IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 23, 2019, accepted March 10, 2019, date of publication March 18, 2019, date of current version April 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2905638

An Efficient Method for DPM Code Localization Based on Depthwise Separable Convolution

YUSHENG LI[1], YONG TIAN[1], JINDONG TIAN[1], AND FEI ZHOU[2]
[1]College of Physics and Optoelectronic Engineering, Shenzhen University, Shenzhen 518060, China
[2]College of Information Engineering, Shenzhen University, Shenzhen 518060, China

Corresponding authors: Jindong Tian (jindt@szu.edu.cn) and Fei Zhou (fei.zhou@szu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 51775352, Grant 61727814, and Grant 61803268, and in part by the Science and Technology Plan Project of Shenzhen under Grant JCYJ20170412110241478.

ABSTRACT With the popular application of direct part mark (DPM) technology, DPM code inspection has been a hot issue in the machine vision. It mainly consists of two steps, namely, localization and decoding. DPM code localization is a key and complex step in the DPM code inspection. However, the traditional localization methods suffer from complex imaging environment, involving various imaging background, illumination, imaging distance, and exposures. Furthermore, the target itself, i.e., the DPM code, could be severely polluted or worn. Aiming at improving the performance and robustness of DPM code localization, an efficient method with depthwise separable convolution is proposed in this paper. The optimized network model has the advantages of a few parameters, high computational efficiency, high precision localization, and good generalization ability. Meanwhile, the precision of the DPM code region is improved with the help of multi-scale prediction. The experiments on our DPM code localization database demonstrate the effectiveness and flexibility of the proposed method in comparison with the YOLOv3 network and the Tiny_YOLO network. Furthermore, the proposed method can estimate the exposure level of the DPM code region, which is benefiting to the DPM code recognition and enables the adaptive ability.

INDEX TERMS Direct part mark, data matrix, depthwise separable convolution, deep learning.

I. INTRODUCTION

With the rapid development of information technology, the barcode technology is more and more popular. DPM (Direct Part Mark) is a special label fabrication technology. The label is directly made on the object surface by stippling, etching or spraying, without any label carrier such as paper, which is widely used in the field of industrial manufacturing. Usually, only a small area can be used to label the DPM code. So the DPM needs to choose a code with high coding capacity. DM (Data Matrix) code is the most commonly used type of the DPM code. Because it has the characteristics of large coding capacity, high density and good information security. With the same size and density, the information contained in DOM is more than that in other two dimensional (2D) barcodes. Since the DPM code is fabricated on the object directly, the localization of the DPM code can be highly influenced by the wear, illumination, dirty, material, code sizes, and so on. Thus, it is difficult to localize the DPM codes, especially in the complex scenes as mentioned above. However, the practical application has high requirements on the accuracy and speed of the DPM code recognition.

An optimal edge combination method [1] based on Hough transform domain was proposed to extract the edge features of DM codes firstly and then filter the boundary points using the prior knowledge to localize the DM code. However, this method is not suitable for localizing the DM code with damaged edge or the multiple edge features. The multi-channel Gabor filtering method [2] was demonstrated to be effective for long-distance barcode localization. Whereas its precision is declined when the code is small or the view direction changes. Based on AdaBoost, an adaptive SpatialBoost algorithm [3] was proposed to combine image texture information and spatial information adaptively to detect 2D barcode. Experiments on 60 test sets showed that the precision to the 2D barcode reached to 100%. However, this algorithm was only tested in a small number of dataset collected from metal surface. In order to resolve the problem of low resolution, variation Bayesian framework [4] was adopted to get super resolution barcode image. However, it is mainly suitable for

The associate editor coordinating the review of this manuscript and approving it for publication was Fan Zhang.

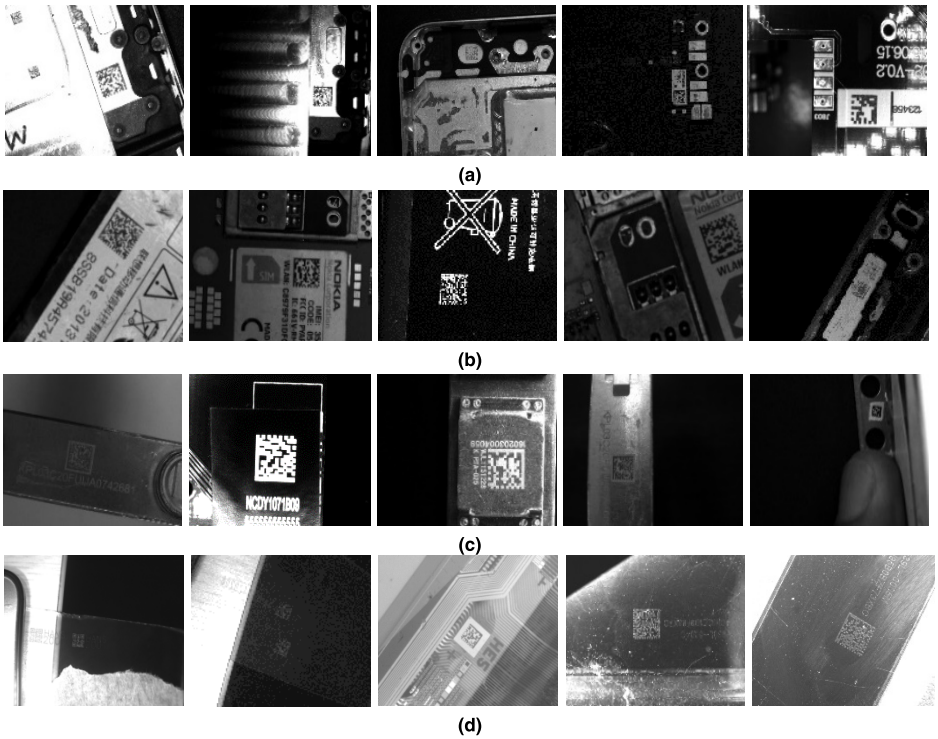


FIGURE 1. DPM codes on various imaging background. (a) DPM codes on cell phones, (b) DPM codes on the batteries, (c) DPM codes on the metal, (d) DPM codes on the transparent material and screen.

the barcode in the simple scene. In a word, the traditional methods have a satisfactory barcode localization in the simple scenes, however, their localization precision and generalization abilities become worse under complex imaging environments. Figure 1 shows examples of complex environments, including cell phones, batteries, metal, transparent materials and screens. It is clear from Figure 1 that even on the surface of the same type of objects, the localization of DPM can be very different due to other factors, such as the illumination, imaging distance, and exposures.

