SFGNet detecting objects via spatial fine-grained feature and enhanced RPN with spatial context

Jun Hu,a,b, Yongfeng Wangb, Shuai Chengb, Jiaxin Liu,a,b, Jiawen Kanga,b and Wenxing Yanga,b

aSchool of Computer Science and Engineering, Northeastern University, Liaoning, People’s Republic of China; bNeusoft Reachauto Corporation, Liaoning, People’s Republic of China

ABSTRACT
Object detection, which is one of the most fundamental visual recognition tasks, has been a hotspot in computer vision. CNN (Convolutional Neural Networks) have been widely employed for building detector. Due to the success of RPN (Region Proposal Network), the two-stage detectors get both classification accuracy and precise regression bounding boxes. However, they still struggle in small-size object detection. In this paper, we present a deep network, namely Spatial Fine-Grained Network (SFGN). The SFGN that exploits Spatial Fine-Grained Features (SFGF) concatenates the higher resolution features, which is fine-grained with the low resolution features and high-level semantic by stacking spatial features for fine-grained features. An enhanced region proposal generator is proposed to get the object less for small object to obtain a small set of proposal. The contextual information surrounding the region of interest is embedded using local spatial information for increasing the useful information and discriminating the background. For improving the detection performance, we use a simple yet surprisingly effective online hard example mining (OHEM) algorithm for training region proposal generator. It embeds an efficiently implemented soft non-maximum suppression (soft-NMS) for replacing with tradition NMS to obtain consistent improvements without increasing the computational complexity in inference. On PASCAL VOC 2007 and PASCAL VOC 2012 datasets, our SFGN improves baseline model from 81.2% mAP to 80.6% mAP. On MS COCO dataset, SFGN also performs better than baseline model. As intuition suggests, our detection results provide strong evidence that our SFGN improves detection accuracy, especially in small object test.

1. Introduction
Object detection is a composite task which has at least two sub-tasks, recognizing image categories and predicting each object by a bounding box. Before the success of CNN, the widely used detection systems are based on the combination of independent components: HOG (Dalal & Triggs, 2005) and SIFT (Lowe, 2004 et al.). The DPM (Felzenszwalb et al., 2010) and its variants (Azizpour & Laptev, 2012; Dollár et al., 2014) have been the dominant methods for years. These methods use object component descriptors as features and sweep through the entire image to find regions with a class-specific maximum response. The basic idea of object detector is to divide the task pipeline into three parts: (1) feature extraction; (2) proposing ROI (regions of interest); (3) region classification.

After the great success of CNN in image classification, CNN have been widely combined into detector pipeline as one of above three parts for improving detector generalization and robustness. Many strong CNN structure have been applied as feature extractor, also called backbone, such as VGG (Simonyan & Zisserman, 2014), ResNet (He et al., 2016), Mobilenet (Howard et al., 2017) and EfficientNet (Tan & Le, 2019). For the ROI generation part, generic object detection methods are moving from dense sliding window approaches to region-based proposal framework and region-free approaches.

The region-free approaches, also called one-stage detector, remove the ROI proposal part and frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities, such as YOLO (Redmon et al., 2015), YOLOv2 (Redmon & Farhadi, 2016) and SSD (Liu, Anguelov, et al., 2015). YOLO is the pioneering work that a single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. Every cell in its feature map gives a prediction which covers the area surrounding the centre of cell. It removes the ROI proposal part and applies a sliding window for each cell on feature maps. Inspired by this work, SSD generates a set of anchors with
multiple scales and aspect ratios in each cell on feature maps. SSD produces a set of predictions of different scales from feature maps of different scales on a single grid cell and explicitly separates predictions by aspect ratio, even with relatively low-resolution input images. A well-designed anchor with multiple prior boxes replaces the ROI proposal part in traditional object detectors. Later one-stage detectors make further improvement. RetinaNet (Lin et al., 2018) addresses the sample imbalance between foreground and background. YOLOv2 defines a better anchor by k-mean clustering. CornerNet (Law & Deng, 2018) detects object as a pair of corners instead of manually designed anchors. However, without ROI proposal, both classification accuracy and location precision are hurt.

Another object detection family builds ROI proposal part with CNN, namely two-stage detector. High-quality and category-independent object proposals reduce the number of windows each classifier needs to consider, thus promoting the development of object detection. Two-stage object detection methods adopt such pipeline, R-CNN (Girshick et al., 2013), SPPnet (He et al., 2014), MR-CNN (Gidaris & Komodakis, 2015), Fast RCNN (Girshick, 2015), Faster RCNN (Ren et al., 2015). RCNN extracts region proposals by selective Search (Van de Sande et al., 2011) method and then classifies them with a pretrained convolutional neural network (CNN). Fast RCNN shares the computation of proposals and runtime of object detection can be reduced dramatically. Faster R-CNN combines object proposal and detection into a unified network and thus the detection speed can be boosted with the help of GPU, replacing Selective Search with RPN to get proposal. These methods based on region proposal use the last layer output of a very deep CNN is too coarse leading to poor performance for detecting the small object. In this case, a 32×32 object will be just 2×2 when it goes to the last convolutional layer of CNN. The feature map size is too small and coarse for classification of some instances with small size. This is the reason why these approaches struggle with small objects.

We modify the region proposal network which is based on fully convolutional network for proposal generation. The original RPN attaches multi-scale anchors and outputs the object proposal. However, the anchor boxes are larger than most of small instance and result in ignoring the small and the scale and ratio do not cover most scale of object. The size of the small instance is less 32×32 that refers to COCO dataset. We propose our enhanced RPN for addressing that.

