Real-Time Small Object Detection Method Based on Improved PVANet (Postprint)

Authors: Duan Binghuan, Wen Pengcheng, Li Peng

Date: 2018-12-13T00:00:00+00:00

Abstract

Existing object detection algorithms primarily focus on large objects in images, with relatively limited research on small objects that suffers from low detection accuracy and failure to meet real-time requirements. To address this, we propose a real-time small object detection method based on the deep learning object detection framework PVANet. First, we construct a benchmark dataset specifically dedicated to small object detection, in which targets occupy a very small proportion of the image and exhibit disturbances such as truncation and occlusion, thereby enabling better evaluation of small object detection methods. Second, by incorporating the Region Proposal Network (RPN), we propose a method for generating high-quality small object proposals to enhance both detection accuracy and speed. Additionally, we employ two novel learning rate strategies, "step" and "inv" , to improve model performance and further boost detection accuracy. On the constructed small object dataset, the proposed method achieves a 10.67% improvement in average detection accuracy and approximately 30% increase in speed compared with the original PVANet algorithm. Experimental results demonstrate that the proposed method is an effective small object detection algorithm that achieves real-time detection performance.

Full Text

Preamble

Real-time Small Object Detection Method Based on Improved PVANet

Duan Binghuan†, Wen Pengcheng, Li Peng
(Aviation Key Laboratory of Science & Technology on Airborne & Missile-borne Computer, AVIC Xi' an Aeronautical Computing Technique Research Institute, Xi' an 710065, China)

Abstract: Existing object detection algorithms primarily focus on large objects in images, with limited research addressing small objects. Current approaches suffer from low detection accuracy and fail to meet real-time requirements. To address these challenges, this paper proposes a real-time small object detection method based on the deep learning framework PVANet. First, we construct a specialized benchmark dataset for small object detection that contains objects occupying a very small proportion of each image, with interference factors such as truncation and occlusion to better evaluate small object detection methods. Second, we propose a method for generating high-quality small object proposals by integrating with the Region Proposal Network (RPN) to improve both detection accuracy and speed. Finally, we adopt two new learning rate strategies—"step"and "inv"—to enhance model performance and further boost detection accuracy. On our constructed small object dataset, the proposed method achieves a 10.67% improvement in mean average precision (mAP) and approximately 30% speedup compared to the original PVANet algorithm. Experimental results demonstrate that our method is an effective small object detection algorithm that achieves real-time detection performance.

Keywords: small object detection; small object dataset; PVANet algorithm; region proposal network; learning rate policy

0 Introduction

In practical applications such as aerial resource exploration and earthquake/fire rescue, targets appear small due to long shooting distances, while complex background information introduces detection interference. Real-time detection of such small objects has become a challenging and hot research topic. In recent years, various object detection algorithms including Faster R-CNN [**?**], SSD [**?**], and YOLOv2 [**?**] have achieved remarkable results in computer vision, demonstrating continuously improving performance on general datasets like PASCAL VOC [**?**]. However, these general datasets typically contain objects that occupy relatively large proportions of images. As evaluated in [**?**], the aforementioned detection algorithms exhibit poor accuracy on small objects, failing to meet the demands of small object detection applications.

Several researchers have addressed small object detection. Chen et al. [**?**] first combined contextual information with the R-CNN algorithm [**?**] for small object detection, improving accuracy compared to traditional methods but suffering from low efficiency and large storage requirements. Subsequent researchers applied improved R-CNN variants—Fast R-CNN [**?**] and Faster R-CNN—to small object detection to enhance accuracy and speed. References [**?**, **?**] utilized contextual information from Fast R-CNN to detect small objects and improve performance. Reference [**?**] employed Faster R-CNN for pedestrian detection, analyzing that detection errors primarily stem from low-resolution feature maps and background interference, and modified the Region Proposal Network (RPN) [**?**]

to improve accuracy. Reference [**?**] used Faster R-CNN to detect small targets like company logos, further analyzing the impact of object size and multi-level feature maps on detection effectiveness. Additionally, some scholars designed new network structures for specific small object categories; reference [**?**] proposed an end-to-end convolutional neural network for small traffic sign detection, outperforming Fast R-CNN in both accuracy and speed. Although these studies have achieved considerable results and provided novel insights, the small objects they investigated still occupy relatively large proportions in images, and their real-time processing capabilities remain inadequate.

