Image Processing: Facilitating Retinanet for Detecting Small Objects

Zihao Zhou, Yang Li, Chenglei Peng, Hanrong Wang and Sidan Du*
School of Electronics, Science and Engineering, Nanjing University

*Corresponding author email: coff128@nju.edu.cn

Abstract. Detecting small objects is a challenging task in object detection due to low spatial resolution and interference by background. Specifically, one-stage detectors struggle with small objects for they generate worse candidate bounding boxes. In this paper, several modifications are made to the original Retinanet to tackle the problem. Dilated convolutional layers are added to the backbone to get fined-grained features along with semantic information. The gradient of loss function is increased near the origin to enhance the quality of candidate boxes for small objects. A novel feature fusion method is also proposed to directly guide low-level features with semantic information. Significant improvement of 5.1 mAP can be seen when evaluating on MOCOD small object dataset, which contains a large amount of small objects. Our method can be easily migrated to other backbone networks with feature pyramids for detecting small objects.

Keywords: Small object detection; Retinanet; One-stage detector; Backbone network; Feature fusion; Loss function.

1. Introduction
Fundamental progress has been made to the area of object detection in the past few years and it is useful in many scenarios. Mainstream object detectors use a two-stage framework [1], which generate certain number of candidate proposals of the original image. These proposals are fed into a neural network for classification and bounding box regression. There are also multi-stage detectors like [2], which train multiple cascaded detectors using different IoU thresholds. Bounding boxes are filtered according to the current IoU threshold, then fed into the next stage. Meanwhile, one-stage object detection is becoming more popular mostly because its speed. One-stage detectors usually frame the whole network as a regression problem, predicting object location and category simultaneously.

However, small object detection remains a difficult problem, compared to big object detection. In general, multi-stage object detectors and one-stage detectors can detect big objects with almost the same accuracy, while multi-stage object detectors outperform one-stage object detectors to a large margin, when detecting small objects. There are some plausible explanations for the performance gap between one-stage and two-stage detectors. First, one-stage detectors fail to handle class imbalance problem, most candidate bounding box generated by network are negative examples which have little IoU with ground-truth boxes. The gradient of positive examples is unable to take effect in large amount of negative boxes. Second, most candidate bounding boxes are easy examples, which are of little use for parametric convergence. In two-stage detectors, however, most background samples are filtered to obtain foreground and background balance. Particularly, due to the spatial resolution of small objects, even candidate boxes assigned positive labels share low IoU with relative ground truth boxes. Under this circumstance, getting qualified candidate boxes is more important.
In this paper, our main objective is to enhance the performance of small object detection, based on the famous one-stage pipeline of Retinanet [3]. Several adaptations are made to it, focusing on changing the network backbone, feature fusion method and the loss function of original implementation, and considerable improvement can be seen on small object detection performance. These methods are tested on ICIG 2019 MOCOD small object detection dataset and see large improvement on detection accuracy, compared with the basic implementation.

2. Related Work
The methods addressing object detection are generally divided into two categories: two-stage approaches and one-stage approaches. Many methods and novel techniques have been proposed to handle with small objects. Faster-RCNN [1], a famous two-stage detector, views the whole detection problem as a multi-task optimization, reducing classification error and bounding box regression error at the same time. The Region Proposal Network generates a fixed set of anchors by sliding window on the deepest feature map, and the anchors are regressed towards ground truth boxes. These anchors cannot provide appropriate reception field for small objects and fine-grained texture information are lost after several steps of down sampling. To handle this, MS-CNN [8], predicts bounding boxes on different network layers, each focusing on certain object scales, and predictions on shallower layers have smaller reception fields. YOLO [15], abbreviation for “You Only Look Once”, uses a unified loss function to train an end-to-end framework. It is fast but cannot localize objects well and struggles with small objects. SSD [16], abbreviation for “Single Shot Detect box”, views the whole detection procedure as a regression problem like YOLO, however uses feature maps of different depths to generate multi-scale candidate bounding boxes, and significantly improves performance of small object detection. DSOD [17] applies dense connections to the base network of SSD and fuse multiscale features by concatenating all bottleneck layer outputs. FPN [5], abbreviation for “Feature Pyramid Network”, makes the first attempt to combine low-level features with high-level information by using feature pyramids and lateral connections. It performs detection on multiple layers similarly. SNIP [14], known as “Scale Normalization for Image Pyramids”, aims to get multi-scale information as well as alleviate domain shift. It build feature pyramids and only back-propagate the gradient of bounding boxes within certain scale range during training. Retinanet [3] introduces focal-loss, which sees to the imbalance between positive and negative samples, as well as easy and hard samples, improving the quality of bounding boxes largely. There are also anchor-free detectors such as CornerNet [18] and FSAF [19]. CornerNet is a one-stage object detector. It predicts the top-left coordinate and the bottom-right coordinate for individual bounding boxes and uses an Hourglass network, which captures local and global information simultaneously, enabling the network to get high-quality boxes. Another single-shot object detector, FSAF [19], abbreviation for “Feature Selective Anchor-Free”, allows each instance to choose the optimal feature layer for bounding box regression.