For improving detection, a number of approaches use different layers in a deep convolutional networks (ConvNet). Several approaches HyperNet (Kong et al., 2016), ParseNet (Liu, Rabinovich, et al., 2015) and ION (Bell et al., 2015) concatenate features of multiple layers to make feature more abundant. There are recent methods exploiting lateral/skip connections that associate low-level feature maps across resolutions and semantic levels, including U-Net (Ronneberger et al., 2015) and SharpMask (Pinheiro et al., 2016). The multi-scale information is especially important for small objects, which requires the higher spatial resolution provided by lower-level layers. FPN (feature pyramid network) generates multi-scale feature maps after backbone achieving significant progress in detecting multi-scale objects.

Generally, context is useful for improving the object detection performance. The context is the information which captures the basic pixels/patches from the surrounding areas of an object proposal. MR-CNN proposes a multi-region object detection system that can steer the ConvNet to focus on different regions of the object. Zhu et al. (2015) use both segmentation and context to improve object detection accuracy. ION stacks spatial RNNs (iRNN, Le et al., 2015) model context outside the region of interest and shows improvements on small object detection. We propose a simple and efficient approach. When given an object proposal, in addition to cropping the proposal region, we extend the corresponding context region surrounding the proposal region, called ‘border’. The border is set to be several times larger than the proposal region. We then feed both regions into classification neural network.

We develop a novel Spatial Fine-Grained Features to combine deep, coarse information with shallow, fine information to make features more abundant. More specifically, we take a different approach with concatenating the higher resolution features with the low-resolution features by stacking spatial adjacent features into different channels. The Spatial Fine-Grained Features contain more detail information at higher resolution and keep the feature map spatial character. For combining low-resolution features with spatially coarser, but semantically stronger, these features are then enhanced with features from the bottom-up pathway via lateral connections.

For improving the performance further without extra cost, we use an online hard example mining (OHEM) for training detection models and yield consistent and significant boosts in detection performance. We use soft-NMS (Bodl et al., 2017) to obtain consistent improvements. Further, soft-NMS does not require any extra-training and is simple to implement, and it can be easily integrated in the object detection pipeline.

We demonstrate that both sources of additional information, Spatial Fine-Grained Feature and context, are complementary. This matches our intuition that Spatial Fine-Grained Feature captures more fine-grained detail
and keeps spatial information, while embedded context features look broadly for small object across the image. Then, an enhanced region proposal generator is proposed to get the objectness for small object to obtain a small set of proposal while keeping a high recall rate. Further, the efficient and efficiently OHEM and soft-NMS improve the performance. We show large improvements on the PASCAL VOC (Everingham et al., 2010) and MS COCO (Lin et al., 2014) object detection datasets. In general, we find that our approach is more adapt at detecting small objects than previous state-of-the-art methods.

We make the following contributions:

1. We introduce the SFGN architecture that leverages the Spatial Fine-Grained Feature and embedded context for object detection.
2. We propose an enhanced region proposal generator to increase the small object detection performance.
3. The efficient and efficiently OHEM and soft-NMS methods improve the performance for object detection.
4. We achieve state-of-the-art results on PASCAL VOC 2007, with a mAP of 81.2%, VOC 2012, with an mAP of 80.6%, and on MS COCO, with an mAP of 32.3%.
5. We analyse the detector’s performance and find improved accuracy across the board, but, in particular, for small objects.

2. Related work

CNN backbone for Object Detection: CNN have recently enjoyed a great success in large-scale image classification. As a typical network, VGG16 (Simonyan & Zisserman, 2014) achieves the best results than its previous model on ImageNet datasets. After the success of VGG16, its structure combines into object detector aiming to extract feature vectors from input images. With the network depth increasing, a degradation problem is exposed, and accuracy gets saturated and then degrades rapidly. ResNet (He et al., 2016) addresses the problem by introducing a deep residual learning framework and significantly increases the depth of network architecture to over 1000 layers. ResNet-50, which balances number of parameters and model performance, is a most popular choice for feature extractor in ResNet family. Later many backbones are designed for different purposes. MobileNet and shuffleNet (X. Zhang et al., 2018) aim to build light-weight backbone. ResNeXt (Xie et al., 2017) considers reducing memory and computation cost in ResNet structure. DetNet (Li et al., 2018) is a special backbone designed only for object detection task. Hourglass Network (Newell et al., 2016) is proposed for capturing both global and local information. EfficientNet introduces a new family of strong backbone searched by NAS (network architecture search). Even though those new designed backbones are structural advanced, the most wildly used structure is ResNet.

Deep ConvNet object detectors: With the development of CNN, there are two established classes of methods for object detection, one based on region proposal called two-stage detector, the other based on region-free namely one-stage or single-stage detector.

The one-stage detector family is the region-free methods. These methods treat object detection as a single shot problem, straight from image pixels to bounding box coordinates by fully convolutional networks. The main advantage of these detectors is high efficiency. YOLO uses a single neural network that predicts bounding boxes and class probabilities directly from full images frame. SSD discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location. RetinaNet introduces focal loss for rebalance foreground and background samples. YOLOv2 uses anchor strategy and achieves state-of-the-art results at that time. CornerNet removes anchors by feature maps, class heatmaps, pair embeddings and corner offsets and inspires key-point-detection based methods. All those region-free methods design a special component for proposal region instead of ROI proposal part. At prediction time, the network generates scores for the presence of each object category in each default box and produces adjustments to the box to better match the object shape. In practice, two-stage methods perform better than one-stage methods. YOLOv3’s (Joseph & Ali, 2018) prior detection system reuses the classifier or locator to perform detection tasks. They apply the model to multiple locations and scales of the image. Those areas with higher scores can be regarded as the test results. In addition, a single neural network is applied to the entire image. The network divides the image into different regions, thereby predicting the bounding box and probability of each region, and these bounding boxes are weighted by the predicted probability. During the test, the entire image is viewed, so its prediction uses the global information in the image. C. Zhu et al. (2020) trains a single-stage anchor-free detector called Soft Anchor-Point Detector (SAPD), which boosts the performance of the anchor-point detector over the key-point counterparts while maintaining the speed advantage. YOLOv4 (Bochkovskiy et al., 2020) is improved on the basis of YOLOv3, which greatly improves the detection accuracy of the model while ensuring the speed. YOLOF (Wang et al., 2021) uses first-level features to detect and replace complex feature pyramids to solve
the optimization problem. This method achieves performance comparable to RetinaNet and the reasoning speed is 2.5 times faster.