This paper focuses on real-time small object detection. We define small objects not as physically small objects in the real world, but in a generalized sense: objects that occupy a very small proportion of an image. Small object detection presents several key challenges. First, compared to the entire image, the target proportion is extremely small, causing significant background interference that increases the difficulty of precise localization. Second, small objects contain fewer pixels, resulting in less effective feature information that can be extracted. Additionally, small objects often appear in large quantities and frequently overlap in practical applications, further increasing detection difficulty.

PVANet [**?**] is a deep yet lightweight convolutional neural network for real-time object detection. It uses a feature extraction network to generate feature maps, then employs the RPN from Faster R-CNN to generate high-quality region proposals for subsequent detection and localization. Tests on general datasets like PASCAL VOC show that PVANet outperforms Faster R-CNN, SSD, YOLOv2, and other algorithms. Notably, PVANet' s feature extraction layers use small convolutional kernels that preserve low-level features as much as possible, which is advantageous for small object detection. Therefore, this paper improves PVANet to enhance small object detection performance. Our main contributions are:

a) We construct a benchmark dataset specialized for small object detection. Compared to datasets used in other small object detection studies, targets in our dataset occupy a smaller image proportion, with incomplete target information from truncation and occlusion increasing detection difficulty, enabling training of more robust small object detection models.

b) To address the poor localization of small objects in the original PVANet, we propose a method for generating high-quality small object proposals that improves detection accuracy and speed. Additionally, we select two new learning rate strategies based on model training characteristics to further enhance performance.

1 Construction of Small Object Dataset

Datasets are critical for analyzing and evaluating deep learning-based network models. The Neovision2 Tower dataset [**?**], constructed by the Defense Advanced Research Projects Agency (DARPA), is a video image dataset for object detection and real-time tracking. Due to long shooting distances, objects in this dataset are small, and the scenes contain numerous cluttered targets with variable illumination and occlusion interference, making it a challenging object detection dataset. Therefore, we selected the Neovision2 Tower dataset to construct our small object dataset. The Neovision2 Tower dataset contains 100 video clips, with each clip extracted into 900 high-resolution PNG images of 1920$×$1080 pixels. High-definition images preserve as much small object information as possible. We constructed our small object dataset following these steps, with formats referencing PASCAL VOC for generalizability:

a) We downsampled images to 960$×$544 pixels and compressed them to .jpg format to match common datasets like PASCAL VOC, which is necessary for accelerating network training.

b) We modified the target bounding boxes. Original annotations consisted of four boundary coordinates—(X1,Y1), (X2,Y2), (X3,Y3), (X4,Y4). Although these four coordinates formed tight boundaries around targets, they were not rectangular and did not conform to general dataset standards. We therefore modified the annotations to rectangular bounding boxes defined by top-left coordinates (Xmin,Ymin) and bottom-right coordinates (Xmax,Ymax), as shown in Figure 1 [Figure 1: see original paper]. The new coordinates can be expressed as:

$$
\begin{aligned}
X_{min} &= \min(X_1, X_2, X_3, X_4) \\
Y_{min} &= \min(Y_1, Y_2, Y_3, Y_4) \\
X_{max} &= \max(X_1, X_2, X_3, X_4) \\
Y_{max} &= \max(Y_1, Y_2, Y_3, Y_4)
\end{aligned}
$$

Figure 1(b) also illustrates examples of truncation and occlusion interference in our constructed dataset. Finally, we divided the processed Tower dataset into two subsets: training and testing, each corresponding to 50 image sets. We randomly selected 10 image sets from the training set as validation data, leaving 40 sets for training. The Neovision2 Tower dataset contains five target categories: person, bicycle, car, truck, and bus. We selected person and bicycle as our small object categories of interest.

The constructed small object dataset includes images from two shooting angles: the first captured by a fixed-angle downward-looking camera, and the second rotated 90 degrees from the first perspective. Using these two viewpoints increases training data diversity and improves model generalization. In our dataset, the

average person size is 17$\times$24$pixels$, $occupying approximately$0.078$\times$544$image$; $the average bicycle size is40\times$$38 pixels, occupying about 0.291% of an image.

The average target proportion in our proposed small object dataset is 0.184%, which is smaller than both the PASCAL VOC general dataset and the small object dataset proposed in [**?**], as shown in Table 1 .

