Object Detection: Training From Scratch

KAI ZHAO, YAN ZHOU, AND XIN CHEN
Air Force Early Warning Academy, Wuhan 430019, China
Corresponding author: Yan Zhou (ch51322@163.com)

ABSTRACT The development of deep neural networks has driven the development of computer vision. Deep neural networks play an important role in object detection. To improve network performance, before using neural networks for object detection, they are commonly pre-trained on the data set and fine-tuned to their object detection tasks. Pre-training is not always helpful in object detection tasks, so studies have been performed on training neural networks from scratch. By consulting many relevant studies, we performed a systematic analysis of training networks from scratch for object detection. Our article is divided into the following three parts: (i) the reasons for which target detection requires training from scratch, (ii) mainstream networks that can be trained from scratch, and (iii) the criteria for training from scratch. Finally, we summarize some research directions relevant to this topic.

INDEX TERMS Object detection, train from scratch, deep neural networks.

I. INTRODUCTION
Convolution Neural Networks (CNNs) play an important role in computer vision and perform well in areas such as object classification, object detection, and semantic segmentation. Object classification is used to sort the objects in an image into specific categories. Semantic segmentation is also called pixel segmentation. Objects in an image are classified into specific categories by pixels. Object detection is used not only to classify the objects in the image, but also to locate the objects.

Traditional object detection methods can be divided into three steps: candidate frame generation, feature vector extraction, and region classification [1]. The features used in traditional object detection algorithms are artificially designed, for example, scale invariant feature transformation [2], histogram of gradient [3], and speeded up robust features [4]. These features are used to identify an object, and then the object is combined with the corresponding strategy to locate the same.

With the improvement of storage capacity and computing ability, images have greatly grown in quantity and quality. Traditional object detection methods cannot deal with high-resolution and massive data. The rise of machine learning has led to the development of computer vision. CNNs have outstanding performance in big data processing, and they are increasingly being applied to computer vision tasks.

Le et al. [5], [6] proposed the first CNN, LeNet, and established its basic structure: convolution layers, pooling layers, and fully connected layers. However, owing to the limitation of computing power, CNNs have been overshadowed by Support Vector Machines (SVMs) [7] and other algorithms. Early CNNs were used for face detection [8], [9], face recognition [10], character recognition [10], and other applications. The vanishing gradient problem, as well as limitations in the number of training samples and computing ability, limit the use of neural networks.

The appearance of the Rectified Linear Unit (ReLU) and Dropout [11] and the acceleration of GPUs have promoted the development of Deep Neural Networks (DNNs). The DNNs was originate from AlexNet [12]. Compared with the LeNet, AlexNet has a deeper level and more parameters. AlexNet appeared in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012. ILSVRC is one of the most popular and authoritative academic competitions in computer vision in recent years, representing the highest level. The ILSVRC is based on the Imagennet [13], and mainly considers three aspects: image classification, single object localization, and object detection. VGGNet [14], GoogleNet [15], and Residual Neural Network (ResNet) [16], among others, have achieved good results in this competitions.

This paper focuses on the application of training from scratch in object detection, most previous reviews have focused on the application of DNNs in object detection, and there is no specific discussion on training from scratch. We study the latest related literature and conduct a systematic analysis of the application of training from scratch in object detection. Our main contributions are as follows:
1) This paper discusses the latest research on training from scratch but does not go into great detail for ease of reader understanding.
2) We systematically analyze the importance of training from scratch in object detection.
3) We briefly introduce the networks specially designed for training from scratch, summarize the necessary conditions for training a neural network from scratch, and make some predictions and analysis on the development trend of training from scratch.

In summary, we aimed to summarize the development of training from scratch in object detection, not a comprehensive summary of the application of deep learning in object detection, specifically for those readers who want to gain an understanding of the application of training from scratch in object detection or achieving training from scratch.

The remainder of this paper is organized as follows. The first part presents our contribution and writing motivation. The second part provides a brief introduction to DNNs. The third part briefly introduces transfer learning, the reasons people use transfer learning to train networks, and some problems in transfer learning. The fourth part introduces representative object detection networks specially designed for training from scratch. The fifth part summarizes the necessary conditions for the network to achieve training from scratch. In the sixth part, we make a prediction on the development trend of training from scratch.