The two-stage detector shows dramatic improvements in accuracy. Object detection task is divided into two sub-problems: at the first stage, a dedicated region proposal generation network is grafted on CNN which could generate high quality candidate boxes. Then at the second stage, a region-wise subnetwork is designed to classify and refine these candidate boxes. To fully inherit the advantages of one-stage and two-stage detectors while overcoming their disadvantages, S. Zhang et al. (2018) present a novel RefineDet which achieves better accuracy than two-stage detectors and maintains comparable efficiency of one-stage detectors. Xie et al. (2021) proposes an efficient and simple two-stage oriented detector, named oriented R-CNN, that achieves competitive results with respect to two-stage detectors, while maintaining the efficiency of single-stage detectors.

R-CNN (Girshick et al., 2013) detector leads this wave with impressive results on PASCAL VOC and ImageNet detection. Since R-CNN, there has been rapid progress in region-based CNN. SPPNet (He et al., 2014) speeds up detection. Since R-CNN, there has been rapid progress with impressive results on PASCAL VOC and ImageNet. Tan et al. (2020) systematically study network architectures design choices for efficient object detection and propose a weighted bidirectional feature network and a customized compound scaling method, to improve accuracy and efficiency. Kong et al. (2020) present FoveaBox, a simple, effective and completely anchor-free framework for generic object detection, by simultaneously predict the object position and the corresponding boundary.

Context: Generally, context is useful for improving the object detection performance in natural scenes. Zhu et al. (2015) use both segmentation and context to improve object detection accuracy. Mottaghi et al. (2014) study the role of context in existing object detection approaches and further proposed a model that exploits both the local and global context. For finding small instances, context is key as shown in multiple recognition tasks. In object detection, ION stacks spatial RNNs (IRNN Le et al., 2015) model context outside the region of interest and shows improvements on small object detection. In pedestrian detection, Kim et al. (2016) uses ground plane estimation as contextual features and improves Hard example selection in deep learning.

Hard example selection in deep learning: There is recent work (Loshchilov & Hutter, 2015; Shrivastava et al., 2016; Simo-Serra et al., 2014; Wang & Gupta, 2015) that selects hard examples for training deep networks. All these methods base their selection on the current loss for each data point. Simo-Serra et al. (2014) independently select hard positive and negative example from a larger set of random examples based on their loss to learn image descriptors. Given a positive pair of patches, Wang and Gupta (2015) find hard negative patches from a large set using triplet loss. Loshchilov and Hutter (2015) investigate online selection of hard examples for minibatch SGD methods. Their selection is also based on loss, but the focus is on ConvNets for image classification. Complementary to Loshchilov and Hutter (2015) and Shrivastava et al. (2016) propose a novel bootstrapping technique called online hard example mining (OHEM) for training detection models based on Fast RCNN. B. Zhu et al. (2020) propose a differentiable label assignment strategy named AutoAssign. It tackles label assignment in a fully data-driven manner by automatically determine the positives/negatives in both spatial and scale dimensions. Cao et al. (2020) present Prime Sample Attention (PISA), a simple and effective sampling and learning strategy to highlight important samples.

3. SFGNet architecture

Our SFGNet framework is illustrated in Figure 1. Initially, a single-scale image is forwarded through the full convolutional layers (ResNet101) and the activation maps are
produced. We aggregate feature maps and then compress them into a uniform space by using spatial pooling and spatial upsampling, Spatial Fine-Grained Feature. The Spatial Fine-Grained Feature is shared both in generating proposals and detecting object. In addition, the enhanced RPN is constructed to produce proposals based on Spatial Fine-Grained Feature. Moreover, for each proposal, we extract a fixed-size feature map from the proposal and its spatial context region by ROI Pooling, respectively. Each feature is concatenated to produce a fixed-length feature descriptor, Spatial Context Feature. Finally, these proposals with Spatial Context Feature are classified and located.

### 3.1. Spatial fine-grained feature

As shown in Figure 2, the Spatial Fine-Grained Feature is extracted as follows. First, an image is processed by ResNet-101 backbone network. The convolutional feature maps at each stage of the backbone network are produced. Second, we extract a fine-grained feature descriptor by spatial pooling from layer Res3 that is shallow but high-resolution layers. We compute the highly-abstracted feature by spatial upsampling from layer Res5 that is the deep but highly semantic layer. Finally, the intermediate but complementary feature combines fine-grained from Res4 details with highly-abstracted information to construct Spatial Fine-Grained Feature.

Compared with a high semantic but coarse feature, Spatial Fine-Grained Feature has several advantages: (a) Multi-level feature concatenation. The fine-grained features and semantic features are really complementary for object detection task. (b) Spatial information holding. The feature exploits position indices to keep space details for improving small size precise localization.

#### 3.1.1. Backbone network

State-of-the-art models for object detection are based on adaptations of convolutional networks that had originally been designed for image classification. In this paper, the backbone architecture is based on ResNet-101 (He
et al., 2016), pre-trained on ImageNet. For ResNet-101, we just use several feature activations output by each stage’s last residual block. We denote the output of these last residual blocks as (Res2; Res3; Res4; Res5) for conv2, conv3, conv4, and conv5 outputs, and note that they have strides of (4, 8, 16, 32) pixels with respect to the input image.

3.1.2. Spatial pooling
Multi-scale representation and its combination are proven to be effective in many recent object detection tasks, especially for localizing smaller objects (e.g. HyperNet, Kong et al., 2016 and ION, Bell et al., 2015). To combine multi-level maps at the same resolution, these methods carry out different sampling strategies for different layers. The fine-grained details with highly-resolution layer are subsampled by the pooling. The original pooling layer selects the max or average value represents the local region. The operation results in the loss of detailed information to some extent. Furthermore, the pooling layer ignores the local relation between pixels in the same feature map and loses spatial information between the different feature.