The small object dataset proposed in [**?**] represents an excellent example of dataset construction and has been adopted by many researchers for small object detection. Our dataset offers at least two advantages compared to [**?**]: First, our targets are smaller and feature both passive occlusion from backgrounds and active occlusion between targets. Additionally, since the dataset originates from video, moving targets are frequently truncated at image boundaries, all of which increase detection difficulty and better evaluate model performance on small objects. Second, our use of video images provides temporal information and correlations between frames, enabling continuous sampling of object appearances and facilitating training of more robust detection models.

2.1 Fundamentals of PVANet

PVANet is a lightweight object detection algorithm implemented in two stages. First, the feature extraction network outputs feature maps to the RPN to generate object proposals. Second, these proposals and feature maps pass through pooling and fully connected layers before entering a classification layer to determine object categories and a bounding box regression layer to refine object locations. The overall PVANet framework is shown in Figure 2 [Figure 2: see original paper].

PVANet' s primary contribution is an efficient feature extraction network based on the principle of many layers but few channels. It employs techniques including C.ReLU [**?**], Inception [**?**], HyperNet [**?**], and residual connections [**?**] to generate feature maps, achieving the goal of accelerating performance without reducing detection accuracy. The PVANet feature extraction network is illustrated in Figure 3 [Figure 3: see original paper].

The initial layers of PVANet' s feature extraction network consist of C.ReLU modules. Research shows that convolutional kernels in early CNN layers exhibit negative correlations. Leveraging this characteristic, C.ReLU simply concatenates each convolutional kernel' s output with its negated values, applies scaling or shifting, then performs ReLU computation. This makes each channel' s slope and activation threshold differ from its opposite channel while halving the number of output channels—eliminating the need to store opposite channel parameters without losing accuracy. The C.ReLU module is crucial for PVANet' s lightweight design. Inception modules are used in the remaining feature extraction network. As one of the most cost-effective components for simultaneously capturing small and large objects, Inception modules generate activations for

different receptive field sizes. Specifically, the 1$×$1 convolutional kernels in Inception modules facilitate locating small object proposals and capturing small objects more precisely.

In summary, for real-time small object detection, PVANet offers three advantages over other algorithms: First, it employs C.ReLU modules to reduce computation and improve detection speed. Second, it uses an RPN to generate high-quality object proposals. Third, the Inception module enables preservation of necessary low-level network information, which benefits small object detection.

2.2 Improvements to PVANet for Small Object Detection

Reference [**?**] identifies that the primary challenge in small object detection lies in proposal generation. Therefore, this paper focuses on generating high-quality small object proposals, primarily by setting appropriate anchor boxes in the RPN. Additionally, compared to other hyperparameters, the learning rate is one of the most important factors affecting detection performance, controlling the model' s effective capacity in a more complex manner. When the learning rate is optimal, the model' s effective capacity is maximized. Based on this, we compare different learning rate strategies and select the optimal one to fine-tune the model for improved small object detection.

2.2.1 Generating Small Object Proposals

SelectiveSearch [**?**] and EdgeBoxes [**?**] are commonly used methods for generating object proposals in object detection, achieving good results on general datasets like PASCAL VOC. However, SelectiveSearch is slow, requiring approximately 2 seconds to process one image on a CPU. While EdgeBoxes achieves a good balance between proposal quality and processing speed, it still requires 0.2 seconds per image [**?**]. Compared to the entire detection pipeline, both methods consume too much time in proposal generation and cannot meet real-time requirements. Moreover, SelectiveSearch and EdgeBoxes perform well for large objects but poorly for small objects, as they are sensitive to important features like contours and distinctive colors—features that small objects inherently lack, preventing generation of high-quality small object proposals.

RPN has proven to be the state-of-the-art method for generating object proposals, significantly reducing proposal generation time. It applies a $3\$\times 3 sliding window and anchor boxes on feature maps from the feature extraction network to output 512-dimensional features, which are then fed to two sub-fully-connected layers — a classification layer and a boundi$ pixels, respectively. Clearly, PVANet' s original scales are too large for our small objects, resulting in poor accuracy when directly applied. The latest PVANet version uses 6 scales (32, 48, 80, 144, 256, 512) and 7 aspect ratios (0.333, 0.5, 0.667, 1.0, 1.5, 2.0, 3.0) to form 42 anchors [**?**]. While this increases the number of anchors to expand detection range and improves mAP on PASCAL VOC by nearly 3% compared to the initial version, these anchor sizes

vary too widely, with even the smallest being larger than our small objects' average size.