II. DNNs

A. ONE-STAGE AND TWO-STAGE DETECTORS

Current neural networks for object detection can be divided into two categories: i) One-stage detectors, including You Only Look Once (YOLO) [17] and its variants [18], [19], Single Shot MultiBox Detector (SSD) [20] and its variants [21], RetinaNet [22], etc. One-stage detectors directly return the class probability and position coordinates of an object. ii) Two-stage detectors including Fast Region-based Convolutional Network (Fast R-CNN) [23], Faster R-CNN [24], etc. Two-stage detectors use proposal generators to generate sparse proposals, extract features from these proposals, and then classify the proposals through a convolutional neural network. Under normal circumstances, two-stage detectors can achieve higher recognition accuracy compared with one-stage detectors. In many cases, the highest recognition accuracies are achieved by two-stage detectors, but one-stage detectors have faster detection speeds, so they can achieve real-time detection, which is more suitable for low-end applications. The recognition mechanisms of one-stage detectors and two-stage detectors are shown in Figure 1.

B. BASIC ARCHITECTURE OF DNN

CNNs have excellent performance in computer vision. The structure of most CNNs is derived from LeNet, which is formed by stacking a series of convolution layers. It also includes pooling layers, nonlinear activation layers, and fully connected layers. The convolution layer converts the input image into a feature map through an n x n convolution kernel. The feature map can be regarded as a multi-channel image, and each channel represents different image information. The size of the mapped area of the pixels on the feature map on the original image is called the receptive field. A nonlinear activation layer is applied after the feature map. The pooling layer is used to compress the data size, increase the calculation speed, and enhance the robustness of the network. The fully connected layer is mainly used for classification, mapping distributed features to sample label space.
Through a reasonable combination of convolution layers, nonlinear activation layers, pooling layers, and fully connected layers, a basic DNN structure has been established. To improve network performance, people usually define corresponding loss functions, such as cross-entropy loss and focal loss [22], and optimize the network through gradient-based optimization methods, such as Stochastic gradient descent (SGD) and Adaptive Moment Estimation (Adam) [25].

1) BACKBONE NETWORKS
The backbone network plays the role of feature extraction in the object detection task, converts the input image into feature maps, and the subsequent network makes predictions based on the feature maps. The backbone network used for object detection at this stage stems from the modification of the classification network. Many neural networks are closely related to ImageNet. Many networks are designed for better performance on ImageNet. AlexNet, VGGNet, ResNet, etc. were originally designed for ImageNet classification tasks. Many DNNs used for object detection refer to these networks.

In the actual process, for accuracy and efficiency, people can flexibly choose the corresponding backbone network. To achieve higher recognition accuracy, people can choose deep and densely connected backbone networks, such as ResNet and ResNeXt [26]. To apply the network to low-end devices, some lightweight backbone networks can be selected, such as MobileNet-v1 [27] and MobileNet-v2 [28].

The widely used backbone networks include VGG16 [14], ResNet, and ResNeXt. VGG16 is based on AlexNet. VGG16 consists of 5 groups of convolution layers and three fully connected layers. The first two groups of convolution layers include two convolution layers, and the last three groups include three convolution layers. There is a maximum pooling layer between each group. AlexNet and VGG16 prove that stacked convolution layers can improve network performance. However, as the depth of the network increases, the network becomes difficult to optimize. Thus, people have tried to solve the problem of network optimization. Szegedy et al. [29] added auxiliary loss as an additional supervision in the middle layer, but it did not solve the problem sufficiently. He et al. [16] proposed ResNet on the basis of VGG19. In view of the problem that an increase in network depth will lead to difficulty in network optimization, they proposed a shortcut connection, also known as skip connection, referring to the idea of a highway network. That is, they added a direct connection channel to transfer the original information to a deeper layer directly, a shortcut connection and several convolution layers constitute a residual block. With the help of residual blocks, the ResNet model depth can reach an amazing 152.