To solve the problem above, the spatial pooling is proposed. The spatial pooling distributes every pixel with position indices into different channels instead of representing the local region with a max/average value. Every pixel not only involves detailed information but also contains the relative position for keeping the 2D spatial characters. The detailed information and spatial characters are important for object detection and accurate localization.

The spatial pooling processes as follows: (a) slide a small window over the convolutional feature maps, and we will get local detailed information within the small window, and arrange the information with special position indices to different feature map respectively; (b) slide the window without overlap to next position and arrange the information; (c) repeat (a), (b), until the end position of feature map. Figure 3 presents an illustration of the spatial pooling operation.

3.1.3. Spatial upsampling
To compress the multi-scale features into a uniform space, the spatially coarser, but semantically stronger layer is upsampled by deconvolution operation or max-pooling indices (Dai et al., 2016). However, the deconvolutional network requires extra parameters that increase the computer cost and store memory. The max-pooling indices produce a sparse feature map that do not contain abundant semantic information.

To address the dilemma, the spatial upsampling produces the upsampled feature maps by using different channel maps and position indices. The region of upsampled map consists of the same position pixel at different maps. The region encodes the position indices of every pixel. The spatial upsampling processes: (a) combine every value within the small resolution map at same position in different feature maps to get the local region

![Figure 3](image_url) The illustration of the spatial pooling.
of the high resolution. Then, the combined local region contains the detailed value and position indices; (b) stride the next position in small resolution and combine the region for high resolution; (c) repeat (a) and (b), until the end position of the feature map. The operation is shown in Figure 4.

In this paper, the size of the sliding window is $2 \times 2$ and the stride is 2 during the spatial pooling. The size and the channel of the outputs are 1/4 and 4 times of the inputs. During the spatial upsampling, we select four different feature maps to combine a higher feature map. The size and the channels of the outputs are 4 and 1/4 of the inputs.

3.1.4. Combining spatial fine-grained feature

To compress the high resolution but fine-grained feature and the coarse but semantic feature into a uniform space, the lower layer and the higher layer are processed by spatial pooling and spatial upsampling respectively. The intermediate layer is concatenated via lateral connections, thus enabling to construct Spatial Fine-Grained Feature.

To reduce redundant information with much higher compute requirement, we combine the last layer Res5 and two intermediate layers whose scales are 2x(Res4) and 4x(Res3) of the last layer. We choose the middle-sized layer as a reference scale ($= 2x$) and concatenate the 4x-scaled layer and the last layer by spatial pooling and spatial upsampling. Figure 5 shows the building block that constructs Spatial Fine-Grained Feature.

3.2. Enhanced RPN

The RPN is the current state-of-the-art method for proposal generation with high recall. To generate region proposals, the traditional method slides a small network over the convolutional feature map to get multiple region proposals, called ‘anchor’. Through the multi-scale anchor, the RPN achieves a high recall in big object proposal generation. However, the RPN results in poor performance in the small object proposal generation. The main reasons are: (1) the RPN anchors are too large to detect the small object; (2) the scales and ratios of anchor are handpicked.

For predicting good detections, we propose the enhanced RPN by data statistic and prior knowledge. Instead of choosing priors by hand, we run k-means clustering on bounding boxes of the training set to automatically find good priors. Furthermore, we add smaller scales. For the sake of efficiency and efficiency, we use 5 scales and 5 ratios. This architecture is illustrated in Figure 6. We add several small scales and aspect ratios to cover the multi-scale object. Therefore, the enhanced RPN can improve the performance in the small object detection.

3.3. Spatial context feature

Context is an important cue for object detection. And it is even more important for small object detection, since small objects contain little signal to exploit. The feature from the proposal region contains less information and
is less discriminative, so it is difficult to recognize. Hence we argue that one must use context beyond the object extent to increase available cues.

We propose a simple and efficient approach to embed the context and construct the Spatial Context Feature. An object proposal is given by the enhanced RPN, and we
crop the corresponding context region around the proposal region. The context region is 1.2 times larger than the proposal region. A fixed-size feature from proposal and context region is extracted. Each feature is concatenated after ROI pooling to produce a fixed-length feature descriptor, Spatial Context Feature, as shown in Figure 7. An extra convolutional layer is used to reduce the number of model parameters. It helps to compress redundant context and object information without the loss of accuracy.

### 3.4. Training and inference

Our use a multi-task loss following

$$L(p, u, t^u, v) = L_{cls}(p, u) + \lambda[u \geq 1]L_{loc}(t^u, v)$$  \hspace{1cm} (1)

Each training RoI is labelled with a ground-truth class $u$ and a ground-truth bounding-box regression target $v$. We use a multi-task loss $L$ on each labelled RoI to jointly train for classification and bounding-box regression: in which

$$L_{cls}(p, u) = -\log p_u$$  \hspace{1cm} (2)

is log loss for true class $u$. The second task loss, $L_{loc}$, is defined over a tuple of true bounding-box regression targets for class $u$, $(v_x, v_y, v_w, v_h)$, and a predicted tuple $t^u = (t^u_x, t^u_y, t^u_w, t^u_h)$, again for class $u$. The Iverson bracket indicator function $[u \geq 1]$ evaluates to 1 when $u \geq 1$ and 0 otherwise. By convention the catch-all background class is labelled $u = 0$. For background RoIs, there is no notion of a ground-truth bounding box and hence $L_{loc}$ is ignored.

