Measurement 172 (2021) 108885

Contents lists available at ScienceDirect

Measurement

journal homepage: www.elsevier.com/locate/measurement

EDDs: A series of Efficient Defect Detectors for fabric quality inspection

Tong Zhou [a,c,*,1], Jiabin Zhang [b,c,*,1], Hu Su [b,**], Wei Zou [b], Bohao Zhang [b]

[a] Institute of High Energy Physics, Chinese Academy of Sciences, Beijing 100049, China
[b] Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[c] University of Chinese Academy of Sciences, Beijing 100049, China

ARTICLE INFO

Keywords:
Defect detection
Convolutional neural network
Fabric quality inspection
Feature fusion

ABSTRACT

Deep Convolutional Neural Network (DCNN) has recently advanced state-of-the-art performance on vision-related tasks and its application is further extended to industrial fields. The paper focuses on the problem of fabric defect detection to which an efficient DCNN architecture is developed. In contrast to previous methods that directly apply existing DCNN models demonstrated on natural images to industrial images, the proposed Efficient Defect Detectors (EDDs) are sufficiently optimized with consideration of the characteristics of fabric surface images, i.e., resolution, defect appearance, etc. Firstly, lightweight backbone is suggested in EDD to improve computational efficiency without reduction in image resolution. Secondly, a new feature fusion strategy named L-shaped feature pyramid network (L-FPN) is proposed and utilized to make full use of low-level texture features which are demonstrated to be more important than high-level semantic features in defect recognition. Based on the configurations of lightweight backbone and L-FPN, we use only one hyper-parameter to jointly adjust the proportion of resources occupied by width, depth and input resolution so that a family of defect detectors under different resource constraints can be developed. Experiments are conducted on a large fabric dataset to demonstrated the effectiveness of EDDs. Compared with the recent state-of-the-art detector, EfficientDet-d3, EDD-d3 achieves higher mean Average Precision (mAP) (20.9 vs 19.9) but with fewer parameters. EDD-d3 has 8.59M parameters and 31.78B FLOPs (floating point operation per second), which respectively are 39.8% and 49.0% lower than EfficientDet-d3. The proposed EDDs achieve better trade-off between accuracy and speed than previous methods. EDDs could be applied to fabric production sits with different resource restrictions, which demonstrates that EDDs have important application value.

1. Introduction

In the textile and apparel industry, surface defect is one of the most important factors influencing the quality of fabric, and defect detection is a core link of quality management [1]. In the early days, fabric defect detection depends on manual subjective discrimination, which not only leads to high labor costs, but also lacks consistency and reliability [2]. With the development of modern industry, speed and accuracy of manual inspection can no longer meet the demand.

Vision-based automatic inspection provides an efficient way to solve the problem. In the inspection, product is inspected under standard procedures, overcoming subjectivity and capriciousness of human. Moreover, secondary injury of the products is prevented through non-contact inspection manner [3]. Researches on this field have been carried out for a long time, and significant progress is made. According to previous publications, existing methods can be roughly categorized into two classes [4]: texture analysis-based and deep learning-based methods.

Texture analysis-based method is to distinguish defective images and normal images through the analysis of texture characteristics, such as tropism and homogeneity. This type of methods started early, dating back to the 1980s [5], and can be divided into three classes [6]: statistical [7,8], spectral [9,10] and model-based [11,12] methods. The statistical methods analyze texture and recognize defects based on the statistical distribution characteristics of gray-scale in product images, and this type of methods are effective especially for stochastic textures, such as ceramic tiles [7], castings [13], and wood [14]. The spectral methods are based on the assumption that defects destroy the structural consistency of uniform textures, and accurate defect detection could be accomplished according to response difference between normal texture and defect [9]. The spectral methods are applicable to repeated or regular texture to detect the defects which are difficult to be identified only by gray-scale feature. The model-based methods analyze texture attributes, establish texture image representation, and

* The first two authors contributed equally.
** Corresponding author.
E-mail address: hu.su@ia.ac.cn (H. Su).
[1] Co-first authors.

https://doi.org/10.1016/j.measurement.2020.108885
Received 13 August 2020; Received in revised form 22 October 2020; Accepted 10 December 2020
Available online 28 December 2020

then detect defects by identifying abnormal textures. The models used in surface defect detection include auto-regressive models [11], Markov Random Field (MRF) [12] and Texture Exemplar (TEXEM) [15]. These traditional fabric inspection techniques all aim at explicitly constructing templates or features for images. They require manual designed features and careful parameter adjustment. Since there is no explicit guideline for choosing optimal representations, human experience plays an important role in these technologies, leading to low efficiency and poor performance.

In recent years, deep learning technology has achieved the best performance on many visual tasks with automatic feature extraction, avoiding the difficulty of manually designing features. The application of DCNN is further extended to industrial fields. In order to obtain different types of output, the defect detection network will be quite different in structure, which can be divided into three classes [16]: classification-, detection-, and segmentation-based methods. The classification-based method obtains class label of image, which can identify whether the image is defective. And in some multi-classification tasks, it is also necessary to identify the type of defect [17,18]. The detection-based method not only needs to determine whether the current image is defective and the defect category, but also needs to determine the location and the size of defect [19–21]. The segmentation-based method needs to determine whether each pixel belongs to the defect target, so as to judge the quality of the product [22,23]. However, the existing deep learning-based methods focus on directly applying the detection network verified in natural images to industrial images. Due to the large difference between natural and industrial images in terms of resolution and target appearance, speed and accuracy of the methods could not well meet industrial requirements. As a result, the systems is unpractical in industrial production.