In recent years, deep learning has made breakthrough in the computer vision. For example, convolutional neural networks (CNN) has been proved to be highly effective for image classification [5]–[8], objection detection [9]–[14], semantic segmentation [13], [14]. So far, many researches on QR (Quick Response) code location using deep learning have been reported. For example, a conventional and deep rectifier neural network [15] was proposed to localize the QR code. The method segmented the image into many blocks and processed every block using neural network to get a probability matrix. Based on the probability matrix, the QR code was localized. Further, this method was extended to barcode localization [16] and got a good result. A two-step algorithm [17] was proposed to localize the QR code. Firstly, the Viola-Jones framework was used to train cascade classifiers to output part of patterns. Then, the evaluation of consistency between detected patterns and QR code spatial arrangement was performed to complete the QR code localization. A convolutional neural network [18] was proposed to detect part of QR code and the position was confirmed with majority voting. Then the QR code was segmented and localized from background based on traditional image process. However, there is no literature about the DPM code localization using convolutional neural network. Besides, DPM code is usually applied in various industrial scenes. It is a great challenge to accurately localize the DPM code in different situation because of the complexity of industrial environments.

In this paper, we proposed an efficient method for DPM code localization based on depth-wise separable convolutions. DM codes fabricated by DPM technology on the materials of cellphone case, battery, steel, the transparent material and LCD screen are selected as the tested objects, and to create industrial DPM code localization dataset. To meet the real-time requirements of industrial DPM code localization, we propose a feature extraction network based on DSC (depth-wise separable convolution) [19] method. Compared with the traditional convolution method, the DSC method reduces the amount of both parameters and calculation, and significantly improves the calculation speed without performance decrease. To improve the accuracy of DPM code region localization in DPM code dataset, we use feature pyramid network as the output part of the whole model, realizing multi-scale prediction of DPM code. Meanwhile, considering the complicated scenes such as uneven illumination and overexposure in the industry, the exposure degree of the DPM code area is evaluated and subdivided into under-exposure, normal exposure and over-exposure, which can provide favorable prior knowledge for the imaging and

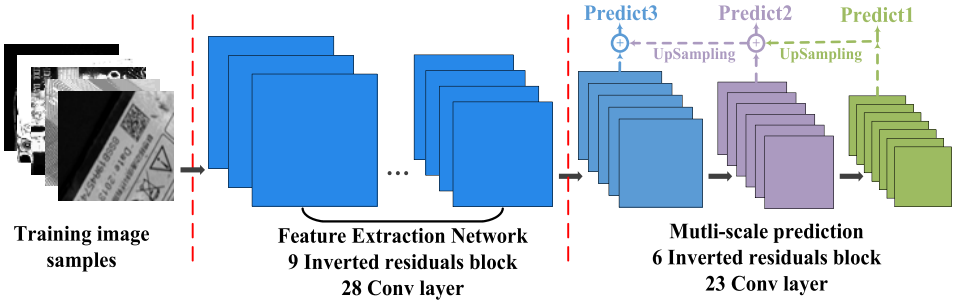


FIGURE 2. Framework of DPMLCN.

reading of subsequent DPM code. The rest of this paper is organized as follow: the network framework is presented in Section 2, DPM code localization dataset construction process is described in Section 3, and network training process is shown in Section 4. In Section 5, the experimental results are given and analyzed, and finally the paper is concluded in Section 6.

II. NETWORK FRAMEWORK

With the rapid development of deep learning, lots of excellent objection detection frameworks have been proposed to realize image recognition. Among them, YOLOv3 [12] is one of the outstanding network models for objection detection, but the computational cost during both training and recognition processes is large. As an improved version of YOLO [20], Tiny_YOLO is a faster network model for objection detection, however, it reduces the recognition accuracy and generalization ability. Accordingly, these network models cannot achieve a good balance between recognition accuracy and computational speed. Therefore, this paper focuses on developing a DPM code localization network that has both high accuracy and fast computational speed.

A. CONSTRUCTION OF DPM CODE LOCALIZATION NETWORK

Due to DPM codes are mostly applied in industry, DPM codes localization network must satisfy the real-time requirement of industry application. To solve this problem, a network based on DSC [21] was proposed to improve the computational speed while keeping the high accuracy. However, there are many scales of DPM codes in the DPM code localization dataset used in this paper. If the network only outputs DPM code localization boundary box on a single-layer network, there will be a good localization effect on one scale of DPM codes, while have a large localization deviation on other scales of DPM codes. Aiming at this problem, an objection detection model based on the feature pyramid network was proposed to improve the accuracy on objects of different scales [22], whereas the computational speed cannot meet the real-time requirements.

Since the parameter amount of the DSC method is less than that of the traditional convolution method, the computational speed is faster than the traditional convolution method with the same accuracy. Meanwhile, as the output of the network, the feature pyramid network locates DPM code region on the last three network layers to realize multi-scale prediction and improve the localization accuracy of multi-scale DPM code in the dataset. Based on the advantages of DSC and feature pyramid network, we propose a DPMCLN (DPM code localization network) by combining DSC and feature pyramid networks to improve the computational speed and localization accuracy of multi-scale DPM codes without reducing the accuracy. The architecture of DPMCLN is provided in Figure 2.

