Neurocomputing 366 (2019) 305–313

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Scale adaptive image cropping for UAV object detection

Jingkai Zhou[a], Chi-Man Vong[b], Qiong Liu[a,*], Zhenyu Wang[a]

[a] South China University of Technology, Guangzhou 510006, China
[b] University of Macau, Macau 999078, China

ARTICLE INFO

Article history:
Received 21 March 2019
Revised 13 June 2019
Accepted 27 July 2019
Available online 31 July 2019

Communicated by Dr. Zhen Lei

Keywords:
Data enhancement
UAV aerial imagery
Object detection
Deep neural network

ABSTRACT

Although deep learning methods have made significant breakthroughs in generic object detection, their performance on aerial images is not satisfactory. Unlike generic images, aerial images have smaller object relative scales (ORS), more low-resolution objects, and serious object scale diversity. Most researches focus on modifying network structures to address these challenges, while few studies pay attention to data enhancement which can be used in combination with model modification to further improve detection accuracy.

In this work, a novel data enhancement method called *scale adaptive image cropping* (SAIC) is proposed to address these three challenges. Specifically, SAIC consists three steps: ORS estimation in which a specific neural network is designed to estimate ORS levels of images; image resizing in which a GAN-based super-resolution method is adopted to up-sample images with the smallest ORS level, easing low-resolution object detection; image cropping in which three cropping strategies are proposed to crop resized images, adjusting ORS.

Extensive experiments are conducted to demonstrate the effectiveness of our method. SAIC improves the accuracy of feature pyramid network (FPN) by 9.65% (or relatively 37.06%). Without any major modification, FPN trained with SAIC won the 3rd rank on 2018 VisDrone challenge detection task.

1. Introduction

In recent years, aerial photography by unmanned aerial vehicle (UAV) has been widely used in various fields, including agriculture, surveillance, express, outdoor search and rescue, etc. These applications typically require instance-level information to help UAVs perceive the scene, make flight strategies, and complete missions. UAV object detection becomes a highly demanded technique to obtain instance-level information automatically.

Among object detection methods, deep learning methods are very promising and have pushed the whole object detection field a big step forward. However, they are still incapable to process aerial images. There are three serious object scale challenges regarding to aerial images, i.e. small object relative scale (ORS), low-resolution objects and serious object scale diversity.

Small ORS is an intractable problem. The scale of objects relative to the image in ImageNet [1], COCO [2] and VisDrone [3], an aerial image dataset released recently, is curved in Fig. 1. The curve of VisDrone is close to the top-left corner, and over 90% of objects occupy less than 1% of total image area. The median relative scale in VisDrone, COCO and ImageNet detection (DET) set are 1.73e-2, 9.56e-2, and 5.14e-1, respectively. That means if we want the objects in VisDrone to be as large as those in COCO, the images in VisDrone must be up-sampled to 5 times of COCO images, which are too big for GPU memory.

The low object resolution is another issue. The histograms of absolute object area of VisDrone and COCO is compared in Fig. 2. The mean absolute object area in VisDrone is 2.49e+3 pixels, much smaller than 2.05e+4 pixels in COCO. Low-resolution objects contain very limited visual information with low signal-to-noise ratios (SNR), making it hard to be distinguished from cluttered backgrounds. It is worth noting that the small object case in aerial images is mainly derived from objects with large distance instead of sensors. In fact, the captured images in VisDrone 2018 are already generally larger than COCO images in resolution.

Scale diversity of objects is also a challenging issue. The absolute object area measured in image pixels in VisDrone 2018 ranges from 3 pixels to 3.29e+5 pixels, which is as wide as the absolute object area distribution in COCO. The preset anchors in existing deep learning detectors are incapable to cover such large object scale span.

* Corresponding author.
E-mail addresses: 201510105876@mail.scut.edu.cn (J. Zhou), cmvong@um.edu.mo (C.-M. Vong), liuqiong@scut.edu.cn (Q. Liu), wangzy@scut.edu.cn (Z. Wang).

https://doi.org/10.1016/j.neucom.2019.07.073
0925-2312/

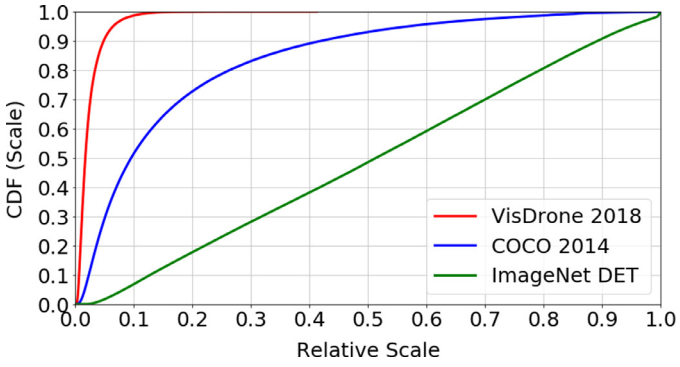


Fig. 1. Fraction of objects in the dataset vs. scale of objects relative to the image.