The target sizes in our constructed dataset do not vary dramatically, particularly for small objects where the scale difference between persons and bicycles does not exceed 20 pixels. Therefore, we reduced the number and size of anchor scales while maintaining as many aspect ratios as possible to precisely locate small objects, since person and bicycle bounding boxes are primarily rectangular.

Based on this analysis, we selected 24 anchors per sliding window position, comprising 4 scales (16, 24, 32, 64) and 6 aspect ratios (0.333, 0.5, 0.667, 1, 1.5, 2). The RPN structure is shown in Figure 4 [Figure 4: see original paper].

Detection results comparing our method with the original PVANet are shown in Figure 5 [Figure 5: see original paper]. Figures 5(a) and 5(b) present detection examples under two viewing angles and different illumination conditions, with target categories and confidence scores displayed on the bounding boxes. The original PVANet produces many false positives that misclassify background as targets, particularly when objects occlude each other, as shown in the left subfigure of (a). Additionally, truncated targets are easily missed due to insufficient information, as shown in the left subfigure of (b). Our improved PVANet generates higher-quality small object proposals, enabling more accurate localization, effectively resisting interference from mutual occlusion (producing fewer false positives), and correctly detecting truncated targets, as demonstrated in the right subfigures of (a) and (b).

2.2.2 Selecting New Learning Rate Strategies

The learning rate is a crucial hyperparameter in deep learning that guides weight adjustments through the loss function gradient. Better learning rate strategies generally enable training superior network models in less time, making learning rate adjustment an important means of enhancing model performance during training.

PVANet employs a "plateau" strategy [**?**] to dynamically control the learning rate. This strategy monitors the average loss function variation and reduces the learning rate by a constant factor when improvement falls below a threshold during an iteration period, indicating the loss is on a "plateau." However, when we initially trained our model using the "plateau" strategy with 100,000 iterations, the learning rate remained constant at the initial value of 0.001. The primary reason is that target regions are extremely small compared to background regions, resulting in a large negative sample space and slow model convergence. Consequently, the "plateau" strategy, which changes the learning rate based on evaluating the dynamic mean of the loss function, struggles to adapt the learning rate effectively. Alternative strategies are needed to change the learning rate and accelerate convergence.

Observing the loss curve, we found it plateaued after 50,000 iterations, with

detection accuracy improving very slowly thereafter. We hypothesized that by 50,000 iterations, the loss gradient had approached a "plateau" state, making it difficult for the loss to improve further. To help the loss escape this plateau, we adopted the "step" learning rate strategy [**?**], defined by:

$$\text{learningRate} = \text{base\_lr} \times \gamma^{\lfloor \text{iter/stepsize} \rfloor}$$

where learningRate is the current learning rate, base_{lr} is the initial learning rate, gamma and stepsize are parameters, and iter is the iteration number. The learning rate decreases when the iteration count reaches integer multiples of stepsize. We set the initial learning rate to 0.001 and reduced it to 0.0001 after 50,000 iterations. Testing after 100,000 iterations showed a 0.45% accuracy improvement over the "plateau" strategy.

While low learning rates ensure we don' t miss any minima, they also require more time for convergence, particularly when the loss function enters a plateau. Reference [**?**] suggests that difficulty in reducing loss primarily stems from saddle points rather than local minima in the error surface. Considering this, we experimented with another strategy– "inv" [**?**]–which dynamically changes the learning rate to accelerate convergence rather than uniformly reducing it like the "step" strategy. We set the initial learning rate to 0.001 with gamma = 0.0001 and power = 0.75. The "inv" learning rate formula is defined as:

$$\text{learningRate} = \text{base\_lr} \times (1 + \gamma \times \text{iter})^{-\text{power}}$$

where learningRate is the learning rate, base_{lr} is the initial learning rate, iter is the iteration number, and gamma and power are parameters. With 100,000 training iterations, the learning rate dynamically decreased to 0.00026 at the 50,000-iteration plateau and to 0.00017 after 100,000 iterations. Experimental results show that the "inv" strategy produces better network models than both "plateau" and "step" strategies, making it more suitable for our small object detection scenario.