B. MODEL BLOCK DESIGN

Since the block can simplify the network design, the network model of convolutional neural network is usually designed using block. Hence, a good block is critical to the network model design. The inverted residuals structure of MobileNetv2 [23] is employed in this paper to design the block, as shown in Figure 3(a). For the block whose stride is one, the channel dimensions of feature maps are upgraded using input feature maps by 1×1 convolution. The features are activated by the activation function of Relu6. Then the depthwise (DW) convolution and Relu6 activation are performed to extract the useful information of the feature maps in each channel. The information cross channels is fused by 1×1 convolution. Finally, the input layer is added to the output layer after the previous 1×1 convolution by shortcut way to form a residual structure. For the block whose stride is two, it is only to reduce the sizes of feature maps.

The principle of the inverted residual structure is to "expand" the input layer channel and then perform DW convolution. It is benefiting to get more feature maps in the DW convolution layer and improve the expression ability of network model. The DW convolution method can reduce the amount of convolution calculation, which speeds up the calculation of the model. However, in the original inverted residual structure, the activation function Relu6 is directly activated after the DW convolution, which would destroy its original features. Because the DW convolution is performed by channel-wise fashion, the information only can be exchanged in one channel and cross-channel information exchange is impossible. When the DW convolution is performed for channel information fusion, the original features

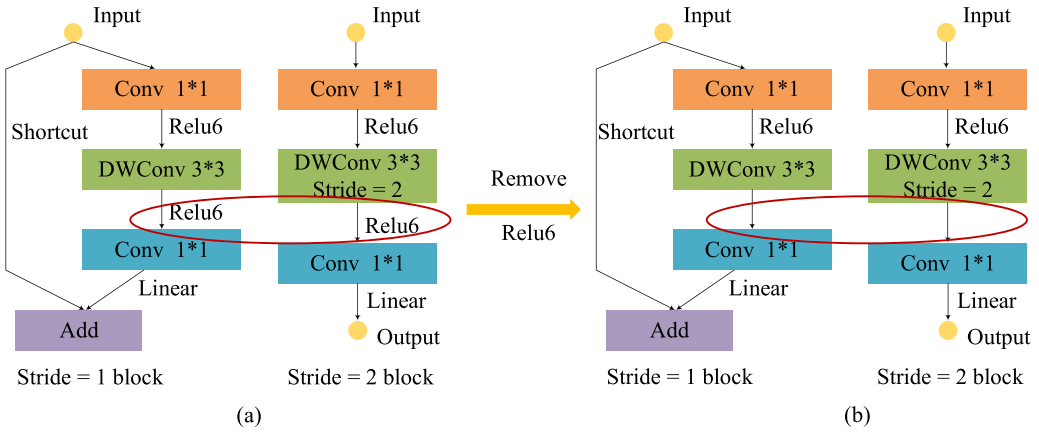


FIGURE 3. Block structures comparison of stride=1 and stride=2. (a) MobileNet V2. (b) DPMCLN.

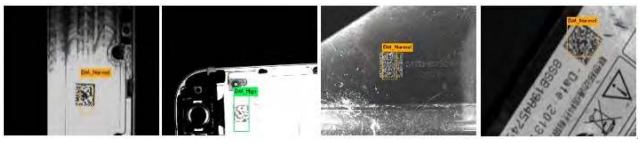


FIGURE 4. Example of inaccurate DPM code location.

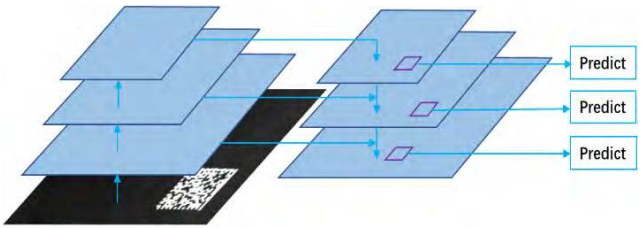


FIGURE 5. Feature pyramid network.

of the DW convolution may be missed. So, we design a network DPMLCN to improve the inverted residuals structure, as presented in Figure 3(b). Comparing with Figure 3(a), the activation function Relu6 after DW convolution is removed to protect the original information. The experimental results indicated that the precision can be improved by 2.5% simply by removing the activation function.

C. MULTI-SCALE PREDICTION

Suffered from imaging orientation, imaging distance, camera distortion and so on, the DPM codes in the dataset have different sizes (multi-scale). If the DPM code localization only depends on the feature map in the topmost layer, the localization box may outline the DPM code imperfectly, as shown in Figure 4.

In the low layers of the convolutional neural network, the feature maps have little semantic information and their fine grit is small, which is suitable for small objects. Whereas in the high layers, the feature maps have large fine grit and is suitable for large objects. Based on the properties of both the low and high layers network, the feature pyramid network [22] is employed to get an adaptive method for different sizes of the DMP codes. As shown in Figure 5, all the last three layers are outputted to improve the accuracy of the DPM code boundary.

After the multi-scale prediction, multiple boundary boxes are obtained. It is necessary to choose the best one. The key step in this process is to update the score for each boundary box. The updating process with non-maximum algorithm (NMS) [24] can be formulated as:

$$s_i = \begin{cases} s_i, & iou(M, b_i) < N_t \\ 0, & iou(M, b_i) \geq N_t \end{cases} \tag{1}$$

where M is the boundary box getting the highest score, b_i is the boundary box filtered from initial detection result, N_t is the threshold value of intersection over union (iou), and iou can be defined as:

$$iou = \frac{area\,(A) \cap area\,(B)}{area\,(A) \cup area\,(B)} \tag{2}$$

where A is the predicted bounding box, B is the ground truth bounding box.

When iou is larger than the threshold value N_t, the score of the boundary box is set to zero. However, iou may still be larger than the threshold value N_t as two objects close nearly, even though the confidence coefficient of boundary box is low, resulting in the leak detection. Thus, the Soft-NMS [25] algorithm is used to delete the repetitive boundary box and get the best boundary box as:

$$s_i = s_i e^{-\frac{iou(M,b_i)^2}{\sigma}}, \quad \forall b_i \notin D \tag{3}$$

where D is the boundary box collection.