After that, they changed post-activation to pre-activation and found that batch normalization (BN) played an important role in network promotion. Based on this, they proposed ResNet V2 [30], which increased the network depth from 152 to more than 1000. To make full use of the features, Iandola et al. [31] proposed a more radical and intensive connection mechanism, where all layers are connected to each other, and each layer in DenseNet is connected through the channel dimension, which directly concatenates the feature maps from different layers, thereby realizing feature reuse and improving efficiency. The connection modes of shortcut connection and dense connection are shown in Figure 2.

![Different connections.](image)

Based on ResNet and Inception [32], Xie et al. [26] proposed ResNeXt. The essence of ResNeXt is the use of group convolution, which can obtain more abundant feature representation. Later, Howard et al. proposed MobileNet [27], which uses depth-wise separable convolution to improve the calculation speed of the network. Depth-wise separate convolution includes depth-wise convolution and point-wise convolution. To balance the calculation speed and accuracy of the network, the width multiplier is also used to introduce a new hyperparameter to adjust the number of channels of the convolution output. MobileNet is a network designed specifically for mobile devices. Later, inverted residual was introduced in MobileNet-v2 [28] to increase the extraction of image features to improve accuracy, and linear bottlenecks were introduced to avoid information loss of nonlinear functions.

The above mentioned networks are designed for image classification. Most of them are pre-trained on ImageNet, and they use the obtained parameters as initial conditions for the next task. However, directly applying the pre-trained network to the object detection task is not the best solution for the following reasons: i) Compared to object detection, a larger receptive field is beneficial to the classification task; therefore, the network usually uses a larger downsampling factor, but this will lead to the blurring of the boundary of large objects and the loss of information of small objects. ii) The classification and object detection tasks require different stages. The classification task usually consists of additional stages. Therefore, Li et al. designed DetNet [33], a backbone network specifically for object detection. DetNet is based on ResNet50 [16] and retains stages 1-4 of ResNet, while stage 6 is added. Stages 5 and 6 all retain the same
channels. To obtain a higher resolution and receptive field, a dilated bottleneck is introduced before each stage. Although DetNet is specifically designed for object detection, it still must be pre-trained on ImageNet.

III. PROPOSAL GENERATION
Proposal generation plays an important role in the process of object detection. A generation generates a series of rectangular bounding boxes (bbox), which represent potential targets. These proposals are used for classification and refinement. The method of proposal generation can be divided into three categories: traditional, anchor-based, and anchor-free.

Traditional methods: Traditional proposal generation methods are usually based on low-level features, such as color, texture, and edges. These techniques can be categorized based on three principles: i) computing the “objectiveness score” of a candidate box, ii) merging super-pixels from original images, and iii) generating multiple foreground and background segments [1].

Anchor-based methods: This type of method is widely used to generate proposals based on predefined anchors. Girshick et al. proposed the Region Proposal Network (RPN) [23] in Fast R-CNN. The RPN generates proposals based on the feature map generated by the network. The SSD also refers to the RPN. Other works have adopted different anchor design strategies, such as [34]–[37].

Anchor-free methods: In addition to the methods mentioned above, there are some other proposal generation methods. There are corner-based and center-based methods. The representative corner-based method is CornerNet [38]. Compared with anchor-based methods, the upper left corner and lower right corner of the feature map are used to predict the bounding box. The representative center-based methods are CenterNet [39] and CenterNet-keypoint [40]. Compared with CornerNet, CenterNet also uses the center point. It takes the upper left corner, lower right corner, and center point as a triple to predict the bbox. CenterNet-keypoint only uses the center point instead of the corner points to predict the bbox.

IV. TRANSFER LEARNING
Human beings have the ability to transfer learning, which can transfer the knowledge learned in one situation to another. A neural network is designed to simulate the pattern of the human brain. Giving neural networks the same ability of transfer learning, which gives the network a strong ability to transform.