2.3 Experimental Results and Analysis

We conducted experiments using a Quadro K6000 GPU. To evaluate our algorithm' s effectiveness on small object detection, we adopted Average Precision (AP) for individual categories and mean Average Precision (mAP) across all categories as evaluation metrics. AP is the most intuitive metric for evaluating single-category detection accuracy, while mAP represents the mean of all category APs and evaluates overall model performance. Table 2 compares the test accuracy and runtime (frames per second, FPS) of our method against mainstream detection algorithms including Faster R-CNN, SSD, YOLOv2, and PVANet on our constructed small object dataset.

1) Analysis of Different Detection Methods

Existing deep learning-based object detection algorithms can be broadly categorized into two types. The first type generates object proposals before performing classification and bounding box regression, represented by Faster R-CNN and PVANet. These algorithms localize objects well but have slower detection speeds. The second type directly predicts object confidence scores and bounding boxes in an end-to-end framework, represented by YOLO and SSD. These methods feature simple network structures and fast testing speeds but cannot determine object locations well, particularly showing poor accuracy for adjacent or nearby objects. Table 2 shows that PVANet and Faster R-CNN achieve higher detection accuracy on small objects than all versions of YOLOv2 and SSD. YOLOv2 is the fastest among all listed algorithms but has the lowest accuracy. SSD combines YOLO and Faster R-CNN concepts, using multi-scale prediction to improve accuracy. Our proposal generation method, which fully considers small object detection characteristics, significantly improves algorithm performance. While our method is slower than YOLOv2, it achieves substantially better accuracy than other methods while basically meeting real-time requirements, making it an effective small object detection approach overall.

2) Analysis of Different Training Strategies

The learning rate significantly impacts model convergence to local minima (i.e., achieving maximum accuracy). We first trained our network using PVANet' s "plateau" learning rate strategy, achieving 70.95% mAP—a 9.53% improvement over the original PVANet. However, in small object detection, the small target region proportion creates a large negative sample space, resulting in slow convergence. Additionally, PVANet' s architecture is deep and narrow (94 convolutional and fully connected layers), making the loss function prone to oscillation during training. The "plateau" strategy reduces the learning rate when loss variation falls below a threshold for a period, but when the loss oscillates with variation exceeding the threshold, the learning rate remains unchanged and the loss fails to converge further.

To improve accuracy, we employed the uniformly-changing "step" strategy and the dynamically-changing "inv"strategy. Both approaches continue reducing the learning rate as iterations increase, even when the loss oscillates, enabling further convergence. Table 2 shows these strategies improve accuracy by 0.45% and 1.14% over "plateau," respectively. The "step" strategy requires manually setting the iteration interval for learning rate reduction, while "inv" reduces the learning rate slightly each iteration, eliminating potential issues from manual interval setting. Results demonstrate that the "inv" dynamic learning rate strategy trains the optimal network model, improving accuracy by 0.69% over "step." Our method with "inv" achieves 72.09% mAP on the small object dataset—a 10.67% improvement over the original PVANet—demonstrating that dynamic learning rate changes play a crucial role in overcoming saddle points and improving accuracy.

3) Impact of GPU Performance on Runtime

Reference [**?**] reports that PVANet achieves 21.7 FPS on 1056$\$\times 640 images using an NVIDIA Titan X GPU. Our images. This discrepancy primarily results from the Titan X' s compute capability of 6.1 versus the K6000' s 3.5 [**?**]—essentially half the performance. Additionally, our test images contain more targets, which increases detection time. By generating high-quality small object proposals, our method achieves 10 FPS on the Quadro K6000 GPU—a 30% speedup over the original PVANet.

---

3 Conclusion

This paper improves the high-performance PVANet algorithm for real-time small object detection scenarios. By combining a high-quality small object proposal generation method with the RPN network and selecting appropriate learning rate strategies, we effectively address the challenges of small object detection arising from tiny target sizes and interference from truncation and occlusion. Experiments demonstrate that our method exhibits excellent robustness for small object detection, achieving a speed improvement of 41ms per image over the original PVANet on a Quadro K6000 GPU and meeting real-time detection requirements. Since our dataset consists of video images, future work will focus on further improving detection speed for video-based applications.