With the Soft-NMS algorithm, when iou is larger than the threshold value N_t, the score of the boundary box is set to be the product between the score and exponential damping coefficient rather than zero. It can avoid the leak detection to the target object. As presented in Table 1, compared with the traditional NMS algorithm, the Soft-NMS algorithm can improve the recall rate about 2.6%. Figure 6 shows the DPM code prediction result in multiple sizes based on the Soft-NMS algorithm. It can be seen that the precision of boundary is obviously improved in comparison with Figure 4.

TABLE 1. Recall rate comparison between the NMS and Soft-NMS algorithm.

Algorithm	Dataset	DPM code numbers	Corrected localization DPM code numbers	Recall
NMS	2500	3375	3256	96.5%
Soft-NMS	2500	3375	3358	99.5%

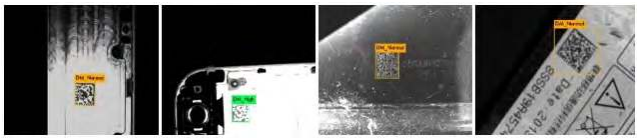


FIGURE 6. DPM code prediction results in multiple sizes using Soft-NMS algorithm.

D. LOSS FUNCTION

In this paper, we unify the separate components of object detection into a single neural network. The proposed DPMLCN uses features extracted from the entire image to predict each bounding box. It also predicts all bounding boxes across all classes for an image simultaneously. This means that DPMLCN can realize end-to-end training and real-time speed while maintaining high precision.

As shown in Figure 7, DPMLCN divides the input image into an $S \times S$ grid. If the center of an object falls into a certain grid cell, then this grid cell is responsible for detecting the corresponding object. Each grid cell predicts B bounding boxes and confidence scores for those boxes. These confidence scores reflect the confidence level that an object is contained in the bounding box. Each bounding box consists of 5 predictions: x, y, w, h, and confidence c. The (x, y) represents the center coordinate of the box relative to the bounds of the grid cell. The width and height are predicted relative to the whole image. Finally, the confidence prediction can be represented as $P_r\,(Object) * IOU_{pred}^{truth}$. If no object exists in that cell, the confidence score should be zero. Otherwise the confidence score should be equal to *iou* between the predicted box and the ground truth. Each grid cell also predicts C conditional class probabilities, $P(\text{Class}_i|\text{Object})$. These probabilities are conditioned on the grid cell containing an object. We only predict one set of class probabilities per grid cell regardless of the number of boxes B.

For evaluating DPMLCN on DPM code localization dataset (DPMCLD), we use the K-means clustering algorithm to determine the difference scales of DPM code on DPMCLD. According to the number of DPM codes sorted by different scales, we select the top 9 clustering centers, and then we set B to 3 as each grid cell on the last three output layers of the network has 3 bounding boxes priors. Meanwhile, based on interactive experiments, it is found that the network performs best as the feature map sizes of the last three output layers are set to 13×13, 26×26 and 52×52 respectively, so we set S to 13, 26, 52 on three different output layers. The purpose of this work is not only to locate the DPM code area, but also to evaluate its exposure level. Therefore, the DPM codes are classified into three categories: under-exposure, normal exposure and over-exposure. The definition of loss function is improved by locating the DPM code regions with three different exposures. Thus, DPMCLD has three labeled classes and then C is set to 3. The final prediction is an $S \times S \times 18$ tensor, where S is 13, 26 or 52, respectively corresponding to the last three output layers of the network.

The training process is to optimize the multi-part loss function consisting of the bounding box center coordinate error, the bounding box width and height error, the confidence value error and the class probability error. The loss function is commonly expressed in the form of mean square error as shown in (4).

$$
\begin{aligned}
&loss \\
&= \lambda_{cood} \sum_{i=0}^{S^2} \sum_{j=0}^{B} R_{ij}^{obj} \left[\left(x_i - \hat{x}_i\right)^2 + \left(y_i - \hat{y}_i\right)^2 \right] \\
&\quad + \lambda_{cood} \sum_{i=0}^{S^2} \sum_{j=0}^{B} R_{ij}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2 \right] \\
&\quad + \sum_{i=0}^{S^2} \sum_{j=0}^{B} R_{ij}^{obj} \left(C_i - \hat{C}_i\right)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \left(1 - R_{ij}^{obj}\right) \\
&\quad \times \left(C_i - \hat{C}_i\right)^2 + \sum_{i=0}^{S^2} R_i^{obj} \sum_{c \in classes} \left(p_i\,(c) - \hat{p}_i\,(c)\right)^2 \qquad (4)
\end{aligned}
$$

S - *The number of grids in which the input image is divided* (*here is* 13, 26 *and* 52)

B - *Number of bounding boxes predicted by each grid* (*here is* 3)

$\mathbb{R}_i^{obj}$ - *the value of* 1 *if object appears in cell i, otherwise* 0.

$\mathbb{R}_{ij}^{obj}$ - *the value of* 1 *if the jth bounding box predictor in cell i is "responsible" for that prediction, otherwise* 0.

x_i, y_i - *the center coordinate of boundary box in cell i*

w_i, h_i - *The width and height of the bounding box in cell i* (*wrt the whole image*)

λ_{cood} - *Weight coefficient of the center coordinate, width and height error of the bounding box* (*here is* 4)

λ_{noobj} - *Confidence error for no object in the cell* (*here is* 0.6)