A deep neural network has an obvious hierarchical structure. The bottom layer of the model usually has low-level semantic features (such as edge information and color information). The top layer of the model uses high-level semantic features for classification and recognition. Generally, low-level semantic features are unchanged in different classification tasks. The real difference is high-level information, which gives the network a strong ability to transform.

Currently, a neural network is trained from scratch only in few cases. Training from scratch requires a large amount of data, which can be difficult to obtain. Even if there are sufficient data, the network will take a long time to converge [47]. In general, it is better to start training with the weights obtained from a previous training, rather than to start training with randomly initialized weights [48].

As the difference between the pre-training dataset and the target task data increases, the transferability will decrease. Yosinski et al. [49] proved that even features learned from less relevant tasks are better than those directly learned from random initialization.

A more common training method is to pre-train the network to be used for a dataset and save the parameters. During the detection process, the network is initialized with these parameters and is then fine-tuned for specific tasks, which is referred to as transfer learning. Transfer learning can save significant training time, so in most cases, pre-trained parameters are used as initial conditions when performing object detection. In image classification and semantic segmentation tasks, pre-training has helped the network achieve state-of-the-art results [24], [50]–[53]. Pre-training and fine-tuning have also played an important role in object detection tasks [50], [54], and since then pre-training and fine-tuning have become a paradigm.

The emergence of some larger data sets [55] has further improved the performance of pre-training in image classification, but with the increase in the number of dataset, the improvement in object detection tasks as a result of pre-training has been very limited [55].

He et al. [56] studied the ImageNet [13] pre-training method and determined that, although pre-training can speed up network convergence, the total training time is the same as that of random initialization. Moreover, using the ImageNet pre-training method does not automatically provide better regularization. When the training data is less, additional hyperparameter adjustments are needed to avoid overfitting, but if these hyperparameters are applied to the network trained from scratch, the same effect can be obtained. When the target task requires more use of local spatial information for prediction, the ImageNet pre-training method does not show much better results.

In addition, in the object detection task, pre-training still has the following issues:

1) Most of the datasets used for pre-training, such as ImageNet, are designed for classification tasks. Most networks are pre-trained on these datasets and then fine-tuned. Object detection and object classification have different transformation sensitivities. Object classification has transformation invariance: when the size or shape of an image changes, the classification result should be constant. Object detection has transformation for transformation: when the image size or position changes, the detection result will also change.

2) The pre-training network structure is relatively complicated; the structure of the network is relatively fixed, and the remaining adjustment space is limited. If we
want to modify the network during the training, it is more difficult.

3) In many cases, the domain of the pre-trained dataset and the target domain are slightly different. This is known as domain mismatch [57]. Fine-tuning is an important means to reduce domain mismatch. Sometimes the domains are too different, and fine-tuning cannot completely eliminate the differences, such as for RGB images and medical images, and RGB images and SAR images.

The above issues indicate that pre-training on the object detection task is not an optimal solution. Many researchers have begun to study the possibility of training the network from scratch.

V. RELATED NETWORKS

Because of the gap between object detection and classification, research on training from scratch is constantly being conducted. Such research can be divided into two categories: design specifically for training from scratch and avoiding pre-training to successfully implement training from scratch.

Starting from the first train-from-scratch network, Deeply Supervised Object Detector (DSOD) [57], which was proposed by Shen et al. in 2016, studies on training from scratch have been ongoing, and there are also some representative networks, such as Tiny-DSOD [58], Gated Feature Reuse Deeply Supervised Object Detector (GFR-DSOD) [59], DetNet [33], and ScratchDet [60]. The publication time of these networks is shown in Figure 3.

A. DSOD

DSOD [56] was the first network specifically designed to be trained from scratch. DSOD refers to SSD and DenseNet [31]. The DSOD network is a one-stage network. DSOD can be divided into a backbone sub-network and a front-end sub-network. The backbone network uses a structure similar to DenseNet. The backbone network consists of a stem block, four dense blocks, two transformation layers, and two transformation layers without pooling. The stem block consists of three convolution layers and a pooling layer.