3. Method
3.1. Network Backbone
The keras implementation of Retinanet [3] was employed as our baseline network. For better performance in small object detection, the network structure is modified as well as adding several useful features.

Inspired by [4], several changes are adapted to the default base network, resnet-50 [6], which contains 4 convolutional blocks in total, and each block down-samples the feature map from the last block by pooling 2x or stride 2 convolution. In the original implementation, the last three of the four blocks are used to build the feature pyramid, the spatial size of each 8x, 16x, 32x, respectively. In order to detect large objects, larger stride is applied to construct even deeper feature maps in the feature pyramid, the spatial size of each 64x, 128x respectively, as shown in Fig. 1.

However, there are two major defects of too deep feature map. First, deep feature maps struggle with object localization. Little offset on the deep feature map leads to large localization error, when mapped to the input image. Second, deep feature maps are harmful to small objects. The information of small objects tend to lose as the spatial resolution deteriorates. Though Feature Pyramid Network [5] uses
early layers to predict small objects, lack of semantic information makes it difficult for the network to classify the object category. Therefore, the blocks whose output feature maps are more than 8x down-sampled are replaced with new bottleneck blocks. Instead of using stride-2 convolution layers, dilated convolution layers are applied like in [4]. In this way, the stage 1, 2, 3 are maintained like in original resnet-50 backbone, the spatial resolution remains 8x down-sampled after stage 4, as plotted in figure 1.

These bottleneck blocks are used to enlarge reception field while maintaining high spatial resolution. Feature maps are not further down-sampled because the dataset only consists of small objects, according to the ratio of area of ground truth boxes to the area of original images. Feature maps in stage 4 and 5 have already contained enough semantic information for the task of classification, while large spatial resolution is good for bounding box localization.

Similarly as in [3], stage 2-5 are used to construct feature pyramids with lateral connections p2-p5, however, the down-sampling pyramids p6 and p7 are removed, which are specialized for large objects. Each pyramid level is attached to a classification model and a regression model comparatively, then the outputs of all pyramids are combined and fed to other operations, such as none maximum suppression, etc.

![Figure 1. Modifications to the base retinanet framework, including dilated convolutional backbone and a novel feature fusion method](image)

3.2. Feature Fusion

To further fuse high-level semantic features with low-level features, bottom-up path augmentation is employed like in [5]. As is mentioned above, low level features such as edge and shape features capture many fine details and are important for the task of localization. Top-down and skip-connections are used like in [9]. Since the employed backbone network are all deep networks that have hundreds of layers, fine-grained information tend to lose after propagation of low-level features through the network. As is argued in [7], this kind of top-bottom method only pay more attention to adjacent spatial resolution. This leads to the dilution of semantic information of none-adjacent feature maps during feature fusion process, which is an imbalanced design.

Inspired by concurrent connections in HRNet [25], “HR” for “High Resolution”, another bottom-up feature pyramid to obtain low-level features guided with semantic information. First, another convolutional block is added following our dilated bottleneck layer in 2.1, which extract features with most semantic information. Then feature pyramids P2-P4 are facilitated with direct high-level guidance.
In original feature hierarchy $P_2-P_5$, a feature map only fuse with an adjacent one. Low-level feature maps are manually merged with deepest layer outputs additionally, as is shown in Fig 1.