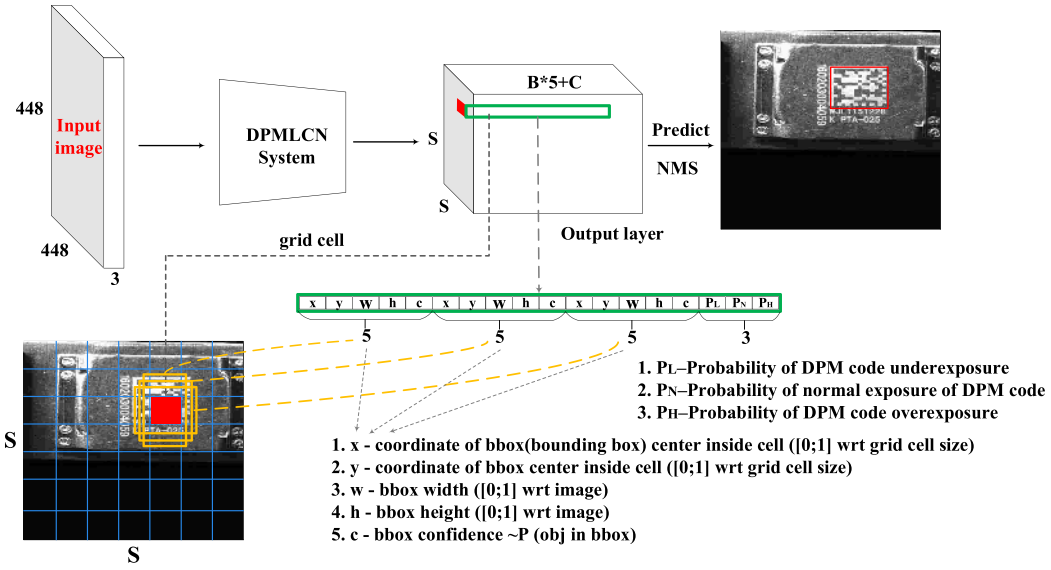


FIGURE 7. Prediction process of DPM code localization boundary box.

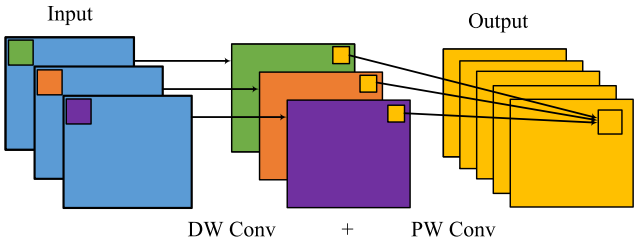


FIGURE 8. Framework of depthwise separable convolution.

C_i - *the value of IOU between the predicted box and the ground truth if object appears in cell i, otherwise* 0.

$p_i(c)$ - *probabilities are conditioned on the grid cell i containing an object*

where *classes* label includes underexposure (DM_Low), normal exposure (DM_Normal), and overexposure (DM_High) of DPM code area labeled in DPMCLD.

E. DISCUSSION ON DW

Depthwise separable convolution consists of depthwise (DW) convolution and pointwise (PW) convolution as shown in Figure 8. The DW convolution convolutes the input feature maps by channel-wise fashion and the PW convolution convolutes the feature maps of all channels with a 1×1 convolution kernel.

The computational advantage of depthwise separable convolutions can be verified by comparing parameter numbers and theoretical calculation amounts between the traditional convolution and depthwise separable convolutions. To be brief, we set the convolution kernel to be $K_h \times K_w$, the input and output channels as C_{in} and C_{out} respectively. The resolution of input image is $H \times W$. *Params* denotes weights of the network without the bias. $FLOP_s$ denotes theoretical calculation amounts, not taking the additions into attention.

a) The parameter numbers and theoretical calculation amounts in the traditional convolution are:

$$\begin{aligned} &Params:\ K_h \times K_w \times C_{in} \times C_{out} \\ &FLOPs:\ K_h \times K_w \times C_{in} \times C_{out} \times H \times W \end{aligned} \quad (5)$$

b) The parameter numbers and theoretical calculation amounts in the depthwise separable convolutions are:

$$\begin{aligned} &Params:\ K_h \times K_w \times C_{in} \times C_{out}/C_{in} = K_h \times K_w \times C_{out} \\ &FLOPs:\ K_h \times K_w \times C_{out} \times H \times W \end{aligned} \quad (6)$$

According to (5) and (6), it can be found that the parameter numbers and theoretical calculation amounts in the depthwise separable convolutions are reduced by C_{in} times.

III. DPM CODE LOCALIZATION DATASET

The process of constructing DPM code localization dataset includes two key parts. The first part is collecting a diverse set of DPM code images in different scenarios. The second part is annotating large amounts of collected images to obtain a DPM code localization dataset.

A. SOURCE OF DPM CODE LOCALIZATION DATASET

Since DPM codes are mostly used in industry, images used for DPM code localization are captured by industrial cameras in various complex industrial environments.

Herein, the process is briefly introduced as follows: 1) Candidate images are collected from a variety of different materials, such as cell phone, battery, metal, transparent material, screen etc. 2) To obtain as many images as possible, we expand the number of DPM codes images on each material by adjusting the camera's shooting distance and angle. For example, we fixed the camera at a certain height and adjusted

the roll, pitch, and yaw angles to take multiple shots of DPM codes on different materials, then adjusted the camera height and repeated the process. 3) Meanwhile, in order to evaluate the exposure of DPM code area in experiments, we obtained different exposure DPM code images by adjusting illumination intensity in industrial environments.

B. DPM CODE LOCALIZATION DATASET ANNOTATION

1) DEFINING OBJECT CATEGORIES FOR DPM CODE LOCALIZATION DATASET

In this paper, we not only need to locate the DPM code area, but also evaluate the exposure level of the area. Besides, considering the fact that the DM code is the most commonly used barcode for DPM codes in practice, we classify the DPM code area into three categories, namely DM_Low, DM_Normal, and DM_High, which represent the under-exposed DPM code, the DPM code of normal exposure, and the over-exposed DPM code, respectively, as shown in Figure 9.

FIGURE 9. DPM codes on different exposure. (a) DM_Low (the under-exposed DPM code). (b) DM_Normal (the DPM code of normal exposure). (c) DM_High (the over-exposed DPM code).

2) DPM CODE DATASET ANNOTATION PROCESS

The labeling of the dataset mainly includes two steps: The first step is to set the folder name, picture name and format, and generate the .txt file used for training, testing, and verification; The second step is to use the labelImg tool to draw a bounding box on the target object on the image and mark the corresponding category, and generate .xml files.

In the first step, we create three folders, namely JPEGImages, Annotations, and ImageSets. JPEGImages folder is used to store training and test pictures (image format is .jpg). Annotations folder is used to store files in .xml format, which is the label corresponding to the image. Each .xml file corresponds to a picture in the JPEGImages folder. ImageSets folder contains the main folder, which stores train.txt, val.txt and test.txt files. These files store the image paths used for table training, validation and testing respectively.