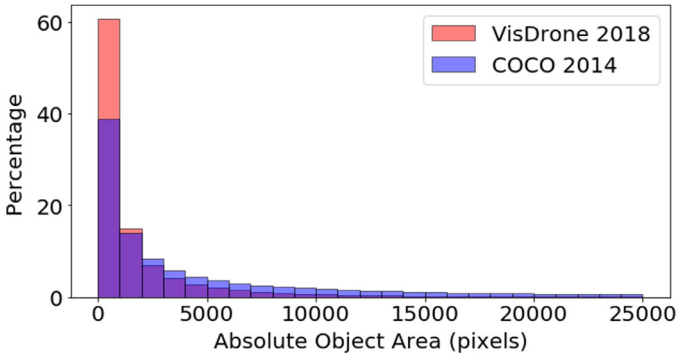


Fig. 2. Distribution of absolute object area.

Most recent researches deal with scale challenges by modifying network structure, such as using aggregated features [4–7], feature hierarchy [8–14], pyramid pooling strategy [15–17], or deformable network components [18,19], while few studies focus on data enhancement. Data enhancement can be used in combination with model modifications, and its gain on detection accuracy is sometimes greater than the total gain of multiple model modifications. This motivate us to explore a data enhancement method to handle scale challenges in UAV object detection.

We found that, in aerial images, object scale diversity is mainly caused by scale differences between images rather than scale differences inside images. To demonstrate it, we take 'car' as an example, which is the rigid body category with the largest number of objects in both VisDrone and COCO. To reflect the differences inside images, we calculate the variance of the car's area for each image, and then get the average variance over the entire dataset, expressed as *AV_car*. To reflect the differences between images, we calculate the car's average area for each image, and then get the variance of the average area over the entire dataset, expressed as *VA_car*. The statistics for COCO and VisDrone are shown in Fig. 3. *AV_car* of VisDrone is similar to COCO, while *VA_car* is nearly 5 times larger than COCO, demonstrating that scale differences between images is the key factor of object scale diversity in aerial images. If we can identify the object scale level (or ORS level) for each image and resize the image according to the level, scale diversity problem will be significantly alleviated.

From this inspiration, we propose a data enhancement method named scale adaptive image cropping (SAIC). There are three steps in SAIC: ORS estimation in which a specific neural network is designed to estimate ORS levels of images; image resizing in which a GAN-based super-resolution method (SRGAN) is adopted to upsample images with the smallest ORS level, significantly easing low-resolution object detection; image cropping in which three

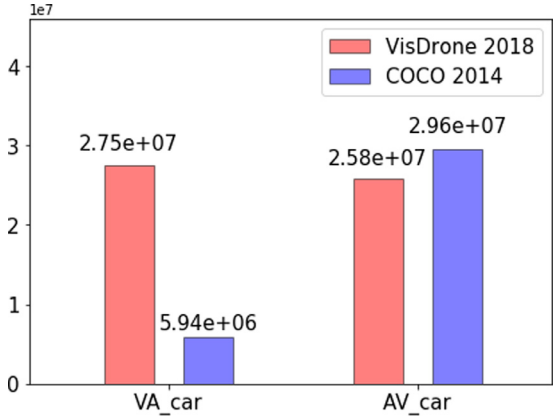


Fig. 3. Variance of average area and average area variance of the car object. In terms of formula, VA_car = Variance(Average(car area)), AV_car = Average(Variance(car area)).

cropping strategies are proposed to crop resized images, adjusting image ORS.

The main contributions of this paper are summarized as follows:

1. A novel data enhancement method SAIC is proposed to solve three scale challenges in UAV object detection. SAIC resizes images based on estimated ORS level to alleviate object scale diversity; SAIC uses different cropping strategies to adjust image ORS; SAIC adopts a GAN-based super-resolution method in resizing step to ease low-resolution object detection.
2. To better define ORS level of each image, normalized average ORS (NAORS) is proposed as an indicator, which takes object category information into account.
3. To better classify scale level, an ORS classification network is designed, in which an adaptive receptive structure is proposed to handle the scene scale variation. This structure shows better performance than hyper structure in HyperNet [4] and channel-wise attention structure in SEnet [20].
4. Extensive experiments are conducted to demonstrate the effectiveness of our method. SAIC improves the accuracy of FPN by 9.65% (37.06% relatively). Without any major modification, the resulting data enhanced FPN won the honorable mention in 2018 VisDrone challenge DET task.