The deepest feature layer $C_5$ is deconvoluted to be the same shape as the target layer in feature hierarchy $P_2-P_4$ then each feature map $P_i$ is concatenated with the up-sampled feature map $C_5$ as is denoted $P'_i$.

Before concatenation, a l2 normalization layer is added to keep values from different layers on the same scale. After that, 1x1 convolution is used to resize the feature map to demanded channels. Feature pyramids $\{P'_2, P'_3, P'_4, P'_5\}$ are attached to a classification model and a bounding box regression model respectively. It is believed that high-level guidance passed from deep layers of the network help with classification accuracy, especially for small objects, when low-level features are similar among classes.

3.3. Loss Function

In [3], Lin et al. bring forward a novel loss function, focal loss, which meticulously addresses the problem of class imbalance between background and foreground in one-stage detection. Besides, smooth l1 loss [7], is applied for bounding box regression. The loss function is also changed for small object detection.

ASL1-loss [21] is used for authentic smooth-l1 loss, instead of smooth l1 loss, for bounding box regression, see formula 1. The loss function was originally proposed for its universally continuous gradient in order to perform normalization. The function is only used for its good properties, without gradient-harmonizing operations. Object detection is a multi-task learning problem and it is a subtle job to balance between different tasks. Samples with regression loss more than 1.0 are called outliers and those with regression loss less than 1.0 inliers. Pang et al. [22] think that, as bounding box regression is an unbound task, increasing regression loss directly leads to the network more sensitive to outliers, which are often considered as noise. Pang et al. use balanced-l1 loss to tackle the problem by giving more gradient to inliers, balancing the gradient propagation of all samples. Small object localization benefit from balanced-l1 loss in that most localization errors are little deviation from assigned ground truth box and increased inlier gradient is helpful for the training process. Despite this, the functional relationship is too complex and there are three hyper-parameters, which requires careful tuning. ASL1-loss has a relatively simple form. As is shown in Fig. 2, the overall gradient of inliers is also increased, compared to smooth-l1 loss and loss function with different parameter $m$ are compared. In addition, it is able to distinguish between gradient of outliers, while all gradient of outliers is equal to 1. The parameter $m$ is used to control the steepness of the derivative function near the origin. Further, all the degrees of derivatives of ASL1-loss are existed and continuous, see Formula 2. The range of gradient falls in [0,1), which is insensitive to outliers. Experiments with different values of $m$ are performed, which will be discussed later.

\[ L_{reg}(x) = \sqrt{x^2 + m^2} - m \]

\[ \frac{\partial L_{reg}(x)}{\partial x} = \frac{x}{\sqrt{x^2 + m^2}} \]
Figure 2. Gradient of loss function with different parameter m compared to smooth-l1 loss

4. Experiments
The performance of our network is evaluated on 2019 MOCOD small target dataset. All our models are trained on NVidia GTX 1080ti GPU. Ablation studies is performed for different modules of our framework and mean Average Precision is used as our evaluation metric.

Figure 3. Scale range of instances in MOCOD datasets.
4.1. MOCOD Small Object Dataset

This dataset is built under virtual environment, simulating drones photographing multiple vehicles on the road. Detecting targets include 12 categories of vehicles, with 2 weather conditions (sunny and foggy) and 2 illumination conditions (day and dusk). The dataset consists of over 13,000 synthetic images of vehicles on 5 city routes in total. Some vehicles are occluded by trees or buildings, however, no special annotations are made for these detection targets. The first 4 routes are used for training and validation (about 10,000 images) and the last route for testing (about 3,000 images).

Brief statistics are made on the annotation files and find that the average size of an object is less than 30×30 pixels, see Fig.3. In MS COCO [10] dataset, if an instance bounding box is less than 32×32 in area, it is deemed as a small object, most objects in this dataset are small objects according to this definition. In order to detect small objects more accurately, the image is not resized, the spatial resolution of input images remain 2048×1080.