In the second step, we firstly pick out all the DPM code areas in the image and draw the bounding box. Meanwhile, every bounding box is required to be as small as possible while including all visible parts of the object instance. Then, we put the corresponding category label on each bounding box. Notice that every DPM code needs to have a bounding box. This is important for training localization algorithms because it decides whether an area contains an object or not. Finally, after tagging the DPM code in the image, we generate the .xml file and store it in the Annotations folder. The file name must also be the same as the current image. The above steps are repeated until all images are tagged.

IV. NETWORK TRAINING AND EXPERIMENTAL ANALYSIS

A. TRAINING DATA

Based on the above DPM code localization dataset, we only use the DPM code image on the cell phone as the training set, and the dataset of other scenes as the test set to verify the generalization ability of DPMLCN. To increase the diversity of the dataset, the DPM code images on the cell phone are processed by means of data augmentation such as rotation, translation, reversal, cutting and so on. Finally, there are 2,500 DPM code images on the battery as a training and validation set, in which the proportion of DM_Low, DM_Normal and DM_High is 1:1:1, and the proportion between the training and validation set is 9:1. That is there are 2250 images for training and 250 images for verification. Meanwhile, 1200 DPM code images in other scenes including battery, metal, transparent material and screen are chosen as the test set in order to verify the generalization ability of the proposed networks.

B. TRAINING

The computer configurations used in this paper include CPU (Inter Core-i7), basic frequency (3.6 GHz), graphics card (TATAN XP), video memory (12 GB), operation system (Ubuntu 16.04). Two frameworks, including Keras and TensorFlow are applied.

To optimize the network, we use the Adam algorithm [26] and set the exponential decay rate of the first moment estimate to be 0.9 and the exponential decay rate of the second moment estimate to be 0.99. Meanwhile, epoch and Batch sizes are set to 625 and 32 respectively. The learning rate is dynamically changed as follows: the initial learning rate is 0.001 and learning rate will be reduced 10 times if the loss function is not declined after training the Batch data 200 times. In order to accelerate the network convergence, the Batch Normalization [27] is performed behind the convolution layer.

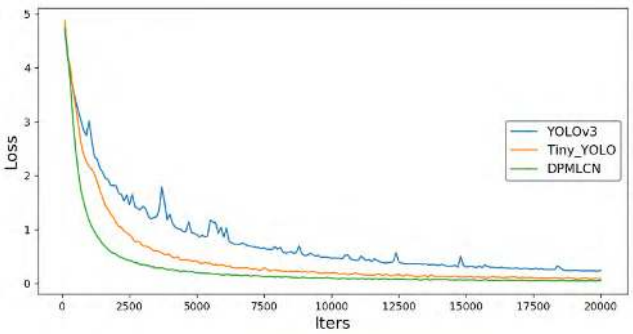


FIGURE 10. Loss function for different networks.

C. LOSS FUNCTION RESULT ANALYSIS

Figure 10 shows the loss function curves for different networks, including YOLOv3, Tiny_YOLO and the proposed DPMLCN. It can be observed that the convergence rate of loss function in the YOLOv3 network is slower than the other two networks because it has more layers. The proposed

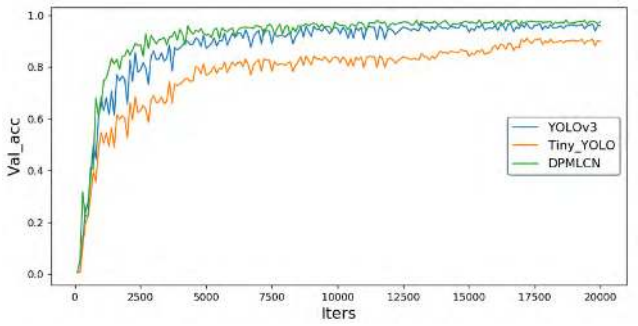


FIGURE 11. Accuracy of the validation datasets for different networks.

DPMLCN has the fastest convergence speed and the smallest loss error, indicating that the DPMLCN has better training results than the other two networks.

Figure 11 presents the accuracy of the validation datasets. It is clear that the Tiny_YOLO network has the lowest accuracy, indicating that the model expression ability of network with few layers is limited. Usually, adding the network layers can improve the model expression ability. Although the DPMLCN has less layers than the YOLOv3 network, their accuracy is comparable. Therefore, it can be concluded that the DPMLCN can improve the accuracy and reduce the calculation simultaneously.

D. PERFORMANCE ANALYSIS

1) PRECISION AND RECALL

In order to evaluate the performance of different networks on DPM code location tasks, the validation datasets in the scene of cell phone are tested. The precision and recall are chosen to evaluate the network performance. The precision denotes the rate of discovery of real positive samples, and the recall denotes the proportion of predicted positive samples that are correctly real positives.

Comparison results of the three networks are presented in Table 2. It is clear that the detection rates of DPMLCN and Tiny_YOLO network are faster than the YOLOv3 network about 4.5 times. With the similar detection rate, the DPMLCN network has a better precision and recall than the Tiny_YOLO network. It means that the proposed network can achieve a good localization precision and accelerate localization speed to the DPM codes.

2) GENERALIZATION ABILITY

To further evaluate the generalization ability of different networks, the test datasets (1200 images) in the complex scenes including battery, metal, transparent material and screen are tested using different networks. The corresponding results with different networks are shown in Table 3.

As shown in Table 3, the localization precision of DPM codes using the DPMLCN network is still higher than 99% although the test scenes are not included in the training scenes and the recall is 99.3%. Obviously, the proposed DPMLCN network has a strong generalization ability, so that it can be well applied to different complex scenes.

According to the above analysis, DPMCLN is superior to the YOLOv3 network in DPM code localization. Moreover, its computational speed is significantly faster than the YOLOv3 network, reaching to more than 90 fps on a Titan X GPU. Furthermore, DPMCLN achieves higher accuracy than the real-time systems Tiny-YOLO. It indicates that the DPMLCN exhibits high localization performance and computational speed in the application of DPM code location.