2. Related work

Aerial Object Detection. With the rapid development of deep learning methods, many studies try to adapt deep learning methods for aerial object detection. Ševo and Avramović [21] use sliding window to generate RoIs and attach a simple CNN classifier for object-level vehicle classification. Similarly, Audebert et al. [22] adopt a semantic segmentation network to generate RoIs and use a CNN classifier to classify vehicles. Deng et al. [23] combine hierarchical feature maps to improve RPN performance in aerial images. Sommer et al. [24] directly extend the Faster R-CNN [25] to aerial images by redesigning the anchor settings and backbone structure. SAPNet [26] adds an RPN on shallow feature maps to detect small object in aerial images, which is similar to MS-CNN [10]. In general, these aerial object detection methods are too naïve compared to recent generic methods. Therefore, in this paper, a generic method FPN [11] is chosen as the baseline.

Model modification. The most common way to handle scale problems is to modify various model components in detection pipeline.

By utilizing aggregated features and feature hierarchy, the scale capability of backbone can be improved. Several approaches

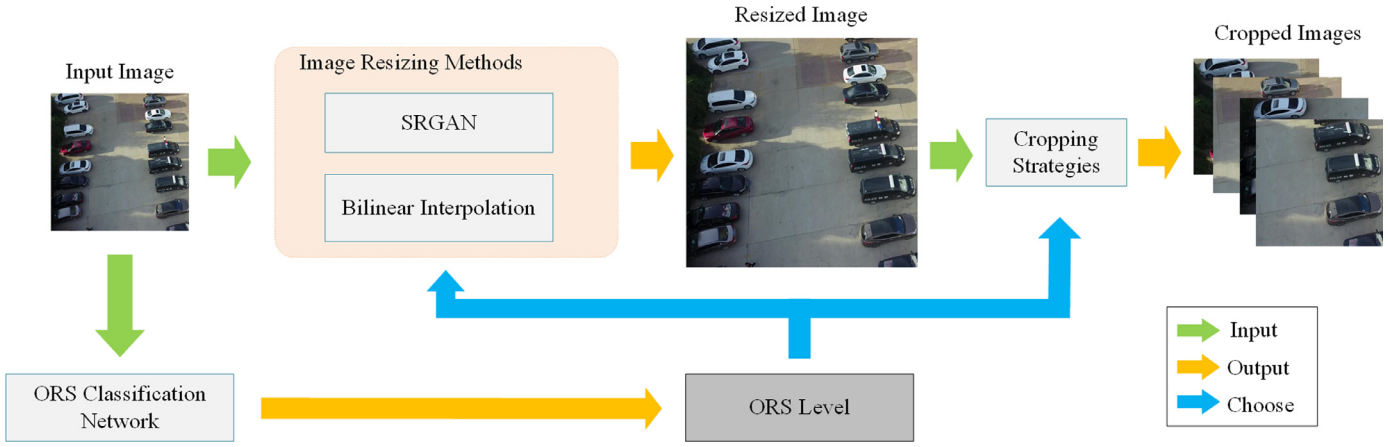


Fig. 4. The flow diagram of SAIC. After SAIC, cropped images are feed into the following detector.

[4–7] aggregate features from multiple layers before prediction. Benefiting from redundant information and high-resolution feature maps, those methods can detect small objects better. However, since they aggregate feature maps only on one scale, their performance is limited for handling scale diversity. The other way is to use feature hierarchy. SSD [8], DSSD [9] and MS-CNN [10] detect objects at multiple layers of the feature hierarchy without combining features. FPN [11], RetinaNet [12] and PANet [13] improve the feature hierarchy by adding pathways to pass information between different layers.

Another components is RoI pooling. PSPNet [15] proposes a pyramid pooling module, obtained the state-of-the-art semantic segmentation result on Cityscapes [27]. GBD-Net [16] and Craft GBD-Net [17] use similar ideas to get better performance in generic object detection. Jifeng et al. [18] introduce deformable network components to further enhance the transformation modelling capability of CNNs. Jiayuan et al. [19] propose to learn region features for generalizing deformable RoI pooling.