All of our networks are trained from scratch in the following reasons. First, no optimal pre-trained models. Almost all objects in this dataset are small objects and the object categories are limited to vehicles. All object instances are from synthetic images and there are no similar real-life datasets for pre-training. The task of pre-trained models on ImageNet is classification, which is not optimal for detection. Weights can be loaded from specific layers of a pre-trained model and conduct transfer learning, however, it demands painstaking manipulation [23]. Second, the number of instances is large enough for training from scratch. There are 50 instances per image in average. This is much larger than Pascal VOC [10], which often requires ImageNet-pretrained model for better convergence. In addition, several augmentations are performed on images before they are fed into the network, decreasing the chance of overfitting. Third, normalization layers are applied for gradient optimization. Batch normalization [11] and group normalization [12] are applied in our work comparatively, which help model convergence and allow us to initiate the training process with a larger learning rate.

Table 1. Evaluation results on MOCOD dataset with different hyper parameter m in our loss function.

| m    | 0.01 | 0.02 | 0.03 | 0.05 |
|------|------|------|------|------|
| mAP(%) | 61.4 | 61.9 | 62.2 | 61.1 |

Table 2. Ablation studies on MOCOD dataset. From left to right: with resnet-50 backbone and no additional features; with our backbone and no additional features; with our backbone and ASL1 loss; with our backbone, ASL1 loss and our feature fusion method.

|                             | ×    | √    | √    | √    |
|-----------------------------|------|------|------|------|
| our backbone network        |      |      |      |      |
| modified loss function      | ×    | ×    | √    | √    |
| feature fusion method       | ×    | ×    | ×    | √    |
| MOCOD small object dataset mAP(%) | 60.9 | 62.5 | 64.4 | 66.0 |

4.2. Base Network

Our modification to the original resnet-50 network can be easily adapted to other residual networks with deeper and wider layers such as resnet-101 or resneXt-101 [24] to build feature pyramids. Basically, the stride-2 convolutional layer of the first block of stage 4 and 5 in resnet-50 is removed, and is replaced with dilated convolutional layers. In this way, the spatial resolution of feature maps remain 8x down-sampling after stage 4. Choosing resnet-50 as our base network is a compromise between speed and accuracy. It is considered that with the growth of model complexity the final mAP of our model will somehow increase. Due to our limited computing resources, only mAP increase to the baseline is reported. Anchors [1] are applied for region proposals extraction, however, default anchor parameters are changed for our base network. Anchor size are set to 8, 16, 32, 64 on separate feature pyramids to fit instance size in the dataset. The original settings of 3 anchor scales and 3 aspect ratios are followed, and each sliding position generates 9 anchors. The anchor scale ranges from 8 pixels to 102 pixels and that covers the size of all instances in the dataset.
4.3. Feature Fusion
As is discussed in 2.2, FPN [5] is employed to build lateral connections with element-wise feature map adding and a convolutional layer is simply used with kernel size 3 to remove aliasing effect. After that, the deepest feature map with low-level feature maps is concatenated to guide small object detection with high-level semantic information. A convolutional layer with kernel size 1 is added to resize the feature map to demanded channels. This method can be easily applied to other detection technique with top-down and lateral-connected feature pyramids.

4.4. Loss Function
Class-balanced focal loss is used for classification and ASL1 loss for localization. $\alpha=0.25$ and $\gamma=2$ are used for focal loss, which is experimented to have the best performance in [3]. Different values of parameter $m$ are compared on MOCOD dataset with no additional features, see Table 1. First its value is set to 0.02, as the author suggests in [21]. The value is finally chosen as 0.03 for ASL1 loss, which is tested to have the highest mAP. It is presumed that with a smaller value of $m$, the gradient of small candidate bounding boxes is increased and this is good for small object detection. However, when $m$ is too small, the loss function is close to l1 normalization, whose derivative function is always one, which lacks the ability to distinguish rather small deviations. This leads to the loss fluctuating around the stable value and difficult to converge afterwards.

4.5. Other Features
Several state-of-the-art modules are applied to improve our network’s performance. As all inputs are high-resolution images, batch size is simply set equal to one due to our limited computing resources. As batch normalization [11] layers compute the mean value and variance of training samples in a single batch, performance drops and the model becomes unstable with smaller batch size. Group normalization [12] is used instead, which avoid a small batch size deteriorating our detection model. Also, soft-nms [13] is used instead of nms (non-maximum suppression) to preserve more potential bounding box predictions.