Focusing on the problem of fabric defect detection and to overcome the mentioned shortcomings, we have developed a new family of efficient defect detectors, EDDs, which consistently achieve better accuracy with much fewer parameters and FLOPs than previous methods. This series of lightweight detectors can make better use of high resolution image and low-level information, in which way the difference between the two types of images are well addressed and the computational efficiency is improved. Moreover, compound-scaling strategy is introduced to jointly adjust the proportion of resources occupied by width, depth and resolution. EDDs could be applied to fabric production sits with different resource restrictions to achieve better trade-off between accuracy and speed. Fig. 1 shows the performance comparison on fabric defect dataset. The contribution of the paper could be summarized as

- The difference between natural image and fabric surface image are analyzed based on which lightweight backbone is suggested and a new feature fusion strategy, L-FPN, is proposed to pay more attention to low-level features.
- The R-Compound Scaling (Resolution-Compound Scaling) strategy is utilized to jointly balance related factors, including image resolution, the depth and the width of DCNN. Accordingly, a family of defect detectors under different resource constraints are developed.
- The proposed EDDs can achieve better trade-off between speed and accuracy than previous methods which is of significant importance in industrial applications.

The remainder of this paper is organized as follows. Section 2 introduces the related work of this paper. Next, Section 3 describes the framework of our proposed fabric defect detector. Section 4 carries out extensive experiments, where the experimental results and related analysis are provided. Finally, Section 5 gives the conclusion of this paper.

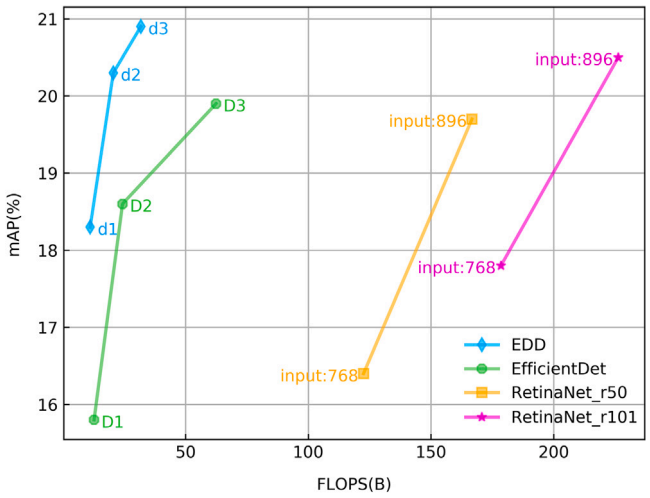


Fig. 1. Comparison of different one-stage models. The horizontal axis represents FLOPs and the vertical axis represents accuracy. The increase in input resolution improves the accuracy of each model. Among them, the model closer to the upper left corner has higher efficiency. One can find that EDDs have higher accuracy and fewer parameters.

2. Related work

2.1. Defect detection

In recent years, more and more DCNN-based methods [17–23] are proposed to perform surface inspection tasks without the need of hand-crafting a set of features like in traditional ways. As pointed out in [16], the three types of DCNN-based methods, i.e., classification-, detection-, and segmentation-based methods, have different forms of outputs and accordingly, different human efforts involved in image labeling are required in the methods. Bounding box and image tag annotations are more economical than segmentation masks. The workload of labeling segmentation masks is more than 15 times heavier than that of spotting object locations [24]. The classification-based methods could not provide size and location information of the defect which, however, is important in the fabric quality judgment. Taken together, detection strategy is adopted in the paper. Note that classification-based methods are closely related to detection-based ones. To pursue a clarity and complete description on previous works, both classification-based and detection-based methods are introduced.

A significant number of methods [17,18,25–28] accomplish the task by classifying normal and defect images. For example, MSPyrPool [17] is proposed to solve the steel defect classification problem on arbitrarily sized images. The network can be seen as a fully supervised hierarchical bag-of-features extension that is trained online and can be fine-tuned for any given task. Wang et al. [18] design a joint detection CNN architecture that contains two major parts: the global frame classification part and the sub-frame detection part. The former learns to classify the whole image, and the later is implemented on the image patches generated by the sliding-window method. Ren et al. [25] propose a generic DCNN-based surface inspection approach. There are two phases in the proposed method. The first phase includes supervised training of patch classifier. In the second phase, the trained classifier is used to extract patches, and then the heatmap of the whole image is generated to predict the locations of defects. Based on the image partitioning operation, these two methods achieve defect localization roughly by using classification networks.

Benefited from the great success of object detection algorithms applied in natural scenarios, Cha et al. [19] utilize Faster Region-based Convolutional Neural Network (Faster R-CNN) [29] to detect multiple types of damages accurately. Chen et al. [20] propose a cascade network to localize defects in a coarse-to-fine manner. The network

includes two detectors to sequentially localize the cantilever joints and their fasteners and a classifier to diagnose the defects. Zhang et al. [21] propose a method of automatic positioning and classification of yarn-dyed fabric defects based on YOLOv2 (You Only Look Once) [30]. From the above researches, one can find that existing methods focus primarily on directly applying the DCNN models demonstrated on natural images to industrial images and the large difference between the two types of images are not considered. Usually, speed and accuracy of the methods could not well meet industrial requirements.

2.2. Object detection

Object detection is one of the basic tasks in computer vision. At present, deep-learning methods are mainly divided into two categories: two-stage object detection algorithm and one-stage object detection algorithm. Two-stage object detection algorithm, such as the series of region based CNN detectors [31–33], follows the pipeline that first generates a series of candidate boxes as proposals, then classifies and regresses these proposals through CNN. One-stage object detection algorithm does not need to generate candidate boxes, but directly converts the b-box positioning problem into a regression problem. RetinaNet [34] is one of the most commonly used one-stage detector, which is designed to verify the effectiveness of Focal Loss. It is essentially a simple structure composed of ResNet, Feature Pyramid Networks (FPN) and two Fully Convolutional Networks (FCN) sub-networks, but achieves competitive performance with two-stage ones. The family of YOLO [30,35,36] are typical one-stage detectors, which can predict all classes and bounding boxes in a image at the same time. In addition, Single Shot Multi-Box Detector (SSD) [37] uses one-stage idea to improve the detection speed, and generates different scale predictions for different scale feature maps, which significantly improves accuracy. These two types of methods are obviously different in performance. Two-stage networks are superior in detection accuracy, while one-stage networks have higher speed instead. Recently, Google Brain team has systematically studied a variety of object detector architectures and proposed several key optimizations that can improve model efficiency: weighted bi-directional feature pyramid network (BiFPN); a new compound scaling method. Based on these optimizations, researchers develop a series of new object detectors, EfficientDets [38]. Under extensive resource constraints, this type of models still have obvious advantages over previous optimal models. These two key optimizations will be introduced in detail in the following:

Feature Network: Generally, FPN [39] is a typical multi-scale feature fusion structure. Since then, researchers also tried various different feature fusion methods, such as Path Aggregation Network (PANet) [40] with bottom-up and top-down structure, a single-shot object detector based on multi-level feature pyramid (M2Det) [41] with skip-connection strategy. In EfficientDets, the node with only one input edge was removed to simplify PANet and skip-connection was also applied to fuse more features. In the previous pyramid-like module, bilinear interpolation sampling is often used to fuse different scale features. The author believes that it is unfair to add different scale features directly. Considering that their final contribution to detection performance should be different, a common idea is to introduce weight parameters ω_i to automatically learn the importance of different scale features. Thus, a simple and efficient feature fusion basic structure, BiFPN, is realized. In this paper, we hope to obtain a new feature network structure suitable for textile industry scenarios to fuse different scale features efficiently.

Compound Scaling: In EfficientNets [42], authors re-examined several dimensions of previous model scaling strategy to balance both speed and accuracy. The previous model scaling strategy mostly enlarge one dimension to achieve higher accuracy. For example, ResNet [43] can obtain higher accuracy by increasing depth of the network (e.g., ResNet-50 to ResNet-101). EfficientNet backbone jumps out of the previous understanding of model scaling, thinks that depth, width and resolution affect each other. Finally, a new strategy of scaling up all dimensions of parameters is proposed, called compound scaling. This strategy uses a coefficient ϕ to determine the proportion of resources occupied by width, depth and resolution. Based on this, EfficientDets further expand the compound scaling strategy. In feature network, the number of BiFPN channels and repeated layers can also be controlled. In addition, the number of layers in box/class network and the resolution of input images are also parts of compound scaling strategy. In the industrial scenario of this paper, combined with the characteristics of large resolution, rich texture information and simple semantic information, a new joint adjustment strategy is proposed to improve the effectiveness of fabric defect detection.

3. Proposed method

To inspect fabric quality efficiently, we propose a series of lightweight EDDs. An overview of our framework is illustrated in Fig. 2. Specifically, the framework is divided into the following parts: the backbone in which the lightweight EfficientNets are selected; L-FPN to efficiently fuse multi-scale features; a structure similar to RetinaNet to classification and regression of bounding boxes. The above components can be adjusted by the proposed R-Compound Scaling strategy to implement a series of detectors under different resource constrains. In this section, more details for each part of our method will be described individually.

3.1. Backbone

In this paper, the recent successful classification network EfficientNets are selected as the backbone network in EDDs. This series of backbones mainly have the following attractive advantages:

- The state-of-the-art EfficientNet backbone achieves better performance visibly than previous backbones with the same parameters and FLOPs, and fully considers more optimization metrics.
- EfficientNet backbone uses a simple and efficient composite coefficient to uniformly scale the depth, width and resolution of the network, so that the fabric defect detectors can adapt to different resource constraints.
- By adopting EfficientNet backbone, the ImageNet-pretrained checkpoints could be easily used. On the basis, the training time is reduced and computational efficiency is improved.

In this paper, we choose EfficientNet-{b0, b1, b2} as our backbone networks. The detailed structures of these three networks are shown in Table 1. Their main building block is multiple mobile inverted bottlenecks MBConv [44,45] with different specifications.

3.2. Feature network

Multi-scale feature fusion aims to combine features with different resolutions, so that the network has a competitive ability to represent semantic information and texture details.

Previous excellent feature fusion methods, such as FPN, has only one top-down unidirectional information flow. And improvements based on FPN, such as PANet, BiFPN [38], etc., provide bidirectional information flow to fuse low-level texture information and high-level semantic information unbiasedly. All of them fuse features belonging to different layers at equal times. Therefore, these kinds of feature fusion methods are called unbiased feature fusion. Among them, PANet will improve detection accuracy with the cost of more parameters and computations. In order to utilize more low-level features while reducing the proportion of high-level features, feature network needs to have the ability to fuse features in a biased manner. Inspired by the idea

Table 1
Architecture of Backbone Networks.

Model	EfficientNet-b0	EfficientNet-b1	EfficientNet-b2
Conv		3 × 3, stride 2	
Stage 1	[MBConv1, $k3\times3$]×1	[MBConv1, $k3\times3$]×2	[MBConv1, $k3\times3$]×2
Stage 2	[MBConv6, $k3\times3$]×2	[MBConv6, $k3\times3$]×3	[MBConv6, $k3\times3$]×3
Stage 3	[MBConv6, $k5\times5$]×2	[MBConv6, $k5\times5$]×3	[MBConv6, $k5\times5$]×3
Stage 4	[MBConv6, $k3\times3$]×3	[MBConv6, $k3\times3$]×4	[MBConv6, $k3\times3$]×4
Stage 5	[MBConv6, $k5\times5$]×3	[MBConv6, $k5\times5$]×4	[MBConv6, $k5\times5$]×4
Stage 6	[MBConv6, $k5\times5$]×4	[MBConv6, $k5\times5$]×5	[MBConv6, $k5\times5$]×5
Stage 7	[MBConv6, $k3\times3$]×1	[MBConv6, $k3\times3$]×2	[MBConv6, $k3\times3$]×2
Output		Conv1 × 1, Pooling, FC	