Data Enhancement. Data enhancement is another way to address scale problems, which can be combined with model modifications for further performance improvement. Here, we only review methods relative to solve challenges in aerial scenes, i.e. small ORS, low-resolution objects, and object scale diversity

Image cropping is a straightforward way to solve small ORS. YOLT [28] evenly crops the satellite image before detection. LaLonde et al. [29] propose ClasterNet to generate potential regions containing objects and design FoveaNet for finer detection. However, these two methods crop images at a single scale which faces serious scale diversity in UAV scenes. Gao et al. [30] introduce dynamic zoom-in network to enhance data, which relies on detection results on down-sampled images. As objects are too small in aerial images, detection results on down-sampled images are extremely unreliable. Recently, Yang et al. [31] propose ClusDet to simultaneously address the scale and sparsity challenges in UAV object detection. They introduce cluster proposal sub-net and scale estimation sub-net to generate cluster chips. Some ideas in this work are similar to us, so it will be compared with our methods in experiments.

Image up-sampling is a key step to ease low-resolution object detection. Many methods simply use bilinear or cubic interpolation to enlarge images, obtaining high-resolution but low SNR images. In the field of super-resolution, many methods [32–36] are proposed to maximizes the SNR of high-resolution images by minimizing MSE loss. Those methods only fill the low frequency information of the image. Recently, a GAN-based super-resolution method SRGAN [37] is proposed. By introducing adversarial learning and perceptual loss, SRGAN can estimate and fill the missing high-frequency information.

To handle object scale diversity, Bharat Singh and Larry S. Davis [38,39] proposes SNIP/ER to normalize the scale of objects using image pyramid. However, due to high image resolution, the image pyramid of aerial images is too large for current GPU memory. In addition, cropping in the image pyramid introduces potential false alarms.

3. Scale adaptive image cropping

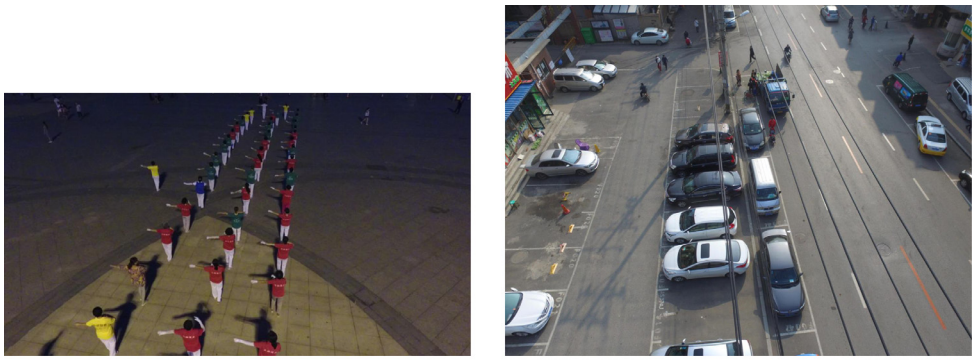
An overview of SAIC is shown in Fig. 4. Images are transmitted from UAV to a server and processed by SAIC in three steps. First, a specific classification network is used to classify ORS level of the input image. Then, based on the estimated level and the image resolution, bilinear interpolation or SRGAN is selected to up-sample the image. Finally, the resized image is cropped by the corresponding cropping strategy. After SAIC, cropped images are fed into the following detector.

3.1. Object relative scale estimation

From our observations, the object scale difference between images is the main reason for object scale diversity in aerial images. SAIC classifies the ORS level of each image, then resizes and corps the image based on the level to reduce the diversity and adjust ORS. To estimate ORS, we first need to define the ORS level, and use it to train a well-designed classification network.

ORS level. The average ORS is a straightforward indicator of image ORS level. The smaller average ORS are directly related to the smaller objects in the image. However, the average ORS ignores category information. When there are various categories with different original sizes in the image, the average ORS can not work well. In this case, adjusting the image size based on the average ORS only reduces inter-class diversity rather than intra-class diversity, which is rarely helpful for detection. As shown in Fig. 5, Fig. 5a and 5 b have the same average ORS. If we resize the image based on the average ORS, these two images will be resized to the same scale, which makes the pedestrian in Fig. 5a as large as the car in Fig. 5b, but much bigger than the pedestrian in Fig. 5b. We want SCAI can reduce both inter-class and intra-class object scale diversity, hence, we need a category-aware indicator to describe image ORS level.