E. DPM CODE LOCATION RESULTS

1) LOCALIZATION ABILITY COMPARISON

To assess the localization ability of different networks on DPM codes at different exposures levels, we make use of the under-exposed DPM code images, the normal exposed DPM code images, and the over-exposed DPM code images from the untrained DPM code localization dataset. Figure 12 shows the localization results of Tiny_YOLO, YOLOv3 and DPMLCN with different exposure DPM code images. It can be seen that the normal exposed DPM code images can be successfully located by the three networks, but the bounding box of the DPMLCN output is the closest to the real DPM code area. For under-exposed and over-exposed DPM codes, Tiny_YOLO can only locate part of the DPM code area, and YOLOv3 can correctly locate the

TABLE 2. Experimental results of DPM code validation datasets in the cell phone scene.

Model	Network layers	Precision			Recall	Frame frequency (f/s)
		DM_Low	DM_Normal	DM_High		
YOLOv3	53	99.5%	99.9%	99.6%	99.4%	20
Tiny_YOLO	16	93.1%	94.6%	93.2%	91.4%	91
DPMLCN	51	99.8%	99.9%	99.7%	99.5%	94

TABLE 3. Experimental results on the DPM code test dataset in the different scenes.

Model	Network layers	Precision			Recall	Frame frequency (f/s)
		DM_Low	DM_Normal	DM_High		
YOLOv3	53	96.5%	99.7%	98.6%	97.6%	20
Tiny_YOLO	16	91.1%	90.6%	91.2%	89.4%	91
DPMLCN	51	99.7%	99.9%	99.5%	99.3%	93

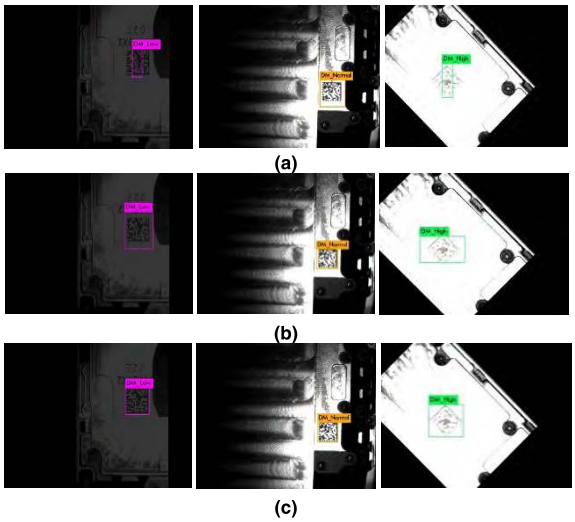


FIGURE 12. Location comparison of the three networks for underexposed, normal exposed and overexposed DPM codes: (a) Tiny_YOLO, (b) YOLOv3, (c) DPMLCN.

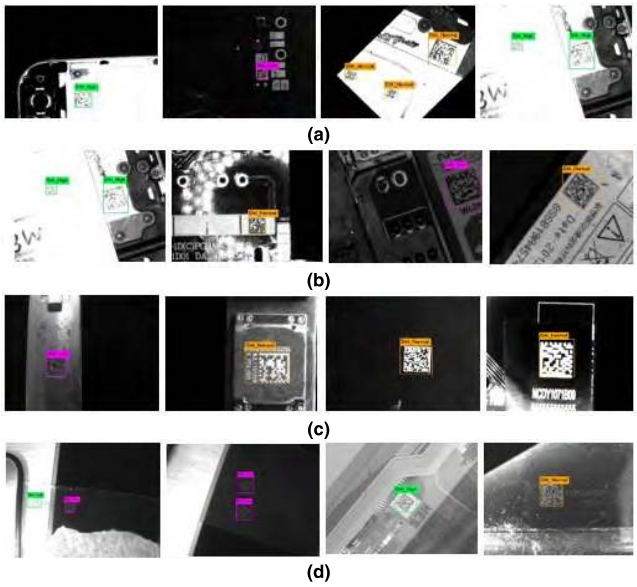


FIGURE 13. Localization results of DPMLCN on different materials: (a) cell phone, (b) battery, (c) metal, (d) transparent material and screen.

DPM code area while the bounding box is a bit larger that the real DPM code area. Comparing with the Tiny_YOLO and YOLOv3 networks, the proposed DPMLCN network can locate the DPM code area well and the boundary box is very close to the actual area.

From the above analysis, it is concluded that the proposed DPMLCN has better localization ability in different exposure DPM codes than the other two networks, and the bounding box can cover the DPM code area exactly. Additionally, Tiny_YOLO and YOLOv3 both have some limitations in underexposed and overexposed DPM codes.

2) LOCATION ABILITY FOR DIFFERENT MATERIALS

The localization effects of DPMLCN in different materials are shown in Figure 13, where the colors of boundary boxes denote the exposure level, which is useful to the subsequent process of DPM codes. Orange, green, and red imply normal, low and over exposure, respectively. As can be seen from Figure 13, DPMLCN has an accurate localization for DPM codes in different scenarios, indicating that DPMLCN is with good localization performance and strong generalization ability.

V. CONCLUSIONS

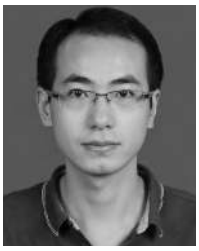
This paper proposes a DPMLCN network based on the depthwise separable convolution and feature pyramid network to realize the DPM code localization precisely and quickly in complex scenes. Benefiting from the improved network structure and multi-scale prediction output, the model expression ability is improved and the calculation speed is enhanced obviously. Experimental results in the DPM code localization datasets demonstrate that the proposed DPMLCN network performs better than the Tiny_YOLO and YOLOv3 networks in terms of localization accuracy and computational efficiency. It can accurately localize the DPM code in different scenes, such as multiple sizes, different materials, uneven illumination, under/over-exposure, polluted, and wear etc. It also has good generalization ability for different materials, and its speed of location reaches to 90fps. In addition, the proposed DPMLCN network can output the exposure level of the DPM code, which is useful to adjust the imaging system to improve quality of the obtained DPM code in practice.

