SCIENCE CHINA
Information Sciences

• LETTER •

January 2023, Vol. 66 119102:1–119102:2
https://doi.org/10.1007/s11432-021-3504-4

TinyDet: accurately detecting small objects within 1 GFLOPs

Shaoyu CHEN[1], Tianheng CHENG[1], Jiemin FANG[2,1], Qian ZHANG[3], Yuan LI[4], Wenyu LIU[1] & Xinggang WANG[1*]

[1]*School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan* 430074*, China;*
[2]*Institute of Artificial Intelligence, Huazhong University of Science and Technology, Wuhan* 430074*, China;*
[3]*Horizon Robotics Inc., Beijing* 100080*, China;*
[4]*Google Inc., Mountain View* CA 94043*, USA*

Received 22 July 2021/Revised 12 February 2022/Accepted 22 April 2022/Published online 4 November 2022

Citation Chen S Y, Cheng T H, Fang J M, et al. TinyDet: accurately detecting small objects within 1 GFLOPs. Sci China Inf Sci, 2023, 66(1): 119102, https://doi.org/10.1007/s11432-021-3504-4

Dear editor,

Object detection plays an important role in computer vision, and has gradually become the technical foundation of many applications. However, the inference procedure of advanced detection models requires massive computational resources, which makes it difficult to apply to resource-constrained mobile or edge devices. To widely apply artificial intelligence (AI), "Tiny AI" models that are both computation-efficient and energy-efficient are becoming more and more popular. In this study, we target at boosting small object detection performance in lightweight detectors. Based on the good practices of designing lightweight networks, we propose TinyDet, which is a two-stage detector with high-resolution (HR) feature maps for dense anchoring and open-sourced at the website[1)]. HR feature maps improve small object detection ability but also bring much more computation cost. To ease the contradiction between resolution and computation, we propose TinyFPN and TinyRPN by introducing a sparsely-connected convolution (SCConv) module, which keeps both high resolution and low computation. Besides, we observe that severe feature misalignment exists in lightweight detectors. Small objects are much more sensitive to such positional misalignment. By eliminating the misalignment, the detection performance of small objects is significantly improved.

Feature alignment. Convolution with stride 2 is widely used to reduce the resolution of the feature maps. However, strided convolutions on even pixels may lead to feature misalignment. Assume that the kernel size is k, the padding size is $\frac{k-1}{2}$, and the feature map is with stride s compared with the input image. As shown in Figure 1(a), in the strided convolution, one padding pixel is dropped, and the feature center is shifted by 0.5 pixels. When mapped to the input image, the misalignment is $\frac{s}{2}$ pixels. The feature misalignment affects subsequent layers and accumulates layer by layer. In TinyDet, six strided convolution layers exist in the backbone network and cause misalignment of $0.5, 1, 2, 4, 8, 16$ pixels, respectively. In FPN and RPN, the accumulated misalignment is as large as 31.5 pixels. The misalignment is negligible for large models with a high input resolution, but significant for lightweight detectors with a low resolution. Assuming the input resolution is 320×320, the misalignment proportion is up to $\frac{31.5}{320} \approx 9.8\%$, which leads to severe mismatching between features and their spatial positions.

To alleviate the feature misalignment, in TinyDet, we adopt an average pooling layer before each strided convolution. As illustrated in Figure 1(b), the average pooling operation converts even pixels to odd ones and avoids asymmetry in the strided convolution and corrects the misalignment. And when optimizing the network, the adjacent convolution layer and average pooling layer can be fused into a single layer for better inference efficiency.

We adopt effective receptive field (ERF) maps [1] to visualize the misalignment of feature maps. As shown in Figure 1(a), we can observe that ERFs obviously deviate from the geometric centers of corresponding anchors. By the average pooling layer, the misalignment is eliminated (Figure 1(b)). Small objects are much more sensitive to such positional misalignment. By eliminating the misalignment, the detection performance of small objects is significantly improved.

High resolution feature maps and computation reduction. Owing to the computation cost limitation, previous lightweight detectors usually adopt feature maps with low resolutions for detection (19×19 in Pelee [2], 20×20 in ThunderNet [3]). However, small feature maps have low spatial resolution. Low-resolution feature maps cannot provide spatially matched features for objects located in arbitrary

* Corresponding author (email: xgwang@hust.edu.cn)

1) https://github.com/hustvl/TinyDet.