For bounding-box regression, we use the loss

$$w_i L_{loc}(t^u_i, v_i) = \sum_{i \in \{x, y, w, h\}} \text{smooth}_{L^1}(t^u_i - v_i)$$  \hspace{1cm} (3)

in which

$$\text{smooth}_{L^1}(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases}$$  \hspace{1cm} (4)

The hyper-parameter $\lambda$ in Equation (1) controls the balance between the two task losses. We normalize the ground-truth regression targets $v_i$ to have zero mean and unit variance. All experiments use $\lambda = 1$.

We train and test object detection networks on images of a single scale. We rescale the images so that their shorter side is $s = 600$ pixels. We use a weight decay of 0.0005 and a momentum of 0.9. Learning rate policy is to control the learning rate dynamically based on plateau detection (Kim et al., 2016). We measure the moving average of loss and decide it to be on-plateau if its improvement is below a threshold during a certain period of iterations. Whenever the plateau is detected, the learning rate is decreased by a constant factor. We train all methods with SGD. The size of mini-batch is 8. The epochs is 50k.

These hyper-parameters are not carefully chosen for a particular dataset. For anchors in original RPN, the 3 scales with box areas of $128 \times 128$, $256 \times 256$ and $512 \times 512$ are used. For anchors in enhanced RPN, we use 5 scales with box areas of $8 \times 8$, $16 \times 16$, $32 \times 32$, $64 \times 64$, and $128 \times 128$ pixels by prior value. The 5 aspect ratios of 0.333, 0.5, 1, 1.5, 2 are used by k-means. Some proposals highly overlap
with each other. To reduce redundancy, we adopt NMS on the proposal regions. We use the top-N ranked proposal regions for detection.

It is easy for our method to adopt OHEM during training. Our negligible per-ROI computation enables example mining with nearly cost-free. Assuming that there are \( N \) proposals in per image, in the forward pass, we evaluate the loss of all \( N \) proposals. Then we sort all ROIs (positive and negative) by loss and select \( B \) RoIs that have the highest loss. Backpropagation is performed based on the selected examples.

During the process of training the enhanced RPN, each mini-batch arises from a single image that contains many positive and negative example anchors. The sampled positive and negative anchors have a ratio of up to 1:1 for optimizing the loss functions ideally. But this will bias towards negative samples as they are dominated when we randomly sample 256 anchors in an image. To optimize the loss functions, we limit the ratios of positive sample and negative sample between 1:2 and 2:1.

During the inference, we evaluate 300 ROIs. Instead of NMS, the results are post-processed by soft-NMS. Soft-NMS propose a single line modification to the traditional greedy NMS algorithm. We set the threshold of IoU is 0.5.

We also measured the speed of our method implemented with PyTorch on Nvidia Titan Xp. The average time for our method to process one image frame is about 5 fps.

4. Experimental evaluation

We evaluate SFGNet on three major datasets: PASCAL VOC 2007, PASCAL VOC 2012 and MS COCO. SFGNet is compared with other state-of-the-art detection methods on three datasets. We also provide deep analysis for the SFGNet and other detection methods.

4.1. PASCAL VOC 2007 results

We comprehensively evaluate our method on the PASCAL VOC 2007. This dataset covers 20 object categories. We primarily evaluate detection by mean Average Precision (mAP), because this is the actual metric for object detection. We compare SFGNet with other state-of-the-art detectors, Faster RCNN (Ren et al., 2015), R-FCN (Dai et al., 2016), HyperNet (Kong et al., 2016), ION (Bell et al., 2015), MR-CNN (Gidaris & Komodakis, 2015), YOLOv2 (Redmon & Farhadi, 2016) and SSD (Liu, Anguelov, et al., 2015) for generic object detection. The experiment results of the compared methods are referenced by the corresponding paper.

As shown in Table 1, SFGNet achieves an mAP of 81.2% which is 6.1 points higher than other methods.

| Method             | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  | Train | mAP  |Train
Figure 8. Selected examples of object detection results on the PASCAL VOC 2007 test set.
at most. As we have shown above, this is because SFGNet is elaborately designed by fine-grained details with highly-abstracted information and benefits from more details and spatial information. Proposals generated by the enhanced RPN are more accurate and possess higher recall than those generated by the original RPN. Embed the context around ROI, the information is enriched to improve the performance. We use OHEM and soft-NMS to obtain consistent improvements. By applying these methods described above, SFGNet makes for better object localization, especially when the object size is small. Figure 8 shows some results on the PASCAL VOC 2007 test set.

Fast/Faster R-CNN and R-FCN use the feature map of the last layer that is highly abstract but coarse and lacking in the detailed information, to detect objects. Therefore, these approaches struggle with small object. Compared with RCNN, MR-CNN uses multi-region to improve the performance. MR-CNN still has the problem in detecting small object, because MR-CNN use the feature map of the last layer for detection. Through adding several new layers, SSD employs multi-scale feature maps to detect objects in different scales leading to performance better than YOLO. SSD still struggles with small objects detection, mainly because of lack of exploring the detection capacities of the higher-resolution maps. HyperNet and ION are the solutions with concatenation of multi-scale features. They combine fine-grained details with high-level feature map of the image for better object detection and precise localization, especially for improving detection of small objects. Since the direct concatenation of all abstract layers may produce redundant information with much higher compute requirement, and ignore the space and detailed information of feature maps that are important for locating the objects.

### 4.2. PASCAL VOC 2012 results

We also compare against these methods on PASCAL VOC 2012 dataset which is similar to the VOC 2007 dataset and evaluation metric. In Table 2, SFGNet achieves an mAP of 80.6%, which is higher than the counterparts, and is the most accurate for most categories. Figure 9 shows some results on the PASCAL VOC 2012 test set.