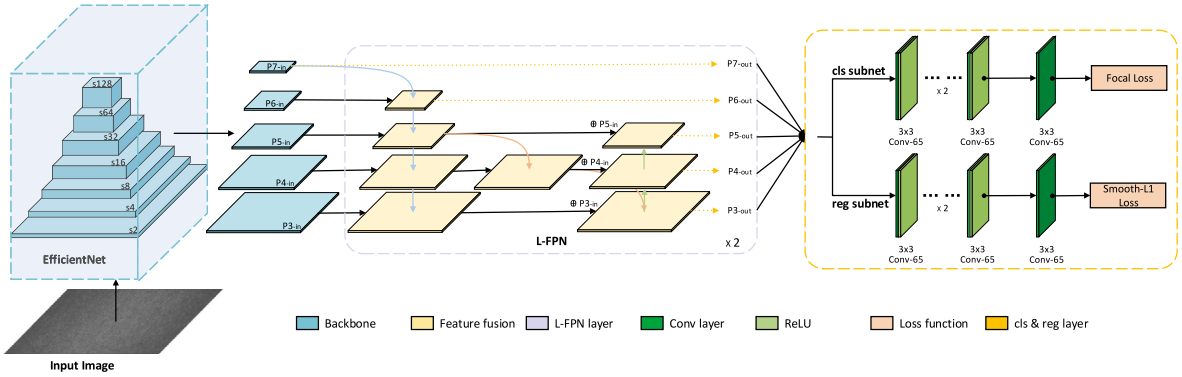
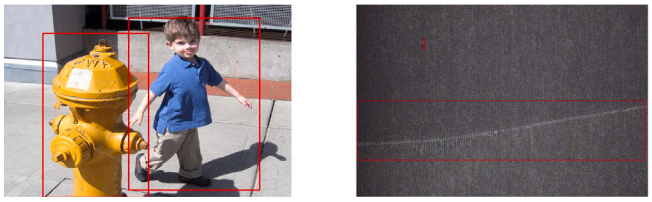


Fig. 2. EDDs Architecture. Features are extracted from each stage of backbone network EfficientNets, and then use our L-FPN to fuse them into multi-level features biasedly. Finally, two FCN sub-networks are used for each feature map to implement bounding box classification and position regression tasks independently.



(a) Image in COCO 2017 (b) Image in fabric dataset

Fig. 3. Comparison of COCO 2017 and Fabric Dataset. (a) is an image in COCO 2017, with rich semantic information and background information. (b) is a fabric image with rich texture information.

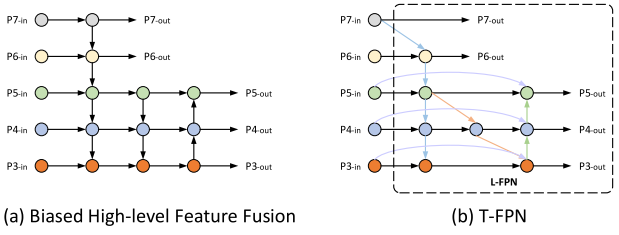


Fig. 4. L-FPN Architecture Design. (a) is biased low-level feature fusion architecture. We use top-down and bottom-up structures to fuse low-level texture information and high-level semantic information biasedly. (b) is our L-FPN architecture, which uses less computing resources, but has higher efficiency.

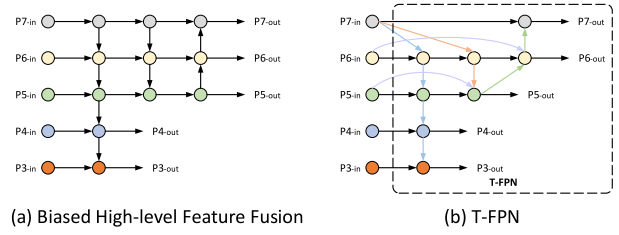


Fig. 5. T-FPN Architecture Design. (a) is biased high-level feature fusion architecture. We increase the fusion times of high-level features to obtain more semantic information. (b) is our T-FPN architecture.

of BiFPN, we propose a structure for biased feature fusion, L-FPN. The design ideas and structural advantages of the proposed structure will be introduced in detail in the following.

Based on the observation of defects in the fabric images, the process of feature learning is obviously different from that in the natural scene. As shown in Fig. 3, considering that the images in textile industry scenario have rich texture information and simple semantic information, an important property of our method is that it is biased. In order to pay more attention to the utilization of low-level features, we proposed L-FPN. To verify the importance of low-level features in defect recognition, a counterpart of L-FPN, T-shaped feature pyramid network (T-FPN), is proposed as well and the comparison between them are conducted. The output feature maps {P3, P4, P5, P6, P7} are selected as the input of L-FPN, which from the last 5 blocks (level 3–7) of EfficientNets. In order to obtain the input of the final box/class network, the multiple L-FPN structures are applied for feature fusion. Inspired by the idea of bidirectional information flow, we add an additional top-down information flow in FPN to increase the fusion times of low-level features {P3-in, P4-in, P5-in}. At the same time, the proportion of high-level features {P6-in, P7-in} is appropriately reduced, so that our L-FPN can fuse features in a biased manner, as shown in Fig. 4(a). In contrast, T-FPN focuses more on increasing the fusion times of high-level features rather than low-level features, as shown in Fig. 5(a). Section 4.4.1 compares the contributions of L-FPN and T-FPN to the fabric defect detector. It is also demonstrated that L-FPN is more suitable for this specific industrial scenario, that is, the biased feature fusion has obvious advantages.

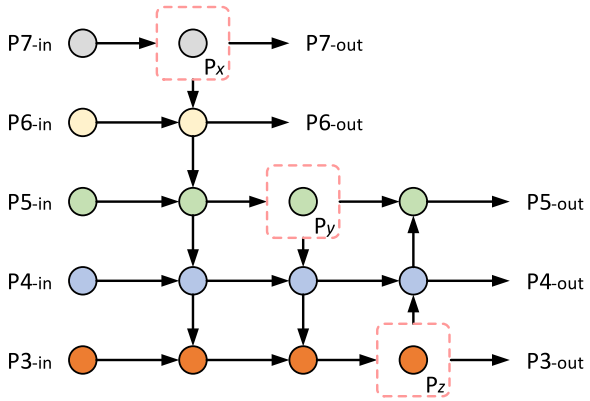


Fig. 6. Simplified Biased Low-level Feature Fusion Architecture. We simplify biased low-level feature fusion architecture by removing nodes with only one input edge.