We propose to apply the normalized average ORS, NAORS in short. We divide 10 meaningful categories in VisDrone (excluding ‘ignore area’ and ‘others’) into 4 super-categories according to



(a) (b)

Fig. 5. Two aerial images in VisDrone 2018 DET training dataset. The average ORS of both images is 0.062.

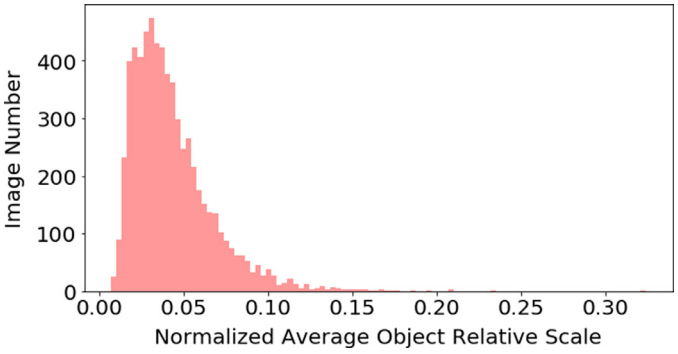


Fig. 6. Distribution of NAORS on VisDrone 2018 DET train dataset.

the original size. 'Pedestrian' and 'people' belong to 'human' super-category. 'Motor' and 'bicycle' are classified as 'small vehicle' super-category. 'Car', 'van', 'tricycle' and 'awning-tricycle' are divided into 'middle vehicle' super-category. 'Bus' and 'truck' belong to 'large vehicle' super-category. NAORS of each image is calculated as

$$NAORS = \sum_{s \in \mathbb{R}_{super}} \frac{n_s * A(s)}{n_{image} * c(s)} \tag{1}$$

where s represents a super-category in the super-category set $\mathbb{R}_{super}$, $A(*)$ represents the average ORS of specified super-category, $c(*)$ represents the normalization coefficient of specified super-category, n_s represents the object number of super-category s, n_{image} represents the object number of the whole image. We use super-category 'middle vehicle' as the reference, because it has the largest number of objects. The normalization coefficient of 'middle vehicle' is set to 1. The normalization coefficient of other super-categories is calculated as

$$c(s) = \frac{1}{|\mathbb{I}_{sub-train}|} \sum_{i \in \mathbb{I}_{sub-train}} \frac{A(s)}{A(middle\ vehicle)} \tag{2}$$

where $|*|$ represents the number of specified collection elements. $\mathbb{I}_{sub-train}$ is a subset of training data, where the images contain both super-category s and 'middle vehicle'.

The NAORS distribution of VisDrone training data is shown in Fig. 6. We define three ORS levels for VisDrone. Images with NAORS at [0, 0.064] are classified to the 'small' level. Images with NAORS at (0.064, 0.085] belong to the 'medium' level. Images with NAORS at (0.085, 1] belong to 'large' level.

NAORS can be used to generate image ORS level for other object detection datasets by simply redefining the super-category and level threshold.

ORS Classification Network. The object scale challenges have little effect on ORS classification task, since this task does not require object localization or classification. Only simpler scene scale variation is required to be noticed. Using features with appropriate receptive field is hopeful to address scene scale variation.

We design a specific network for ORS classification, named ORSC, in which an adaptive receptive structure is proposed to adjust feature receptive field automatically. Adaptive receptive structure combines hyper features with channel attention, using little extra computation to achieve receptive field auto-tuning. Hyper feature is concatenated from multiple feature layers, channels in hyper feature have various receptive fields. Channel attention adjusts the weight of hyper feature channels, making the main receptive field of hyper feature fit the scene scale. The whole structure of ORSC is shown in Fig. 7.

ResNet-50 [40] is adopted as the backbone of ORSC. Following the definition used in FPN, ResNet is divided into five *stages*. Layers producing feature maps with the same size are in the same network *stage*. In our work, the last feature maps of the last 4 *stages*, denoted as {C2, C3, C4, C5}, are concatenated as hyper feature. Feature maps of stage 1 are ignored for saving memory. When concatenating feature maps, {C2, C3, C4, C5} are fed into independent 1×1 convolutional layers to normalize the channel number to 512. This way constructs the hyper feature while avoiding reliance on features with more channels. Normalized features are then max-pooled to the size of C5 and concatenated.

After feature concatenation, channel-wise attention is combined to adjust the channel weight of hyper feature. The modified hyper feature is squeezed and sent into fully connected layers for ORS level classification.

3.2. Image resizing

SAIC resizes the short side of the image to {800, 1200, 1600} pixels according to {'large', 'medium', 'small'} ORS levels.

