarial network. *Text Res J* 2020; 90: 247–270.
22. Mei S, Wang Y and Wen G. Automatic fabric defect detection with a multi-scale convolutional denoising autoencoder network model. *Sensors* 2018; 18: 1064.
23. Li Y, Zhao W and Pan J. Deformable patterned fabric defect detection with Fisher criterion-based deep learning. *IEEE Trans Autom Sci Eng* 2017; 14: 1256–1264.
24. Shelhamer E, Long J and Darrell T. Fully convolutional networks for semantic segmentation. *IEEE Trans Pattern Anal Mach Intell* 2017; 39: 640–651.
25. Badrinarayanan V, Kendall A and Cipolla R. SegNet: a deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans Pattern Anal Mach Intell* 2017; 39: 2481–2495.
26. Falk T, Mai D, Bensch R, et al. U-Net: deep learning for cell counting, detection, and morphometry. *Nat Methods* 2018; 16: 67–70.
27. Kumar P, Nagar P, Arora C, et al. U-SegNet: fully convolutional neural network based automated brain tissue segmentation tool. In: *IEEE International Conference on Image Processing (ICIP)*. Athens, Greece: IEEE, pp.3503–3507.
28. Sandler M, Howard A, Zhu M, et al. MobileNetV2: Inverted residuals and linear bottlenecks. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT, USA: IEEE, 2018, pp.4510–4520.
29. Johnson JM and Khoshgoftaar TM. Survey on deep learning with class imbalance. *J Big Data* 2019; 6: 27.
30. Abdallah A, Maarof MA and Zainal A. Fraud detection system: a survey. *J Netw Comput Appl* 2016; 68: 90–113.
31. Napoletano P, Piccoli F and Schettini R. Anomaly detection in nanofibrous materials by CNN-based self-similarity. *Sensors* 2018; 18: 209.
32. Soleymani R, Granger E and Fumera G. Progressive boosting for class imbalance and its application to face re-identification. *Expert Syst Appl* 2018; 101: 271–291.
33. Jian C, Gao J and Ao Y. Imbalanced defect classification for mobile phone screen glass using multifractal features and a new sampling method. *Multimed Tools Appl* 2017; 76: 24413–24434.
34. Kampffmeyer M, Salberg AB and Jenssen R. Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. Las Vegas, NV, USA: IEEE, 2016, pp.1–9.
35. Tsang CSC, Ngan HYT and Pang GKH. Fabric inspection based on the Elo rating method. *Pattern Recognit* 2016; 51: 378–394.
36. Ronneberger O, Fischer P and Brox T. U-net: convolutional networks for biomedical image segmentation. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI*. Springer, 2015, pp.234–241.
37. Jing JF, Ma H and Zhang HH. Automatic fabric defect detection using a deep convolutional neural network. *Color Technol* 2019; 135: 213–223.
38. Paszke A, Chaurasia A, Kim S, et al. ENet: a deep neural network architecture for real-time semantic segmentation. arXiv preprint, arXiv:1606.02147 2016; 1–10.