Additionally, two enhancements are used to improve detection efficiency. Firstly, in order to solve this limitation of the large cost of parameters and computations, the nodes that have less contribution to feature fusion network are removed, that is, discard the nodes with only one input, such as P_x, P_y and P_z in Fig. 6. Secondly, to improve accuracy, the feature fusion times are increased. Short-cut and block-repeat are used in our feature network, and the final structure of L-FPN is shown in Fig. 4(b). Generally, given a series of multi-scale features $\{P3_{in}, P4_{in}, P5_{in}, P6_{in}, P7_{in}\}$, the corresponding outputs after L-FPN are as follows:

$$P7_{out} = P7_{in}^{(0)} \tag{1}$$

$$P6_{out} = Conv\left(\frac{w_{61} \cdot P6_{in}^{(0)} + w_{62} \cdot P7_{in}^{(0)}}{w_{61} + w_{62}}\right) \tag{2}$$

$$P5_{out} = Conv\left(\frac{w_{51} \cdot P5_{in}^{(0)} + w_{52} \cdot P5_{in}^{(1)} + w_{53} \cdot P4_{out}}{w_{51} + w_{52} + w_{53}}\right) \tag{3}$$

$$P4_{out} = Conv\left(\frac{w_{41} \cdot P4_{in}^{(0)} + w_{42} \cdot P4_{in}^{(2)} + w_{43} \cdot P3_{out}}{w_{41} + w_{42} + w_{43}}\right) \tag{4}$$

$$P3_{out} = Conv\left(\frac{w_{31} \cdot P3_{in}^{(0)} + w_{32} \cdot P3_{in}^{(1)} + w_{33} \cdot P4_{in}^{(2)}}{w_{31} + w_{32} + w_{33}}\right) \tag{5}$$

where w_{ij} is the weight of the jth input of the output node at level i. The left sides of Eqs. (1)–(5) denote the output feature at the corresponding level and the variables in the right sides denote the input features of the nodes. For example, $P1_{out}$ is the output feature at level 1, and $P1_{in}^{(0)}$, $P1_{in}^{(1)}$ and $P1_{in}^{(2)}$ respectively denote the input features of the first, the second and the third fusion nodes at level 1. Based on the definition, the variables P_x, P_y and P_z in Fig. 6 can be denoted as $P7_{in}^{(0)}$, $P5_{in}^{(1)}$, $P3_{in}^{(2)}$, respectively. Other variables have similar meanings.

Following L-FPN, a similar structure to RetinaNet which uses two FCN sub-networks (they have the same structure but do not share parameters) for each feature map is constructed to implement bounding box classification and position regression tasks independently.

3.3. R-compound scaling

In the textile industry scenario, fabric images usually have the characteristics of large resolution, rich texture information, and simple semantic information. To adjust the depth, width and resolution efficiently so that the defect detector can adapt to different resource constraints, how to implement a compound scaling strategy for industrial scenarios is one of the biggest challenges.

Some recent works, using compound scaling strategies, have competitive performance on computer vision tasks. Inspired by the compound scaling strategy in EfficientNets and EfficientDets, a new series of EDDs for large-scale inputs are proposed. Based on the baseline EDD-d0, we have developed a new strategy of scaling up all dimensions of parameters, called R-Compound Scaling, which consistently achieve much better accuracy than prior technique. Different with some common object detection models with small-scale inputs, such as RetinaNet [34] and YOLO [35], our detectors need to adapt to larger-scale fabric images. In the case of limited resources, the most common idea is to sacrifice width and depth of the network to increase the proportion of resolution. With the configuration of backbone, $EfficientNet-b(\eta-1)$, we got the following equations:

Input image resolution:

$$R_{input} = 640 + 128 \cdot \eta \tag{6}$$

Feature network:

$$W_{l-fpn} = 65 \cdot (3/e)^{\eta-1} \tag{7}$$

$$D_{l-fpn} = 2 + \lfloor \eta/2 \rfloor \tag{8}$$

Class/box network:

$$D_{class} = 2 + \lfloor \eta/2 \rfloor \tag{9}$$

where η denotes a hyper-parameter that controls how much computing resources are allocated to each part of backbone, feature network, box/class network. Due to the constraint of parameter η, EDDs can reasonably and efficiently allocate resources to large-scale inputs biasedly, which makes the detectors have a competitive effect on fabric images with higher resolution. In this paper, the specific parameters of EDD-{d0, d1, d2} are shown in Table 2.

3.4. Lightweight backbone

Generally, deep backbones are more conducive to mining high-level semantic information. However, fabric images have more regular background information and intuitively, texture information plays a more important role in defect detection than semantic information.

In order to improve the efficiency of detector and reduce the cost of consumption, EDDs use a series of lightweight backbones. Specifically, compared with EfficientDet-{d1, d2, d3} at the same levels, EDDs use lower-level backbones EfficientNet-{b0, b1, b2}. Section 4.4.3 proves that sacrificing the width and depth of backbone does not significantly reduce accuracy, but can save computing resources and improve model efficiency.

4. Experiments

4.1. Dataset and metric

We use the fabric defect dataset provided by Tianchi Academic Competitions held by Alibaba Cloud in 2019.[2] The official provided 9576 images (2446 × 1000) for training, including 5913 defect images and 3663 normal images. Conventionally, this dataset is divided into training set and test set, which contain 4730 images and 1183 images respectively. The defects commonly arose in textile production are found in these images, which are divided into 20 categories. Note that one image may contain more than one categories of defects. Compared with natural real-world images, accurate object detection for these fabric images is challenging due to the following key points.

- Uneven distribution of object categories.
- The large scale difference of objects, some targets are too small and slender in the original image.
- Annotated bounding boxes usually include a lot of background information due to the morphologies of defects.

[2] https://tianchi.aliyun.com/competition/entrance/231748/introduction

Table 2
R-Compound Scaling Strategy.

Model	Input resolution R_{input}	Backbone network	Feature network W_{l-fpn}	Feature network D_{l-fpn}	Box/Class network D_{class}
EDD-d1	768	EfficientNet-b0	65	2	2
EDD-d2	896	EfficientNet-b1	70	3	3
EDD-d3	1024	EfficientNet-b2	80	3	3

Table 3
Comparison With Other State-of-the-art Methods.