In order to obtain clear images, we use either the bilinear interpolation or the SRGAN based on the image ORS level and the original image resolution. For images at 'low' and 'medium' level, bilinear interpolation is used for resizing, while SRGAN is adopted to generate high quality enlarged images for which at 'high' level and with small original resolution (short side smaller than 800). Although many MSE based super-pixel methods can recover low-resolution images more accurately, they cannot fill the missing high frequency information. Thanks to adversarial learning, SRGAN can generate more realistic images, which is helpful for detector to distinguish low-resolution objects. Since SRGAN can only upsample images by factor of 4, bilinear interpolation is used for subsequent size adjusting. The details of SRGAN can be found in [37].

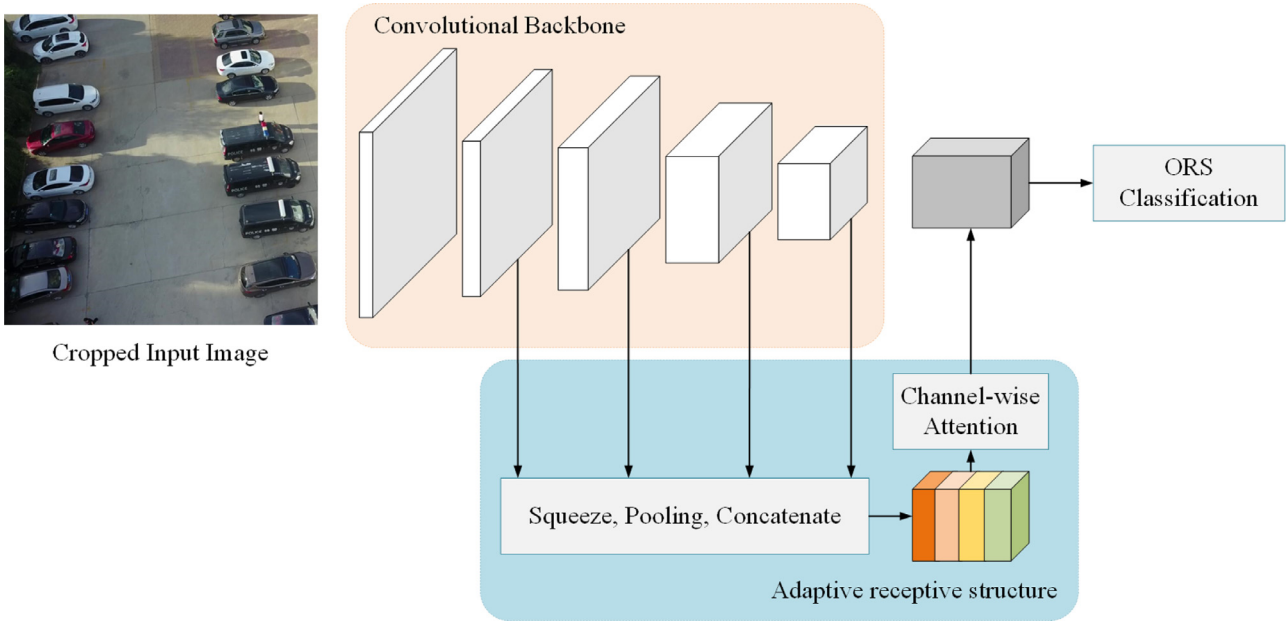


Fig. 7. The flow diagram of ORSC.

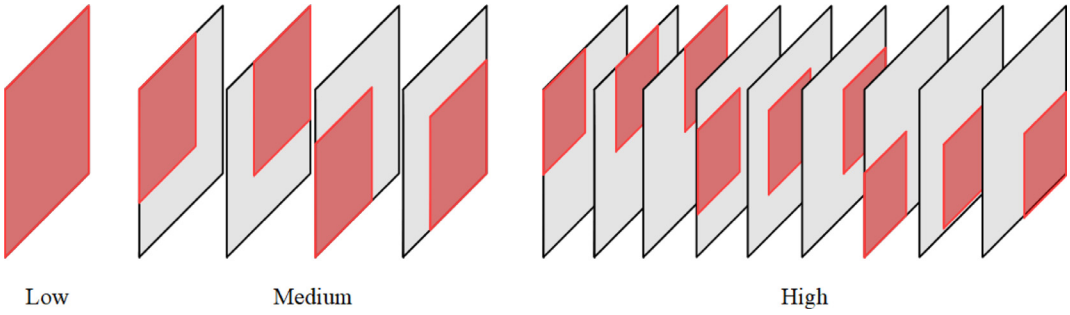


Fig. 8. Cropping strategies of three ORS levels.