The design of the backbone network reflects the principles of deep supervision and stem block. Deep supervision refers to DenseNet’s dense layer-wise connection. A layer in a dense block is connected to all layers in front of it. This can enhance the supervision signal and make the model converge faster. The simple design of stem block is derived from Google’s Inception V3, which reduces the information loss of the original image. Traditional CNNs incorporate pooling operations or large convolution kernels to reduce the size of feature maps before convolution. This can overcome the limitations of storage and computing resources due to the large size of the input image, but it will lead to excessive loss of image information. The bottleneck module of DSOD is placed in a position closest to the input image, and the size is reduced through multiple convolution operations, which can reduce the loss of input image information. Compared with the original DenseNet, stem block reduces the information loss of the original image and significantly improves the detection performance.

The front-end sub-network of DSOD is a dense prediction structure, and a feature fusion module is added, based on the plain structure of SSD. In this way, the adjacent shallow feature maps are fused with the high-level feature maps. The principle is learning half and reusing half. In DSOD, except scale-1, half of the feature maps are learned from the previous scale, and the remaining feature maps are down-sampled from the contiguous high-resolution feature maps.

DSOD shows competitive accuracy with fewer parameters on PASCAL VOC [66] and MS COCO [61] compared to state-of-the-art detectors, such as SSD and Region-based Fully Convolutional Networks (R-FCN) [62].
B. GFR-DSOD
Based on DSOD, Shen et al. proposed GFR-DSOD [59]. Compared to DSOD, it mainly improved the front-end sub-network used for prediction. The backbone network is the same as the DSOD backbone network, and GFR-DSOD’s front-end sub-network can dynamically adjust the strength of the middle layer’s supervision signal for targets of different sizes. There are two main innovations in GFR-DSOD: iterative feature pyramid and gating mechanism.

1) Iterative Feature Pyramid: This includes the down-sampling pathway and the up-sampling pathway. As shown in Figure 4, the down-sampling pathway takes the low-level feature maps, and after passing through the down-sampling module, the output is concatenated with the current features. The up-sampling pathway concatenates high-level features to the current features. The concatenation operation is repeated in each scale of the prediction layers.

2) Gating Mechanism: The gating mechanism consists of three different levels of gates: i) Channel-level attention, which aims to model relationships between channels. Here, a squeeze-and-excitation block [50] is used as the channel-level attention. ii) Global-level attention, which aims to adaptively enhance different scale supervision. iii) Identity mapping, which retains the current feature for future processing.

Compared to DSOD, GFR-DSOD offers better performance on PASCAL VOC and MS COCO, the mAP on PASCAL VOC 2007, 2012 increased by 1.4% and 1.7%, and the mAP on COCO increased by 0.6%.

C. TINY-DSOD
Shen et al. proposed Tiny-DSOD [58], which is a lightweight network proposed for use with limited computing resources. Based on DSOD, Tiny-DSOD offers a good trade-off between speed and accuracy. To balance the computation and recognition accuracy, Tiny-DSOD proposes two new structures: the backbone network of the Depth-wise Dense Block (DDB) and the front-end network of the depth-wise feature pyramid network (D-FPN). In Tiny-DSOD, DDB is used instead of dense blocks in DenseNet, which reduces the computation and maintains deep supervision.

The author proposed two types of DDB: DDB-a and DDB-b. The parameters of DDB-a increase rapidly when multi-layer stacking is used, which consumes significant resources, and there are many 1 x 1 convolutions, which leads to redundancy. Therefore, the author proposed DDB-b. First, the input channel is compressed, and then the depth-wise convolution is performed to aggregate the obtained feature map with the input. The output of depth-wise convolution is directly aggregated to the input, and there is no 1 x 1 mapping; the complexity after stacking is significantly reduced compared to that for DDB-a, and the accuracy is higher with the same amount of calculation.

The author believes that, in the front-end, the first few layers of the front-end structure of SSD and DSOD lack semantic object information. Therefore, a feature-pyramid-network (FPN) is introduced into the framework to achieve fusion of feature maps at different scales. A depth-wise feature pyramid-network (D-FPN) is proposed. D-FPN significantly increases the detection accuracy.