### 4.3. MS COCO results

The MS COCO dataset is more challenging than the PASCAL VOC dataset, which involves 80 object categories. The dataset contains 80k images on training set, 40k images on the validation set and 20k images on the test-dev set. The shorter side of the images that we rescale is 800 pixels. The rest of the implementation details are
Figure 9. Selected examples of object detection results on the PASCAL VOC 2012 test set.
Table 3. Detection results on the COCO 2015 test-dev. Legend: 07+12:07 trainval + 12 trainval, SF: include Spatial Fine-Grained Feature, SR: include Enhanced RPN, CF: include Spatial Context Feature, OS: include OHEM and soft-NMS.

| method       | SF  | ER | CF  | OS  | Train       | Test data | mAP0.5 | mAP0.5:0.95 |
|--------------|-----|----|-----|-----|-------------|-----------|--------|-------------|
| SSD512       |     |    |     |     | Trainval35k | Test-dev  | 46.5   | 26.8        |
| YOLO2        |     |    |     |     | Trainval35k | Test-dev  | 44.0   | 21.6        |
| R-FCN        |     |    |     |     | Trainval   | Test-dev  | 51.5   | 29.2        |
| ION          |     |    |     |     | Trainval35k | Test-dev  | 53.4   | 31.2        |
| Faster RCNN  | 1   |    |     |     | Trainval35k | Test-dev  | 48.4   | 27.2        |
| SFGNet 1     | 1   |    |     |     | Trainval35k | Test-dev  | 53.7   | 31.4        |
| SFGNet 1 1   | 1   | 1  | 1   |     | Trainval35k | Test-dev  | 53.9   | 31.8        |
| SFGNet 1 1 1 1| 1   | 1  | 1   | 1  | Trainval35k | Test-dev  | 54.0   | 31.9        |

the same as those on PASCAL VOC. We evaluate the mAP averaged for IoU [0.5 : 0.05 : 0.95] (MS COCO’s standard metric, simply denoted as mAP(0.5: 0.95) and mAP0.5 (PASCAL VOC’s metric).

As shown in Table 3, our single-scale trained SFGNet has a result of 54.1% and 32.3% on the COCO test-dev set for mAP 0.5 and mAP (0.5:0.95). This is comparable to other methods baseline. They are 10.1% points higher for mAP0.5 and 10.7% points higher for mAP (0.5:0.95) than the counterparts under the same protocol at most. It is noteworthy that our method performs better on objects of small sizes. This indicates that SFGNet performs excellently in improving the localization accuracy at mAP(0.5:0.95). Figure 10 shows some results on the MS COCO test-dev set.

4.4. Improvement for small objects

In general, small objects are challenging for detectors: they have smaller in size and fewer pixels on the object; they are harder to localize; and there can be many more of them per image. The feature from the proposal region contains less information and discrimination. Therefore, for other detection approaches, it is difficult to recognize small objects. SFGNet outperforms other state-of-the-art methods in small object detection on the COCO and PASCAL VOC datasets. For COCO, if we look at small objects ('Small' means area < 32×32 pixels), average precision and average recall are 16.4% and 28% for mAP@[.5,95] respectively. Similarly, for small objects ('bottle', 'chair', and 'plant') on the VOC2007 and VOC012, SFGNet has great improvements in Tables 1 and 2. Figure 8 and Figure11 shows some results of small objects on the COCO and PASCAL VOC.

4.5. Ablation experiments

Enhance RPN vs RPN. To investigate the behaviour of enhance RPNs for higher recall, we conducted several ablation studies on PASCAL VOC 2007, PASCAL VOC 2012 and MS COCO. Proposals generated by the enhanced RPN are more accurate and possess higher recall than those generated by the original RPN, as shown in Table 4. The Enhanced RPN achieves a 98.8%, 97.6% and 95.4% recall which are 2.2 points, 3.2 points and 5.1 points higher than the RPN on PASCAL VOC 2007, PASCAL VOC 2012 and MS COCO, respectively. We also evaluate the effects of more powerful Enhanced RPN. First, Enhanced RPN adds several small scales and aspect ratios to cover smaller scale object for higher recall. Next, the good priors of the scales and ratios are found by the k-means clustering automatically on the training dataset. Finally, the spatial context is added around the object to increase available cues.

The size of the context region. We disentangle influence about the size of the context region on the Enhanced RPN. For this purpose, we add the corresponding context region around the proposal region. We fix the same setting as original RPN and evaluate the detection recall by changing the proposal regions that extents the context. The context region that is different scale than the proposal region for recall, as shown in Table 5. In Table 5, we compare the different values of the scale. The recall is highest when the scale is 1.2 times. With the increment of the scale between 1 and 1.2, the recall increases because the proposal region uses context to increase available cues. With the increment of the scale from 1.3 to 1.7, the recall decreases dramatically due to the background information.

The windows size in Spatial pooling and Spatial upsampling. We investigate the role of the windows size in spatial pooling and spatial upsampling. To evaluate the effects, we set up three shapes of kernels, i.e. 2×2, 3×3, 5×5. Table 6 shows the results when we use different settings for mAP0.5. The second column is the highest mAP when the size of window is 2×2.

Backbone. To compare our approach with the state-of-the-art methods using the backbone VGG16, the quantitative results in Table 7. Compared to the other methods, our method achieves a remarkable gain in terms of mAP on different dataset.
Figure 10. Detection examples on MS COCO test-dev set. Legend: $07 + 12$: $07$ trainval $+$ $12$ trainval, SF: include Spatial Fine-Grained Feature, SR: include Enhanced RPN, CF: include Spatial Context Feature, OS: include OHEM and soft-NMS.
Figure 11. Selected small examples of object detection results on the MS COCO and PASCAL VOC test set.