REFERENCES

[1] W. Wei, W. He, L. Lei, G. F. Guo, and J. B. Niu, "Accurate location of polluted data matrix code from multiple views," *J. Comput.-Aided Des. Comput. Graph.*, vol. 25, no. 9, pp. 1345–1353, 2013.
[2] A. K. Jain and Y. Chen, "Bar code localization using texture analysis," in *Proc. 2nd Int. Conf. Document Anal. Recognit. (ICDAR)*, Oct. 1993, pp. 41–44.
[3] Y.-F. Hao, F.-H. Qi, and R.-J. Jiang, "A novel machine learning based algorithm to detect data matrix," *J. Image Graph.*, vol. 12, no. 10, pp. 1873–1876, 2007.
[4] R. Jin, S. Zhao, X. Xu, E. Song, and C.-C. Hung, "Super-resolving barcode images with an edge-preserving variational Bayesian framework," *J. Electron. Imag.*, vol. 25, no. 3, 2016, Art. no. 033016.
[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Proces. Syst.*, Dec. 2012, pp. 1097–1105.
[6] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556
[7] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.
[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[9] T. Y. Lin, P. Goyal, R. Girshick, K. M. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published. doi: 10.1109/TPAMI.2018.2858826.
[10] W. Liu *et al.* (2016). "SSD: Single shot multibox detector." [Online]. Available: https://arxiv.org/abs/1512.02325
[11] J. Redmon and A. Farhadi. (2016). "YOLO9000: Better, faster, stronger." [Online]. Available: https://arxiv.org/abs/1612.08242
[12] J. Redmon and A. Farhadi. (2018). "Yolov3: An incremental improvement." [Online]. Available: https://arxiv.org/abs/1804.02767
[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[14] K. M. He, G. Gkioxari, P. Dolar, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2017, pp. 2980–2988.
[15] T. Grósz, P. Bodnár, L. Tóth, and L. G. Nyúl, "QR code localization using deep neural networks," in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process. (MLSP)*, Nov. 2014, pp. 1–6.
[16] P. Bodnár, T. Grósz, L. Tóth, and L. G. Nyúl, "Efficient visual code localization with neural networks," *Pattern Anal. Appl.*, vol. 21, no. 1, pp. 249–260, 2018.
[17] L. Belussi and N. Hirata, "Fast QR code detection in arbitrarily acquired images," in *Proc. 24th SIBGRAPI Conf. Graph., Patterns Images*, Aug. 2011, pp. 281–288.
[18] T.-H. Chou, C.-S. Ho, and Y.-F. Kuo, "QR code detection using convolutional neural networks," in *Proc. Int. Conf. Adv. Robot. Intell. Syst.*, May 2015, pp. 1–5.
[19] L. Sifre and S. Mallat. (2014). "Rigid-motion scattering for texture classification." [Online]. Available: https://arxiv.org/abs/1403.1687
[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*, Jun. 2016, pp. 779–788.
[21] A. G. Howard *et al.* "MobileNets: Efficient convolutional neural networks for mobile vision applications." [Online]. Available: https://arxiv.org/abs/1704.04861?context=cs
[22] T. Y. Lin, P. Dollár, R. Girshick, K. M. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 936–944.
[23] M. Sandler, A. Howard, M. L. Zhu, A. Zhmoginov, and L. C. Chen. "MobileNetV2: Inverted residuals and linear bottlenecks." [Online]. Available: https://arxiv.org/abs/1801.04381v3
[24] J. Hosang, R. Benenson, and B. Schiele, "A convnet for non-maximum suppression," in *Proc. 38th German Conf. Pattern Recognit.*, Sep. 2016, pp. 192–204.
[25] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS—Improving object detection with one line of code," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2017, pp. 5561–5569.
[26] D. P. Kingma and J. Ba. (2014). "Adam: A method for stochastic optimization." [Online]. Available: https://arxiv.org/abs/1412.6980
[27] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2015, pp. 448–456.

YUSHENG LI was born in Wuhan, Hubei, China, in 1992. He is currently pursuing the master's degree with Shenzhen University, Shenzhen, China.

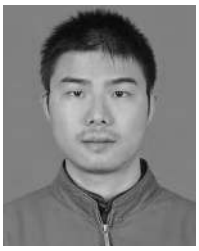


YONG TIAN received the B.S. degree in automation and the Ph.D. degree in control Theory and control engineering from the College of Automation, Chongqing University, Chongqing, China, in 2008 and 2012, respectively.

From 2013 to 2015, he was a Postdoctoral Researcher with the Department of Mechanical Engineering, Tsinghua University, China. He is currently an Assistant Professor with the College of Physics and Optoelectronic Engineering, Shenzhen University. His research interests include artificial neural networks, image process, and automation control.

JINDONG TIAN received the B.S. degree in precision machinery, the M.S. degree in optical instrument, and the Ph.D. degree in optical engineering from Tianjin University, Tianjin, China, in 1995, 1998, and 2001, respectively.

From 2001 to 2002, he was a Postdoctoral Researcher with the Department of Mechanical Engineering, The Hong Kong University of Science and Technology, China. He is currently a Professor with the College of Physics and Optoelectronic Engineering, Shenzhen University. His research interests include deep learning, optical metrology, and machine vision.



FEI ZHOU received the B.Eng. degree in electronics and information engineering from the Huazhong University of Science and Technology, in 2007, and the Ph.D. degree in electronic engineering from Tsinghua University, in 2013.

From 2013 to 2016, he was a Postdoctoral Fellow with the Graduate School at Shenzhen, Tsinghua University. From 2017 to 2018, he was a Visiting Scholar with the Department of Statistical Science, University College London. He is currently an Assistant Professor with the College of Information Engineering, Shenzhen University. His research interests include image super-resolution, image interpolation, and image quality assessment.