3.3. Cropping strategy

To save inference time, we use different cropping strategies at different ORS levels. All cropping strategies are shown in Fig. 8. Images at 'large' level will not be cropped. Images at 'medium' level will be cropped by 4 frames with the same size. Images at 'small' level will be cropped by 9 frames with the same size. In terms of formula, cropping strategies can be expressed as

$$x_{crop} = \frac{id_{col} * W}{crop\_level + 1} \tag{3}$$

$$y_{crop} = \frac{id_{row} * H}{crop\_level + 1} \tag{4}$$

$$w_{crop} = \frac{W}{crop\_level + 1} \tag{5}$$

$$h_{crop} = \frac{H}{crop\_level + 1} \tag{6}$$

where (x_{crop}, y_{crop}) is the top-left coordinate of cropping region, w_{crop} and h_{crop} represent the width and height of cropping region, id_{row} and id_{col} represent the row and column of cropping region, W and H are the image width and height, respectively. *crop_level* is set to {1, 2, 3} corresponding to {'large', 'medium', 'small'} ORS level. Since SAIC resizes the short side of the image to {800, 1200, 1600} pixels, all cropped images have the short side of 800 pixels.

3.4. Training and inference

ORSC and SRGAN are two independent networks in SAIC that cannot be learned end-to-end with the detector. In order to make SAIC and detector cooperate better, a 5-step training process is proposed as shown in Algorithm 1.

Algorithm 1 SAIC training process. After 4 steps, SAIC and detector form a unified detection framework.

1: train ORSC.
2: fine-tune SRGAN.
3: pre-train detector on images with all ORS level.
4: fine-tune detector on images with ORS level classified by ORSC.

First, we train ORSC using ORS level introduced above. Because of using fully connected layers, ORSC takes fixed size images as input in both training and inference stages. In our implementation, images are scaled to 800 on the short side and cropped to 800 × 800 in the centre, which avoids the proportional distortion. ORSC is trained in end-to-end manner, using the cross-entropy loss.

Then we fine-tune SRGAN. SRGAN is pre-trained as [37] and fine-tuned on images at 'large' and 'medium' level. We down-sample 'large' and 'medium' level images to provide low-resolution samples, and use original images as corresponding high-resolution samples. The 'small' level images are excluded since objects in

'small' level images are too small to provide enough information for SRGAN training.

At last we train the detector in two steps. First, we treat each image as having three ORS levels at the same time, and use it to pre-train the detector. Because the training images of VisDrone is not sufficient, if we divide these images into three categories by ORS level and train detector directly on the divided images, the chance of detector meeting various scale objects will be reduced. Secondly, for better cooperation with ORSC, we fine-turn the detector on images with ORS level classified by ORSC.

In the inference stage, images are first scaled and centre cropped to feed into ORSC and estimate ORS level. Then, images are resized according to ORS level, and are cropped by the corresponding cropping strategy. The cropped images are sent to the detector for object detection. Finally, the detection results are merged. Since there are overlaps between cropped images, some objects may have duplicate detection results. NMS or Soft-NMS [41] is adopted to suppress those duplications.

4. Experimental evaluation

4.1. Datasets and evaluation metrics

We perform experiments on a large-scale UAV object detection and tracking benchmark VisDrone 2018 DET dataset. To the best of our knowledge, it is the largest drone image dataset, which is suitable for training and evaluating deep learning methods. The VisDrone provides a detection dataset with 10,209 images in total, 6471 images for training, 548 images for validation and 3190 images for testing.

Top-1 accuracy is used to measure the performance of ORS classification, which is calculated as

$$Acc_{top1} = \frac{\sum \mathbf{1}(\hat{o}_i, o_i)}{n} \tag{7}$$

where $\mathbf{1}(\hat{o}_i, o_i)$ equals 1 when $\hat{o}_i = o_i$ and equals 0 otherwise, n is the number of test images.

Following suggestions in [3], $AP^{IoU=0.50:0.05:0.95}$, $AP^{IoU=0.50}$, $AP^{IoU=0.75}$ are used to measure the performance of our detection framework. These criteria penalize both missing detection of objects and duplicate detections (multiple detection results for the same object). $AP^{IoU=0.50:0.05:0.95}$ is computed by averaging over all 10 intersection over union (IoU) thresholds of all categories (i.e., in the range [0.50 : 0.95] with the uniform step size 0:05), used as the primary metric. All *AP*s are calculated within a maximum of 500 detected bounding boxes per image.