Table 4. The recall results in RPN and Enhanced RPN on different dataset.

|                      | PASCAL VOC 2007 | PASCAL VOC 2012 | MS COCO |
|----------------------|-----------------|-----------------|---------|
| RPN                  | 96.6%           | 94.6%           | 90.3%   |
| Enhanced RPN         | 98.8%           | 97.6%           | 95.4%   |

Table 6. Evaluation results on dataset using different kernel size.

|                      | 2×2      | 3×3      | 5×5      |
|----------------------|----------|----------|----------|
| PASCAL VOC 2007      | 81.2%    | 80.7%    | 80.1%    |
| PASCAL VOC 2012      | 80.6%    | 79.3%    | 78.7%    |
| MS COCO              | 54.1%    | 52.5%    | 51.1%    |

Table 5. The recall in different scale.

|   | 1   | 1.1 | 1.2 | 1.3 | 1.4 | 1.5 | 1.6 | 1.7 |
|---|-----|-----|-----|-----|-----|-----|-----|-----|
| PASCAL VOC 2007 | 96.6% | 97.1% | 97.7% | 96.1% | 92.9% | 87.7% | 81.4% | 78.1% |
| PASCAL VOC 2012 | 94.6% | 95.7% | 96.3% | 94.2% | 90.3% | 85.3% | 79.8% | 75.3% |
| MS COCO          | 90.3% | 92.2% | 94.2% | 90.1% | 85.1% | 80.2% | 75.2% | 70.1% |
Table 7. Evaluation results on dataset using VGG16.

|                     | Faster RCNN | R-FCN | ION   | HyperNet | SSD512 | YOLOv2 | SFGNET |
|---------------------|-------------|-------|-------|----------|--------|--------|--------|
| PASCAL VOC 2007     | 73.2%       | 75.6% | 78.5% | 76.3%    | 78.6%  | 75.2%  | 80.4%  |
| PASCAL VOC 2012     | 70.4%       | 72.1% | 76.4% | 71.4%    | 73.4%  | 72.4%  | 77.9%  |
| MS COCO             | 42.1%       | 45.3% | 53.4% | –        | 46.5%  | 43.5%  | 54.1%  |

5. Conclusion

We have presented SFGNet, a fully trainable deep architecture for jointing region proposal generation and object detection. Spatial Fine-Grained Feature provides an efficient combination of multi-scale features and keeps the spatial information. Enhanced RPN produces small number of object proposals while guaranteeing high recalls. Our architecture embeds the context region around proposal to improve the object description. OHEM and soft-NMS are added to improve the performance without any additional hyper-parameters and test cost. Experiments are conducted on both PASCAL VOC and COCO with state-of-the-art results, and our proposed architecture turn out to be particularly effective in improving small object detection.

The shortcomings and future work are below: the detection is based on anchor. The size and ratio of anchor will change as the data. And the anchor leads to imbalance the positive and negative samples. We will select anchor free. The spatial fine-grained feature is combined by several layers that just concatenate sample. How do combine the layers for valid feature is the future work. We select the two-stage detection. The architecture is slow and the compute cost is high for embedded hardware in vehicle. We will prune the network to reduce redundant information.

Acknowledgments

The authors also would like to thank the support of the Major Science and Technology Projects Fundation of Liaoning Province, the Xing Liao Ying Cai projects Fundation of Liaoning Province, the Jie Bang Gua Shuai Science and Technology Project Fundation of Liaoning Province and the project name is [Development and application of a vehicle-cloud collaborative self-evolution platform for advanced autonomous driving] under Grant number [1649222550708].

Data availability statement

The PASCAL VOC and COCO dataset used in this experiment are derived from public domain resources. The PASCAL dataset are available in [voc2007] and [voc2012] at [http://host.robots.ox.ac.uk/pascal/VOC], reference number (Everingham et al., 2010). The COCO dataset are available at [http://cocodataset.org], reference number (Lin et al., 2014).

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the [Major Science and Technology Projects Fundation of Liaoning Province] and the project name is [Research and development of L3 commercial vehicle autonomous driving system based on artificial intelligence] under Grant number [XYLC190209], the [Xing Liao Ying Cai projects Fundation of Liaoning Province] and the project name is [Research and development and industrialization of key technologies for intelligent driving vehicles ] under Grant number [2019JH1/10100026], the [Jie Bang Gua Shuai Science and Technology Project Fundation of Liaoning Province] and the project name is [Development and application of a vehicle-cloud collaborative self-evolution platform for advanced autonomous driving] under Grant number [1649222550708].

References

Azizpour, H., & Laptev, I. (2012). Object detection using strongly-supervised deformable part models. Proceedings of the European Conference on Computer Vision.
Bell, S., Zitnick, C. L., Bala, K., & Girshick, R. B. (2015). Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. CoRR, abs/1512.04143.
Bochkovskiy, A., Wang, C., & Liao, H. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. CoRR, abs/2004.10934.
Bodl, A., Singh, B., Chellappa, R., & Davis, L. S. (2017). Soft-NMS: Improving object detection with one line of code. IEEE International Conference on Computer Vision.
Cao, Y., Chen, K., Loy, C. C., & Lin, D. (2020). Prime sample attention in object detection. IEEE/CVF Conference on Computer Vision and Pattern Recognition.
Dai, J., Li, Y., He, K., & Sun, J. (2016). R-FCN: Object detection via region-based fully convolutional networks. CoRR, abs/1605.06409.
Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. IEEE Computer Society Conference on Computer Vision and Pattern Recognition.
Dollár, P., Appel, R., Belongie, S. J., & Perona, P. (2014). Fast feature pyramids for object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(8), 1532–1545. https://doi.org/10.1109/TPAMI.2014.2300479
Everingham, M., Van Gool, L. J., Williams, C. K. I., Winn, J. M., & Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. International Journal of Computer Vision, 88(2), 303–338. https://doi.org/10.1007/s11263-009-0275-4
Felzenszwalb, P. F., Girshick, R. B., McAllester, D. A., & Ramanan, D. (2010). Object detection with discriminatively trained
part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9), 1627–1645. https://doi.org/10.1109/TPAMI.2009.167

Gidaris, S., & Komodakis, N. (2015). Object detection via a multi-region and semantic segmentation-aware CNN model. IEEE International Conference on Computer Vision.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2013). Rich feature hierarchies for accurate object detection and semantic segmentation. CoRR, abs/1311.2524.