Tiny-DSOD is better than some other high-efficiency detectors, such as Tiny-YOLO, SqueezeDet, and MobileNet-SSD. Compared to DSOD, the number of parameters is reduced to 1/6, the number of FLOPs is reduced to 1/5, and the accuracy rate is only reduced by 1.5%.

D. ScratchDet
Zhu et al. [60] believe that BN is the key to training from scratch. Based on the addition of a BN layer and the Root-block backbone network, they designed a new model, ScratchDet, which can be trained from scratch. Root-block is based on ResNet with some modifications.

The first change is to remove ResNet’s downsampling operation on the first convolution layer and replace $7 \times 7$ convolution layer with several stacked $3 \times 3$ convolution layers.

The second change is to replace four convolution blocks with four residual blocks, each of which consists of two branches. Branch 1 is a $1 \times 1$ convolution block with strip 2, and branch 2 is a $3 \times 3$ convolution block with strip 2 and a $3 \times 3$ convolution block with strip 1. These residual blocks can improve the calculation efficiency and do not need dropout.

With the Root-ResNet backbone network, ScratchDet achieves state-of-the-art results on the PASCAL VOC 2007 (80.4%), 2012 (78.5%) and MS COCO (32.7%) for the training from scratch detectors.

E. RELATING WORKS
In addition to the above-mentioned networks, some work have also achieved training from scratch by modifying the existing network. He et al. [56] have achieved training from scratch of the SSD network by adding BN layers. He et al. also achieved training from scratch on the Mask R-CNN network by replacing the BN layer with group
normalization [63] and synchronized batch normalization [64, 65].

The performance of the above networks on PASCAL VOC and MS COCO are shown in Table 1 and Table 2, including mean Average Precision (mAP), Frame-Per-Second (FPS), and parameters, etc.

VI. RELEVANT CRITERIA

Currently, there are few rules about how to design a network that can be trained from scratch or to train a deep neural network from scratch. Most of them are summarized from experiments. Shen et al. [57] summed up several rules based on experiments, and designed DSOD, the first training-from-scratch detector.

The first criterion they proposed is proposal-free. In the experiment, they tested three types of convolutional neural network detectors. The first type is a network that needs external object proposal generators, such as R-CNN and Fast R-CNN. The second type requires an integrated region-proposal network to generate relatively fewer region proposals, such as Faster R-CNN and R-FCN. The third type is a single-shot and proposal-free method, such as YOLO [17] and SSD. It was found that only the third kind of proposal-free network can realize training from scratch. The second criterion is deep supervision. GoogleNet [15] and Deeply-Supervised network (DSN) [67], and DeepID3 [68] have proved the effectiveness of deep supervision.

The core idea of deep supervision is to directly provide an integrated objective function for direct supervision to the earlier hidden layers [57]. The gradient can be transferred to the input layer, which can alleviate the problem of gradient disappearance. The third criterion is stem block, inspired by Inception-v3 [69] and v4 [70]. The author constructs the stem block, which is composed of three $3 \times 3$ convolution layers and a max pooling layer. This structure can improve the robustness of the network. The fourth criterion is dense prediction structure. In the plain structure of SSD, the features of each layer are directly transformed from those of the previous layer, and the dense structure of DSOD integrates multiple levels of scale information. The DSOD network designed by the author according to these four criteria realizes training from scratch. Later, they designed Tiny-DSOD and GFR-DSOD on this basis.

Shen et al. [57] believe that only one-stage networks can train from scratch, and two-stage networks need pre-training parameters to initialize the network; otherwise, the gradient disappears when training from scratch, leading to training failure or poor performance. They also emphasized the role of deep supervision. Later, Zhu et al. [60] and He et al. [56] also realized training from scratch on two-stage networks, which proved that one-stage is not a necessary criterion for training from scratch.

Zhang et al. [60] considered that, when training from scratch, the lack of BN [71] was the main cause of poor convergence. They integrated BN into all parts of SSD and found that, when BN was added to any part of SSD, the network’s mAP was significantly improved. They believe that, compared to deep supervision, BN is the key to ensure that a network can successfully train from scratch. According to this theory, they designed a new model, ScratchDet, and a new backbone, Root-ResNet.