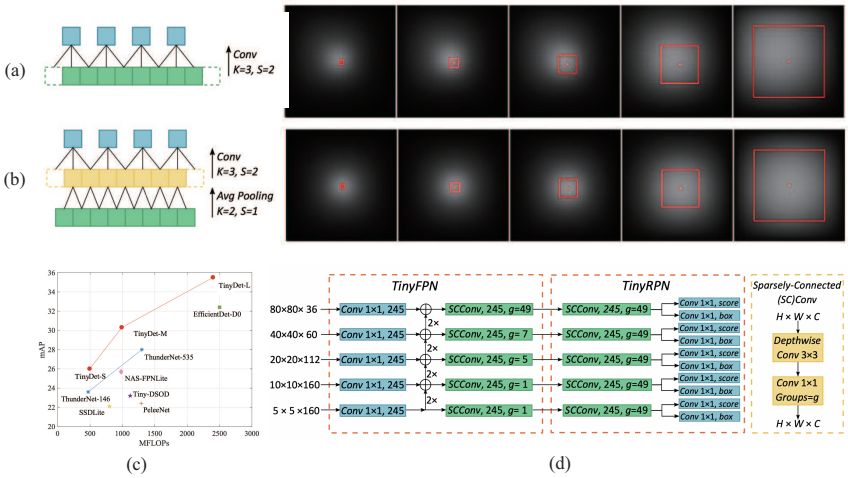

Figure 1 (Color online) (a) Feature misalignment; (b) feature alignment; (c) model FLOPs vs. mAP on the COCO test-dev2017 set. TinyDet achieves higher mAP with less computation cost compared with other detectors. (d) The structure of TinyFPN and TinyRPN.

positions, especially for small objects.

In TinyDet, we enable object detection on high-resolution feature maps. We fetch five feature maps from the backbone for detection, respectively with strides 4, 8, 16, 32, and 64. Note that the resolution of the feature map with stride 4 is 80×80, which is the highest resolution feature map used in lightweight detectors.

Because of the high-resolution design, the computation budget of the detection part becomes extremely high. We adopt the SCConv module, specialized for both efficiency and high resolution in FPN and RPN. As shown in Figure 1(d), the SCConv is a combination of a depth-wise convolution and a point-wise group convolution. Compared with the vanilla depth-wise separable convolution, SCConv further reduces the connections among channels.

Based on SCConv, we propose TinyFPN and TinyRPN (Figure 1(d)). In TinyFPN, SCConv is applied after feature fusion, in place of the normal 3×3 convolutions. We set larger group numbers, i.e., sparser connections, for SCConvs in the high-resolution pyramid levels to reduce the computation cost. TinyRPN consists of an SCConv and two sibling 1×1 convolutions for classification and regression respectively. The parameters of TinyRPN are shared across all pyramid levels.

Experiment. We implement our TinyDet based on the MMDetection toolbox. Our models are trained with batch size 128 on 4 GPUs (32 images per GPU) for 240 epochs. The SGD optimizer is used with momentum 0.9 and weight decay $1\mathrm{E}-5$. We linearly increase the learning rate from 0 to 0.35 in the first 500 iterations and then decay it to $1\mathrm{E}-5$ using the cosine anneal schedule. We detail the experiment in Appendix A and report the average precision (AP) under different IoUs for detecting objects in different scales.

We compare our TinyDet with other lightweight detectors on the COCO test-dev2017 set in Figure 1(c). The results show that our models obviously outperform them with fewer computation costs: TinyDet-S surpasses ThunderNet-SNet146 by 2.4% AP with similar FLOPs; TinyDet-M surpasses ThunderNet-SNet535 by 2.3% AP with 76% FLOPs; and TinyDet-L surpasses EfficientDet-D0 by 3.1% AP with similar FLOPs. And TinyDet has better performance-computation trade-offs than the automatically searched lightweight detector (i.e., NAS-FPNLite [4]).

Besides, TinyDet models obtain extraordinary performance in detecting small objects. With similar computation cost (i.e., FLOPs), TinyDet-M achieves 13.5 AP^s, which is over 100% improvement over ThunderNet-SNet535 [3] with 6.5 AP^s; and TinyDet-S achieves 9.6 AP^s, which is also over 100% improvement over ThunderNet-SNet146 with 4.6 AP^s. TinyDet-L achieves the same 18.3 AP^s with YOLOv3, but it only has 1/30 computation cost of YOLOv3.

We deploy TinyDet on two types of ARM CPUs using Pytorch Mobile with single thread and without any optimization. On Snapdragon 865 CPU, the inference time of TinyDet-S/M/L is 103/179/312 ms. On Kirin 820 CPU, the inference time of TinyDet-S/M/L is 133/236/386 ms.

Conclusion. In this study, we propose a series of lightweight detectors named TinyDet. TinyDet is with good performance-computation trade-offs (30.3 mAP with only 991 MFLOPs) and applicable to resource-constrained mobile or edge devices. Besides, TinyDet is superior to other lightweight detectors in small object detection.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant Nos. 61876212, 61733007), Zhejiang Laboratory (Grant No. 2019NB0AB02), and HUST-Horizon Computer Vision Research Center.

Supporting information Appendixes A and B. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Luo W, Li Y, Urtasun R, et al. Understanding the effective receptive field in deep convolutional neural networks. In: Proceedings of International Conference on Neural Information Processing Systems, 2016. 4905–4913
2 Wang J, Bohn T A, Ling C X. Pelee: a real-time object detection system on mobile devices. In: Proceedings of International Conference on Neural Information Processing Systems, 2018
3 Qin Z, Li Z, Zhang Z, et al. ThunderNet: towards real-time generic object detection on mobile devices. In: Proceedings of IEEE International Conference on Computer Vision, 2019
4 Ghiasi G, Lin T, Le Q V. NAS-FPN: learning scalable feature pyramid architecture for object detection. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019