Girshick, R. B. (2015). Fast-RCNN. Proceedings of the IEEE International Conference on Computer Vision.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. IEEE Conference on Computer Vision and Pattern Recognition.

He, K., Zhang, X. Y., Ren, S. Q., & Sun, J. (2014). Spatial Pyramid pooling in deep convolutional networks for visual recognition. CoRR, abs/1406.4729.

Honari, S., Yosinski, J., Vincent, P., & Pal, C. J. (2015). Recombinator networks: Learning coarse-to-fine feature aggregation. IEEE Conference on Computer Vision and Pattern Recognition.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. CoRR, abs/1704.04861.

Joseph, R., & Ali, F. (2018). YOLOv3: An incremental improvement. CoRR, abs/1804.02767.

Kim, K. H., Hong, S., Roh, B., Cheon, Y., & Park, M. (2016). Pvanet: Deep but lightweight neural networks for real-time object detection. CoRR, abs/1608.08021.

Kong, T., Sun, F., Liu, H., Jiang, Y., Li, L., & Shi, J. (2020). Foveabox: Beyond anchor-based object detection. *IEEE Transactions on Image Processing*, 29, 7389–7398. https://doi.org/10.1109/TIP.83.

Kong, T., Yao, A., Chen, Y., & Sun, F. (2016). HyperNet: Towards accurate region proposal generation and joint object detection. CoRR, abs/1604.00600.

Law, H., & Deng, J. (2018). CornerNet: Detecting objects as paired keypoints. Proceedings of the European Conference on Computer Vision.

Le, Q. V., Jaitly, N., & Hinton, G. E. (2015). A simple way to initialize recurrent networks of rectified linear units. CoRR, abs/1504.00941.

Li, Z., Deng, Y., & Sun, J. (2018). DttNet: Design backbone for object detection. Proceedings of the European Conference on Computer Vision.

Lin, T., Goyal, P., Girshick, R., He, K., & Dollar, P. (2018). Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2), 318–327. https://doi.org/10.1109/tpami.2018.2858826

Lin, T. Y., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollar, P. C., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. CoRR, abs/1405.0312.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. E., Fu, C. F., & Berg, A. C. (2015). SSD: Single shot multibox detector. CoRR, abs/1512.2325.

Liu, W., Rabinovich, A., & Berg, A. C. (2015). ParseNet: Looking wider to see better. CoRR, abs/1506.04579.

Loshchilov, I., & Hutter, F. (2015). Online batch selection for faster training of neural networks. CoRR, abs 1511.06343.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

Mottaghi, R., Chen, X., Liu, X., Cho, N. G., Lee, S. M, Fidler, S., Urtasun, R., & Yuille, A. L. (2014). The role of context for object detection and semantic segmentation in the wild. IEEE Conference on Computer Vision and Pattern Recognition.

Newell, A., Yang, K., & Deng, J. (2016). Stacked hourglass networks for human pose estimation. CoRR, abs/1603.06937.

Pinheiro, P. O., Lin, T. Y., Collobert, R., & Dollár, P. (2016). Learning to refine object segments. CoRR, abs/1603.08695.

Redmon, J., Divvala, S. K., Girshick, R. B., & Farhadi, A. (2015). You only look once: Unified, real-time object detection. CoRR, abs/1506.02640.

Redmon, J., & Farhadi, A. (2016). YOLO9000: Better, faster, stronger. CoRR, abs/1612.08242.

Ren, S. Q., He, K., Girshick, R. B., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. Conference and Workshop on Neural Information Processing Systems.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. CoRR, abs/1505.04597.

Shrivastava, A., Abhinav, G., & Ross, B. (2016). Girshick: Training region-based object detectors with online hard example mining. CoRR, abs/1604.03540.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image classification. CoRR, abs/1409.1556.

Simo-Serra, E., Trulls, E., Ferraz, L., Kokkinos, I., & Moreno-Noguer, F. (2014). Fracking deep convolutional image descriptors. CoRR, abs/1412.6537.

Tan, M., & Le, Q. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. Proceedings of the 36th International Conference on Machine Learning.

Tan, M., Pang, R., & Le, V. (2020). EfficientDet: Scalable and efficient object detection. IEEE Conference on Computer Vision and Pattern Recognition.

Van de Sande, K. A., Uijlings, J. R. R., & Gevers, T. (2011). Smeulders: Segmentation as selective search for object recognition. IEEE International Conference on Computer Vision.

Wang, C., Bochkovskiy, A., & Liao, H. M. (2021). Scaled-YOLOv4: Scaling cross stage partial network. CoRR, abs/2011.08036.

Wang, X., & Gupta, A. (2015). Unsupervised learning of visual representations using videos. IEEE International Conference on Computer Vision.

Xie, S., Girshick, R., & He, K. (2017). Aggregated residual transformations for deep neural networks. IEEE Conference on Computer Vision and Pattern Recognition.

Xie, X., Cheng, G., Wang, J., Yao, X., & Han, J. (2021). Oriented R-CNN for object detection. IEEE/CVF International Conference on Computer Vision.

Zhang, S., Wen, L., Bian, X., Lei, Z., & Li, S. Z. (2018). Single-shot refinement neural network for object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). ShuffleNet: An extremely efficient convolutional neural network for mobile devices. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
Zhu, B., Wang, J., Jiang, Z., Zong, F., Liu, S., Li, Z., & Sun, J. (2020). Autoassign: Differentiable label assignment for dense object detection. *CoRR*, 2007.03496.

Zhu, C., Chen, F., Shen, Z., & Savvides, M. (2020). *Soft anchor-point object detection*. European Conference on Computer Vision.

Zhu, Y., Urtasun, R., Salakhutdinov, R., & Fidler, S. (2015). *segDeepM: Exploiting segmentation and context in deep neural networks for object detection*. IEEE Conference on Computer Vision and Pattern Recognition.