Based on some current research results, we have summarized some guidelines for achieving training from scratch:

1) Requires stable gradients: if we want to train from scratch, we need some appropriate methods to stabilize the gradient, such as deep supervision, BN, group normalization [63], or synchronized batch normalization [64], [65]. We believe that the purpose is to ensure the stability of the gradient during the training process. He et al. [56] have also shown in experiments that, in a shallow network, BN is not used and only normalization is used but they also realized training from scratch. Zhu et al. [60] removed the BN layer in DSOD, and the network performance was significantly degraded, which also proved that maintaining gradient stability is an important condition for training from scratch.

2) Needs a longer training time: Shen et al. [57], Li et al. [58], and Zhu et al. [60] showed that, compared to pre-training networks, training from scratch requires more time to converge, which is worthwhile compared to the results obtained by pre-training.
to the time spent in pre-training. He et al. [56] also showed that after more epoch training, both one-stage and two-stage networks achieved better results than pre-training. The article also reported that, to train a network from scratch, one of the conditions is that it takes a long time.

3) Needs sufficient data: the data set used for pre-training must contain sufficient data. Agrawal et al. [72] showed that, even without pre-training, if the data set used for training from scratch is sufficient, the experimental results can reach 90% of those obtained with pre-training. The data set used for pre-training, ImageNet, contains millions of images. During the pre-training process, the network can learn some low-level information. Sufficient training data is an important condition for training from scratch to meet or exceed the performance of pre-training.

We used SSD as the baseline. We compared the current training-from-scratch network to SSD in terms of the gradient stabilization methods, network iterations, and network performance. Table 3 shows that our generalized guidelines are correct.

**VII. CONCLUSION**

With the development of DNNs, DNNs have become increasingly widely used in object detection. However, training a network from scratch is time-consuming and labor-intensive, and it often fails to converge. In this case, pre-training can better solve the problems arising from training from scratch. Pre-training has gradually become a paradigm for network training. However, the study found that pre-training is not always advantageous, especially when the data set used for pre-training is quite different from the task domain. The neglected training from scratch has re-entered people’s vision, and a group of specialized designed networks, including DSOD and ScratchDet, have appeared, and some anchor-free networks have also achieved training from scratch. It is foreseeable that with the development of computing power and optimization of algorithms, training from scratch has great potential. Here, we list some areas related to training from scratch.

1) Research on network structures suitable for training from scratch. Most of the current training-from-scratch networks are modified based on the existing networks, and the structures still have many limitations. Compared to modifying the existing structures, development of network structures that are suitable for training from scratch is a promising research direction.

2) Study the bias between object detection and classification. The reason pre-training does not perform well on object detection is partly because there is a large bias between the two. Study is needed to determine where the bias exists between object detection and classification. Understanding how to alleviate the deviation between the two will also greatly improve the object detection task.

3) Research on network design for special image tasks. Most of the current networks are designed for RGB images. However, for single-channel images, such as medical images and SAR images, there are few specially designed networks. Designing a special network for these images will be useful.

4) Research on pre-training methods that are suitable for object detection. Currently, many pre-training data sets are designed based on classification tasks, and there is no good pre-training data set for object detection tasks. Research is needed to design a special data set for object detection.

5) Determine the guidelines for training from scratch. Sufficient guidelines are not available on how to train from scratch and how to design a network that can train from scratch. Establishment of more general guidelines for training networks from scratch is worth studying.

6) Find more suitable gradient-optimization methods. The current convergence speed of training networks from scratch is slower than that of pre-trained networks. Further study is needed on accelerating the convergence speed of training networks from scratch and improving the efficiency of network training.

**REFERENCES**

[1] X. Wu, D. Sahoo, and S. C. H. Hoi, “Recent advances in deep learning for object detection,” *Neurocomputing*, vol. 396, pp. 39–64, Jul. 2020.

[2] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, 1999, pp. 1–5.

[3] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 866–873.