4.6. Ablation Studies
To analyse the importance of each component in our network, ablation studies are performed as in table 2. Learning rate is set to 1e-4 with Adam optimizer during training. The training is 10k iterations per epoch. mAP is computed on the test set after each epoch and the training is stopped after 140k iterations. The best mAP of all epochs is reported during training each models. Training speed is about 1.1 s per iteration on an NVidia GTX1080ti GPU.

5. Conclusions
Small object detection requires high spatial resolution to maintain fine-grained details and high-quality candidate bounding boxes. In this paper, several modules are proposed to address these problems, based on the one-stage object detector, Retinanet, specialized for small objects. First, dilated convolutional blocks are added to the original resnet. This is proved to be good for bounding box localization for small objects. Second, a novel feature fusion method is proposed, which is able to directly guide low-level features with semantic information. Third, ASL1-loss is used for bounding box regression. This increases the gradient of loss function near the origin and enhances the quality of candidate boxes for small objects. Our method can be easily migrated to other backbone networks with feature pyramids for detecting small objects. Significant improvement of 5.1 mAP can be seen on MOCOD small object dataset with our proposed components.

References
[1] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 39(6).
[2] Cai Z, Vasconcelos N. Cascade R-CNN: Delving Into High Quality Object Detection[C]// 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2018.
[3] Lin T Y, Goyal P, Girshick R, et al. Focal Loss for Dense Object Detection[C]// 2017 IEEE International Conference on Computer Vision (ICCV). IEEE, 2017:2999-3007.
[4] Li Z, Peng C, Yu G, et al. DetNet: A Backbone network for Object Detection[J]. 2018.
[5] Lin T Y, Dollar P, Girshick R, et al. Feature Pyramid Networks for Object Detection[C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society, 2017.
[6] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[C]// 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2016.
[7] Girshick R. Fast R-CNN[J]. Computer ence, 2015.
[8] Cai Z, Fan Q, Feris R S, et al. A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection[J]. 2016.
[9] Liu S, Qi L, Qin H, et al. Path Aggregation Network for Instance Segmentation[J]. 2018.
[10] Lin T Y, Maire M, Belongie S, et al. Microsoft COCO: Common Objects in Context[J]. 2014.
[11] Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift[J]. 2015.
[12] Wu Y, He K. Group Normalization[J]. International Journal of Computer Vision, 2018.
[13] Bodla N, Singh B, Chellappa R, et al. Soft-NMS -- Improving Object Detection With One Line of Code[J]. 2017.
[14] Singh, B., Davis, L.S.: An analysis of scale invariance in object detection-SNIP.arXiv preprint arXiv:1711.08189. 2017.
[15] Redmon J, Divvala S, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection[C]// Computer Vision & Pattern Recognition. IEEE, 2016.
[16] Liu W, Anguelov D, Erhan D, et al. SSD: Single Shot MultiBox Detector[C]// European Conference on Computer Vision. Springer International Publishing, 2016.
[17] Shen Z, Liu Z, Li J, et al. DSOD: Learning Deeply Supervised Object Detectors from Scratch[J]. 2017.
[18] Law H, Deng J. CornerNet: Detecting Objects as Paired Keypoints[J]. International Journal of Computer Vision, 2020, 128(3):642-656.
[19] Zhu C, He Y, Savvides M. Feature Selective Anchor-Free Module for Single-Shot Object Detection[C]// 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2019.
[20] Cui, Yin, Jia, et al. Class-Balanced Loss Based on Effective Number of Samples. arXiv preprint arXiv:1901.05555. 2019.
[21] Li B, Liu Y, Wang X. Gradient Harmonized Single-stage Detector[J]. 2018.
[22] Pang J, Chen K, Shi J, et al. Libra R-CNN: Towards Balanced Learning for Object Detection[C]// 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020.
[23] He K, Girshick R, Dollar P. Rethinking ImageNet Pre-Training[C]// International Conference on Computer Vision. 0.
[24] Xie S, Girshick R, Dollár, Piotr, et al. Aggregated Residual Transformations for Deep Neural Networks[C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2016.
[25] Sun K, Xiao B, Liu D, et al. Deep High-Resolution Representation Learning for Human Pose Estimation[C]// 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR). IEEE, 2019.