References

[1] Ren Shaoqing, He Kaiming, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks [C]//Proc of International Conference on Neural Information Processing Systems. Cambridge: MIT Press, 2015: 91-99.

[2] Liu Wei, Anguelov D, Erhan D, et al. SSD: single shot multibox detector [C]//Proc of European Conference on Computer Vision. New York: Springer, 2016: 21-37.

[3] Redmon J, Farhadi A. YOLO9000: better, faster, stronger [C]//Proc of IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE Press, 2017: 6517-6525.

[4] Everingham M, Van Gool L, Williams C K I, et al. The pascal visual object classes (voc) challenge [J]. International Journal of Computer Vision, 2010, 88(2): 303-338.

[5] Pham P, Nguyen D, Do T, et al. Evaluation of deep models for real-time small object detection [C]//Proc of International Conference on Neural Information Processing. New York: Springer, 2017: 516-526.

[6] Chen Chenyi, Liu Mingyu, Tuzel O, et al. R-CNN for small object detection [C]//Proc of Asian Conference on Computer Vision. New York: Springer, 2016: 214-230.

[7] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation [C]//Proc of IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE Press, 2014: 580-587.

[8] Girshick R. Fast R-CNN [C]//Proc of IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE Press, 2015: 1440-1448.

[9] Bell S, Lawrence Zitnick C, Bala K, et al. Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks [C]//Proc of IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE Press, 2016: 2874-2883.

[10] Eggert C, Winschel A, Zecha D, et al. Saliency-guided selective magnification for company logo detection [C]//Proc of the 23rd International Conference on Pattern Recognition. Piscataway, NJ: IEEE Press, 2016: 651-656.

[11] Zhang Liliang, Lin Liang, Liang Xiaodan, et al. Is Faster R-CNN doing well for pedestrian detection? [C]//Proc of European Conference on Computer Vision. New York: Springer, 2016: 443-457.

[12] Eggert C, Brehm S, Winschel A, et al. A closer look: small object detection in Faster R-CNN [C]//Proc of IEEE International Conference on Multimedia and Expo. Piscataway, NJ: IEEE Press, 2017: 421-426.

[13] Zhu Zhe, Liang Dun, Zhang Songhai, et al. Traffic-sign detection and classification in the wild [C]//Proc of IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE Press, 2016: 2110-2118.

[14] Sanghoon H, Roh B, Kim K H, et al. PVANET: deep but lightweight neural networks for real-time object detection [C]//Proc of International Conference on Neural Information Processing Systems. Cambridge: MIT Press, 2016: 1608-1614.

[15] Khosla D, Chen Yang, Kim K. A neuromorphic system for video object recognition [J]. Frontiers in Computational Neuroscience, 2014, 8(8): 1-14.

[16] Shang Wenling, Sohn K, Almeida D, et al. Understanding and improving convolutional neural networks via concatenated rectified linear units [C]//Proc of International Conference on Machine Learning. New York: ACM Press, 2016: 2217-2225.

[17] Szegedy C, Liu Wei, Jia Yangqing, et al. Going deeper with convolutions [C]//Proc of IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE Press, 2015: 1-9.

[18] Kong Tao, Yao Anbang, Chen Yurong, et al. Hypernet: towards accurate region proposal generation and joint object detection [C]//Proc of IEEE Con-

ference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE Press, 2016: 845-853.

[19] He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. Deep residual learning for image recognition [C]//Proc of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE Press, 2016: 770-778.

[20] Uijlings J R, Sande K E, Gevers T, et al. Selective search for object recognition [J]. International Journal of Computer Vision, 2013, 104(2): 154-171.

[21] Zitnick C L, Dollár P. Edge boxes: locating object proposals from edges [C]//Proc of European Conference on Computer Vision. New York: Springer, 2014: 391-405.

[22] Jia Yangqing, Shelhamer E, Donahue J, et al. Caffe: convolutional architecture for fast feature embedding [C]//Proc of the 22nd ACM international conference on Multimedia. New York: ACM Press, 2014: 675-678.

[23] Smith L N. Cyclical learning rates for training neural networks [C]//Proc of IEEE Winter Conference on Applications of Computer Vision. Piscataway, NJ: IEEE Press, 2017: 464-472.

[24] CUDA GPUs [EB/OL]. [2018-05-03]. https://developer.nvidia.com/cuda-gpus.

Note: Figure translations are in progress. See original paper for figures.