Model	mAP	Params	Ratio	FLOPs	Ratio	FPS	Speedup
EDD-d1(768)	**18.3**	**4.55M**	1×	**11.12B**	1×	**32.3**	1×
EfficientDet-d1(640)	15.8	7.63M	1.7×	12.71B	1.1×	33.4	–
RetinaNet-R50(768)	16.4	36.5M	8.0×	122.5B	11.0×	29.3	0.91×
RetinaNet-R101(768)	17.8	56.5M	12.4×	178.5B	16.0×	27.0	0.83×
EDD-d2(896)	**20.3**	**7.24M**	1×	**20.57B**	1×	**28.0**	1×
EfficientDet-d2(768)	18.6	9.38M	1.3×	24.31B	1.2×	29.3	–
RetinaNet-R50(896)	19.7	36.5M	5.0×	166.7B	8.1×	27.0	0.96×
RetinaNet-R101(896)	20.5	56.5M	7.7×	226.3B	11.0×	23.1	0.82×
EDD-d3(1024)	**20.9**	**8.59M**	1×	**31.78B**	1×	**24.3**	1×
EfficientDet-d3(896)	19.9	14.28M	1.7×	62.4B	2.0×	26.0	–

Following experiments prove that these issues are well addressed by the proposed EDDs and improved performance can be achieved.

The metric mAP is used to evaluate the defect detection results. Specifically, the area under the Precision-Recall (PR) curve is called Average Precision (AP). In COCO evaluation, the IoU threshold ranges from 0.5 to 0.95 with a step size of 0.05. We calculate mAP according to the standard process.[3]

4.2. Implementation details

The performance of EDDs is evaluated on a GeForce RTX 2080 Ti GPU. All pre-trained models we used in experiments are publicly available. We directly use the same hyper-parameters with RetinaNet: Stochastic Gradient Descent (SGD) optimizer is used with a weight decay of 4e-5 and a momentum of 0.9. Learning rate is linearly increased from 0 to 0.005 in the first training epoch and then decay 1/10 at 35 epochs and 45 epochs. Focal Loss with $\alpha = 0.25$, $\gamma = 1.5$ is used, and at each pyramid level we use anchors at nine aspect ratios {1:50, 1:20, 1:10, 1:2, 1:1, 2:1, 10:1, 20:1, 50:1}.

4.3. Comparison with the state-of-the-art methods

Table 3 compares the performance of the proposed EDDs with the state-of-the-art methods. And we can get the following conclusions:

- EDD-d3 achieves the top accuracy, 20.9 mAP. And when comparing with the same level detector, EfficientDet-d3, EDD-d3 achieves a significant improvement in accuracy but with reduced parameters and FLOPs (reduced by 39.8% and 49.0%, respectively).
- The accuracy of EDD-d2 is similar to that of RetinaNet-R101, but its parameters are reduced by 87.0% and the FLOPs are reduced by 90.9%.
- For every level of EDDs and EfficientDets, EDDs achieve higher accuracy with fewer parameters. Therefore, EDDs make better trade-off between speed and accuracy than EfficientDets.

From the above analysis we can conclude that, EDDs can achieve preferable accuracy at lower cost of computing resources when comparing with other the state-of-the-art methods. From the application point of view, our proposed methods are superiority over existing methods.

[3] https://github.com/cocodataset/cocoapi

Table 4
Comparison Among Different Feature Networks.

Model	mAP	Parameters	FLOPs
EfficientNet-b1 + FPN	18.5	8.61M	25.19B
EfficientNet-b1 + BiFPN	19.4	7.24M	20.54B
EfficientNet-b1 + L-FPN	20.3	7.24M	20.57B
EfficientNet-b1 + T-FPN	19.3	7.24M	20.53B

4.4. Ablation studies

In this section, we conduct a series of ablation experiments of our proposed EDDs on fabric images in textile industry scenario to prove that L-FPN structure, R-Compound Scaling, and lightweight backbone can bring performance improvement to the fabric defect detector significantly.

4.4.1. L-FPN

EDD-d2 is selected as the framework to compare different feature networks. When the resolution of input image is set to 896 × 896, EDD-d2 is equipped with FPN, BiFPN, L-FPN and T-FPN, respectively. For a fair comparison, each of the networks uses the same backbone and the same box/class network. Accordingly, the resolution of the input image and the strategies of training and testing are all the same (multi-scale training with a ratio from 0.5 to 2.0 randomly). What is more, there is only one bottom-up unidirectional information flow in FPN. Therefore, we repeat FPN $2 \times D_{l-fpn}$ times to ensure that the feature network formed by multiple FPNs has a similar structure to the feature network in our EDDs. The performances are provided in Table 4.

From Table 4, one can find that L-FPN, T-FPN and BiFPN have fewer parameters but achieves higher accuracy compared with FPN. The accuracy of T-FPN is competitive with that of BiFPN. L-FPN achieves the top accuracy. For the mAP metric, it is raised by 1.8, 1.0 and 0.9 when comparing with FPN, T-FPN and BiFPN, respectively. Among the feature networks, FPN and BiFPN are unbiased structures while T-FPN and L-FPN are biased structures. Actually, the short-cut strategy employed in BiFPN also increases utilization of low-level features, leading to improved accuracy than FPN and T-FPN. And L-FPN explores the usefulness of low-level feature explicitly and sufficiently, resulting in further improvement on accuracy. In fabric defect recognition, low-level information plays a more important role than high-level information. And L-FPN which makes better use of low-level features is more suitable for fabric defect detection. The experimental results are consistent with our analysis.

4.4.2. R-compound scaling

In this section, we evaluate an efficient strategy to uniformly scale the depth, width and resolution of our network, which is called R-Compound Scaling. In this experiment, the contribution of R-Compound Scaling is verified.

Fig. 7 shows a comparison of detector performance under different parameter adjustment strategies. R-Compound Scaling have obviously better performance than single-factor adjustment strategies and as the input resolution increases, the advantages of R-Compound Scaling become more significant. One can find that although we adjust the network from the same baseline network, the strategy of sacrificing the width and depth to increase the input resolution is more conducive to obtain higher efficiency and accuracy.