[4] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, and B. Scholkopf, “Support vector machines,” *IEEE Intell. Syst. Appl.*, vol. 13, no. 4, pp. 18–28, Jul./Aug. 2008.

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Dec. 1998.

[6] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, “Handwritten digit recognition with a back-propagation network,” in *Proc. Adv. Neural Inf. Process. Syst.*, 1990, pp. 396–404.

[7] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
K. Zhao et al.: Object Detection: Training From Scratch

[55] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharat, and L. van der Maaten, “Exploring the limits of weakly supervised pretraining,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 181–196.

[56] K. He, R. Girshick, and P. Dollar, “Rethinking ImageNet pre-training,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 4918–4927.

[57] Z. Shen, Z. Liu, J. Li, Y.-G. Jiang, Y. Chen, and X. Xue, “DSOD: Learning deeply supervised object detectors from scratch,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 1919–1927.

[58] Y. Li, J. Li, W. Lin, and J. Li, “Tiny-DSOD: Lightweight object detection for resource-restricted usages,” 2018, arXiv:1807.11013. [Online]. Available: http://arxiv.org/abs/1807.11013

[59] Z. Shen, H. Shi, J. Yu, H. Phan, R. Feris, L. Cao, D. Liu, X. Wang, T. Huang, and M. Savvides, “Improving object detection from scratch via gated feature reuse,” 2017, arXiv:1712.00886. [Online]. Available: http://arxiv.org/abs/1712.00886

[60] R. Zhu, S. Zhang, X. Wang, W. Lin, H. Shi, L. Bo, and T. Mei, “Scratchdet: Training single-shot object detectors from scratch,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 2268–2277.

[61] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: Common objects in context,” in Proc. Eur. Conf. Comput. Vis., 2014, pp. 740–755.

[62] J. Dai, Y. Li, K. He, and J. Sun, “R-FCN: Object detection via region-based fully convolutional networks,” 2016, arXiv:1605.06409. [Online]. Available: http://arxiv.org/abs/1605.06409

[63] Y. Wu and K. He, “Group normalization,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 3–19.

[64] C. Peng, T. Xiao, Z. Li, Y. Jiang, X. Zhang, K. Jia, G. Yu, and J. Sun, “MegDet: A large mini-batch object detector,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 6181–6189.

[65] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path aggregation network for instance segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 8759–8768.

[66] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The Pascal visual object classes (VOC) challenge,” Int. J. Comput. Vis., vol. 88, no. 2, pp. 303–338, Jun. 2010.

[67] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, “Deeply-supervised nets,” in Proc. Artif. Intell. Statist., 2015, pp. 562–570.

[68] Y. Sun, D. Liang, X. Wang, and X. Tang, “DeepID3: Face recognition with very deep neural networks,” 2015, arXiv:1502.00873. [Online]. Available: http://arxiv.org/abs/1502.00873

[69] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 2818–2826.

[70] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” in Proc. AAAI Conf. Artif. Intell., 2017, pp. 1–8.

[71] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” 2015, arXiv:1502.03167. [Online]. Available: http://arxiv.org/abs/1502.03167

[72] P. Agrawal, R. Girshick, and J. Malik, “Analyzing the performance of multilayer neural networks for object recognition,” in Proc. Eur. Conf. Comput. Vis., 2014, pp. 329–344.

KAI ZHAO received the bachelor of engineering degree in electronic information science and technology from the Ocean University of China, Qingdao, China, in 2018. He is currently pursuing the master’s degree in electronics and communications engineering with the Air Force Early Warning Academy.

His research interests include ship target detection, image recognition, and artificial intelligence.

YAN ZHOU received the Ph.D. degree in control science and engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2000.

He is currently a Professor with the Air Force Early Warning Academy, Wuhan. His research interests include pattern recognition, data fusion, and image processing.

XIN CHEN received the Ph.D. degree in cryptography from the National University of Defense Science and Technology, Changsha, China, in 2010.

He is currently an Associate Professor with the Air Force Early Warning Academy, Wuhan, China. His research interests include the cloud computing, big data processing and application, artificial intelligence, and intelligent image processing.

* * *