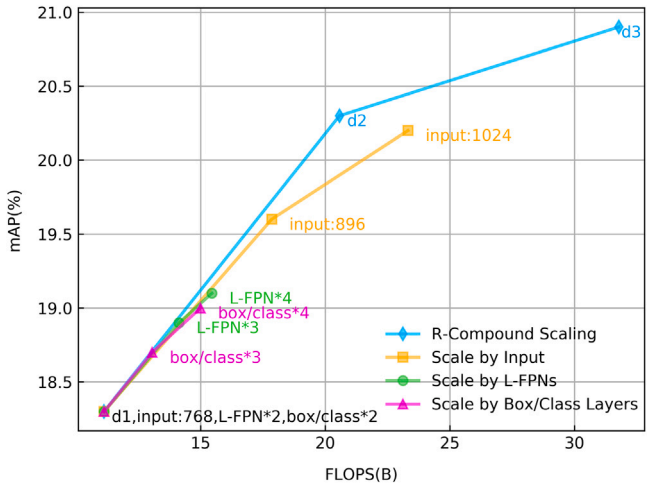


Fig. 7. Comparison of different scale adjustment strategies. The horizontal axis represents FLOPs and the vertical axis represents accuracy. Each strategy can improve accuracy, but our R-Compound Scaling method achieves higher efficiency.

Table 5
Comparison Among Different Baseline Networks.

Model	mAP	Para.	FLOPs
EDD-d1 w EfficientNet-b0	18.3	4.55M	11.12B
EDD-d1 w EfficientNet-b1	18.8	7.05M	13.34B
EDD-d2 w EfficientNet-b1	20.3	7.24M	20.57B
EDD-d2 w EfficientNet-b2	20.6	8.38M	21.96B
EDD-d3 w EfficientNet-b2	20.9	8.59M	31.78B
EDD-d3 w EfficientNet-b3	21.2	11.55M	38.17B

4.4.3. Lightweight backbone

In this section, we evaluate our detectors have better trade-off with fewer parameters and FLOPs. It can be seen from Table 5 that reducing depth and width of the backbone hardly effect the detection accuracy. mAP of EDD-d1 and EDD-d3 drop by 0.5 and 0.3 respectively. But at the same time, the efficiency of our detector has improved significantly. Notably parameters of EDD-d1 drops by 2.5M (35.5%), parameters of EDD-d3 drops by 2.96M (25.6%). We believe that it is worthwhile to sacrifice the backbone size to improve detector efficiency. Especially, when the input resolution is large, the advantages of lightweight backbone will be more obvious.

In order to further understand the advantages of our lightweight backbone, Fig. 1 shows the trend of FLOPs and mAP with the increasing input resolution of several common one-stage detectors. Notably, as the input size increases, EDDs achieve higher efficiency. Our EDDs have fewer parameters than other one-stage detectors, which also proves that the proposed EDDs have more competitive performance in fabric defect detection in industrial scenes.

5. Conclusion

In this paper, we present a family of EDDs for fabric quality inspection. The proposed EDDs utilize a R-Compound Scaling strategy to adjust the depth, width and input resolution so that a series of detectors can be defined. Based on the characters of fabric defect detection, the L-FPN and lightweight backbone are developed to improve the efficiency of EDDs. The former can guide the network to focus more low-level feature which is significant for distinguishing the defects. The later can retain more resources for larger size input, leading to improvement on both accuracy and real-time performance. By adopting the above strategies, EDDs show excellent performance when comparing existing detectors in fabric defect detection even with considerable fewer parameters and FLOPs. As the better trade-off made by EDDs, it is certainly helpful to be utilized in different fabric production scenarios with different resource restrictions.

Future work will be carried out in the following aspects. Firstly, lightweight EfficientNet is used as the backbone in the paper based on the consideration of the characteristics of fabric defect. Although effective, EfficientNet is developed for natural image. The design of backbone network for defect detection would be investigated in the future for further improvement. Secondly, owing to the low probability of occurrence of defective samples in industrial production, it is difficult to collect sufficient defective images. Moreover, accurate labeling of defective images involves much human effort and is commercially expensive, which hinders extensive application of DCNN in industrial fields. In the future, we will investigate effective training strategy to use fewer labeled defective images to accomplish the training process.

CRediT authorship contribution statement

Tong Zhou: Methodology, Software, Data curation, Writing - original draft, Validation, Investigation, Visualization. **Jiabin Zhang:** Conceptualization, Methodology, Software, Data curation, Writing - original draft, Validation, Investigation, Resources. **Hu Su:** Conceptualization, Methodology, Data curation, Investigation, Writing - original draft, Project administration. **Wei Zou:** Supervision, Writing - review & editing, Funding acquisition. **Bohao Zhang:** Software, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Key Research and Development Program of China under Grant 2018YFB1306303, and in part by the National Natural Science Foundation of China under Grant 61773374 and 61702323, and in part by the Major Basic Research Projects of Natural Science Foundation of Shandong Province, China under Grant ZR2019ZD07.

References

[1] A. Kumar, Computer-vision-based fabric defect detection: A survey, IEEE Trans. Ind. Electron. 55 (1) (2008) 348–363.
[2] Henry Y.T. Ngan, Grantham K.H. Pang, Nelson H.C. Yung, Performance evaluation for motif-based patterned texture defect detection, IEEE Trans. Autom. Sci. Eng. 7 (1) (2010) 58–72.
[3] Tamás Czimmermann, Gastone Ciuti, Mario Milazzo, Marcello Chiurazzi, Paolo Dario, Visual-based defect detection and classification approaches for industrial applications—A SURVEY, Sensors 20 (5) (2020) 1459.
[4] Qiwu Luo, Xiaoxin Fang, Li Liu, Chunhua Yang, Yichuang Sun, Automated visual defect detection for flat steel surface: A survey, IEEE Trans. Instrum. Meas. 69 (3) (2020) 626–644.
[5] Roland T. Chin, Automated visual inspection: 1981 to 1987, Comput. Vis. Graph. Image Process. 41 (3) (1988) 346–381.
[6] Mihran Tuceryan, Anil K. Jain, Texture analysis, in: Handbook of Pattern Recognition and Computer Vision, World Scientific, 1993, pp. 235–276.
[7] C. Boukouvalas, J. Kittler, Color grading of randomly textured ceramic tiles using color histograms, IEEE Trans. Ind. Electron. 46 (1) (1999) 219–226.
[8] Şaban Öztürk, Bayram Akdemir, Fuzzy logic-based segmentation of manufacturing defects on reflective surfaces, Neural Comput. Appl. 29 (8) (2018) 107–116.
[9] Chi-ho Chan, Grantham K.H. Pang, Fabric defect detection by fourier analysis, IEEE Trans. Ind. Appl. 36 (5) (2000) 1267–1276.
[10] Şaban Öztürk, Bayram Akdemir, Real-time product quality control system using optimized Gabor filter bank, Int. J. Adv. Manuf. Technol. 96 (1–4) (2018) 11–19.
[11] Jianchang Mao, Anil K. Jain, Texture classification and segmentation using multiresolution simultaneous autoregressive models, Pattern Recognit. 25 (2) (1992) 173–188.
[12] F.S. Cohen, Z. Fan, Automated inspection of textile fabrics using textural models, IEEE Trans. Pattern Anal. Mach. Intell. 13 (8) (1991) 803–808.

[13] Franz Pernkopf, Detection of surface defects on raw steel blocks using Bayesian network classifiers, Pattern Anal. Appl. 7 (3) (2004) 333–342.
[14] Olli Silvén, Matti Niskanen, Hannu Kauppinen, Wood inspection with non-supervised clustering, Mach. Vis. Appl. 13 (5–6) (2003) 275–285.
[15] Xianghua Xie, Majid Mirmehdi, TEXEMS: Texture exemplars for defect detection on random textured surfaces, IEEE Trans. Pattern Anal. Mach. Intell. 29 (8) (2007) 1454–1464.
[16] Xian Tao, Wei Hou, De Xu, A survey of surface defect detection methods based on deep learning, Acta Automat. Sinica online available (2020).
[17] Jonathan Masci, Ueli Meier, Gabriel Fricout, Jürgen Schmidhuber, Multi-scale pyramidal pooling network for generic steel defect classification, in: International Joint Conference on Neural Networks, 2013, pp. 1–8.
[18] Tian Wang, Yang Chen, Meina Qiao, Hichem Snoussi, A fast and robust convolutional neural network-based defect detection model in product quality control, Int. J. Adv. Manuf. Technol. 94 (9–12) (2018) 3465–3471.
[19] Young-Jin Cha, Wooram Choi, Gahyun Suh, Sadegh Mahmoudkhani, Oral Büyüköztürk, Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types, Comput.-Aided Civ. Infrastruct. Eng. 33 (9) (2018) 731–747.
[20] Junwen Chen, Zhigang Liu, Hongrui Wang, Alfredo Núñez, Zhiwei Han, Automatic defect detection of fasteners on the catenary support device using deep convolutional neural network, IEEE Trans. Instrum. Meas. 67 (2) (2017) 257–269.
[21] Hong-wei Zhang, Ling-jie Zhang, Peng-fei Li, De Gu, Yarn-dyed fabric defect detection with YOLOV2 based on deep convolution neural networks, in: 2018 IEEE 7th Data Driven Control and Learning Systems Conference, DDCLS, IEEE, 2018, pp. 170–174.
[22] Xiao Liang, Image-based post-disaster inspection of reinforced concrete bridge systems using deep learning with Bayesian optimization, Comput.-Aided Civil Infrastruct. Eng. 34 (5) (2019) 415–430.
[23] Olaf Ronneberger, Philipp Fischer, Thomas Brox, U-Net: Convolutional networks for biomedical image segmentation, in: Nassir Navab, Joachim Hornegger, William M. Wells, Alejandro F. Frangi (Eds.), Medical Image Computing and Computer-Assisted Intervention, MICCAI 2015, Springer International Publishing, Cham, 2015, pp. 234–241.
[24] Jifeng Dai, Kaiming He, Jian Sun, Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1635–1643.
[25] Ruoxu Ren, Terence Hung, Kay Chen Tan, A generic deep-learning-based approach for automated surface inspection, IEEE Trans. Cybern. 48 (3) (2017) 929–940.
[26] Vidhya Natarajan, Tzu-Yi Hung, Sriram Vaikundam, Liang-Tien Chia, Convolutional networks for voting-based anomaly classification in metal surface inspection, in: IEEE International Conference on Industrial Technology, 2017, pp. 986–991.
[27] Shiyang Zhou, Youping Chen, Dailin Zhang, Jingming Xie, Yunfei Zhou, Classification of surface defects on steel sheet using convolutional neural networks, Mater. Tehnol. 51 (1) (2017) 123–131.
[28] Srinath S Kumar, Dulcy M Abraham, Mohammad R Jahanshahi, Tom Iseley, Justin Starr, Automated defect classification in sewer closed circuit television inspections using deep convolutional neural networks, Autom. Constr. 91 (2018) 273–283.
[29] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, 2015, pp. 91–99.
[30] Joseph Redmon, Ali Farhadi, YOLO9000: better, faster, stronger, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7263–7271.
[31] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.
[32] Ross Girshick, Fast r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.
[33] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, 2015, pp. 91–99.
[34] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2980–2988.
[35] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.
[36] Joseph Redmon, Ali Farhadi, YOLOv3: An incremental improvement, 2018, arXiv: Computer Vision and Pattern Recognition.
[37] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg, Ssd: Single shot multibox detector, in: European Conference on Computer Vision, Springer, 2016, pp. 21–37.
[38] Mingxing Tan, Ruoming Pang, Quoc V. Le, Efficientdet: Scalable and efficient object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10781–10790.
[39] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, Serge Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2117–2125.
[40] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, Jiaya Jia, Path aggregation network for instance segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8759–8768.
[41] Qijie Zhao, Tao Sheng, Yongtao Wang, Zhi Tang, Ying Chen, Ling Cai, Haibin Ling, M2det: A single-shot object detector based on multi-level feature pyramid network, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 9259–9266.
[42] Mingxing Tan, Quoc Le, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, in: International Conference on Machine Learning, 2019, pp. 6105–6114.
[43] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[44] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, Quoc V. Le, Mnasnet: Platform-aware neural architecture search for mobile, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 2820–2828.
[45] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520.