4.2. Implement details

We train all networks on 8 NVIDIA GTX 1080Ti GPUs, using mini-batch SGD as the optimization method. When training ORSC, we set the total number of iterations to 5000, the mini-batch size to 32 and the beginning learning rate to 0.0001. The learning rate is reduced by 0.1 at iteration 2500 and 3750. We use SRGAN pretrained as [37] and fine-tune it slightly.

We train detectors using the detectron [42] implementation. FPN without SAIC are trained as baseline. ResNeXt-101 is adopted as backbone. Since objects are too small to match anchors in P6, the loss of P6 is always zero. For better back propagation, P6 is removed from FPN. The image mini-batch size is set to 1. The ROI mini-batch size is set to 512. The total number of iterations is 4 epochs, i.e. 25200. The initial learning rate is set to 0.01 for 8 GPUs training and is reduced by 0.1 at iteration 16,800 and 22400. When training FPN with SAIC, FPN is first pretrained using above setting. Then, FPN is fine-tuned by additional 8400 iterations with learning rate at 0.0001.

Table 1
Top-1 accuracy of ORS classification.

Method	Hyper feature	Channel-wise attention	Adaptive receptive structure	Top-1 Acc[%]
resnet50				86.50
hyper50	✓			86.31
senet50		✓		87.59
ORSC			✓	**89.23**

Table 2
Ablation evaluation of SAIC using FPN as the baseline detector.

Method	*AP*[%]	AP_{50}[%]	AP_{75}[%]
FPN	26.04	45.96	25.56
SC-FPN_s	28.60	52.06	27.45
SC-FPN_m	31.96	56.70	31.20
SC-FPN_l	33.18	58.40	32.65
PC-FPN	34.35	60.60	33.87
SAIC-FPN w/o SRGAN	35.13	61.98	34.53
SAIC-FPN(DE-FPN)	**35.69**	**62.97**	**35.08**

4.3. ORS classification

To demonstrate the effectiveness of adaptive receptive structure in ORSC, the performance of four classifiers are compared in Table 1. ResNet-50 is chosen as a baseline classifier which obtains 86.50% in top-1 accuracy. As we can see, adding only hyper feature does not improve performance. Despite there are multiple receptive fields in hyper feature, the main receptive field of it may not fit the scene scale. Meanwhile, adding only channel attention just brings a bit gains. Because the last feature map only has a single receptive field, adjusting channel weight cannot change the main receptive field of it. By combining hyper feature and channel-wise attention, adaptive receptive structure helps ORSC obtains 89.23% in top-1 accuracy, gains improvement by 2.73% (3.16% relatively) compared with vanilla ResNet50.

4.4. Ablation experiments

Ablation experiments are conducted on validation set to demonstrate the effectiveness of each module in SAIC, as shown in Table 2. We select FPN as the baseline detector. The short side of image is resized to 800 before fed into the baseline detector. Suffering from serious object scale problems, vanilla FPN only achieves 27.77% *AP*. Image cropping can alleviate small ORS problem. We train FPNs using single scale cropped images, noted as SC-FPN, where {SC-FPN_s, SC-FPN_m, SC-FPN_l} means the short side of input image is resized to {800, 1200, 1600}, respectively. SC-FPN_s achieves 28.60% *AP* improved baseline accuracy by 2.56% (9.83% relatively). SC-FPN_m achieves 31.96% *AP* improved baseline accuracy by 5.92% (22.73% relatively). SC-FPN_l achieves 33.18% *AP* improved baseline accuracy by 7.14% (27.41% relatively). Cropping images at a single scale faces serious scale diversity problem. We train FPNs using cropped images from image pyramid {800, 1200, 1600}, noted as PC-FPN. PC-FPN achieves 34.35% *AP* improved baseline accuracy by 8.31% (31.91% relatively). However, cropping in image pyramid introduces lots of false alarms. SAIC crops images based on ORS level, solving both above problems effectively and achieving 35.13% *AP*. By adopting SRGAN, *AP* is further improved by 0.56%. At last, SAIC-FPN surpasses vanilla FPN by 9.65% (37.06% relatively).

The visualized comparison between FPN, PC-FPN, and SAIC-FPN is shown in Fig. 9. The first three rows show the results on images with large ORS, the middle three rows show the results on images with medium ORS, and the last three rows show the results on images with small ORS. For all cases, only results with confidence

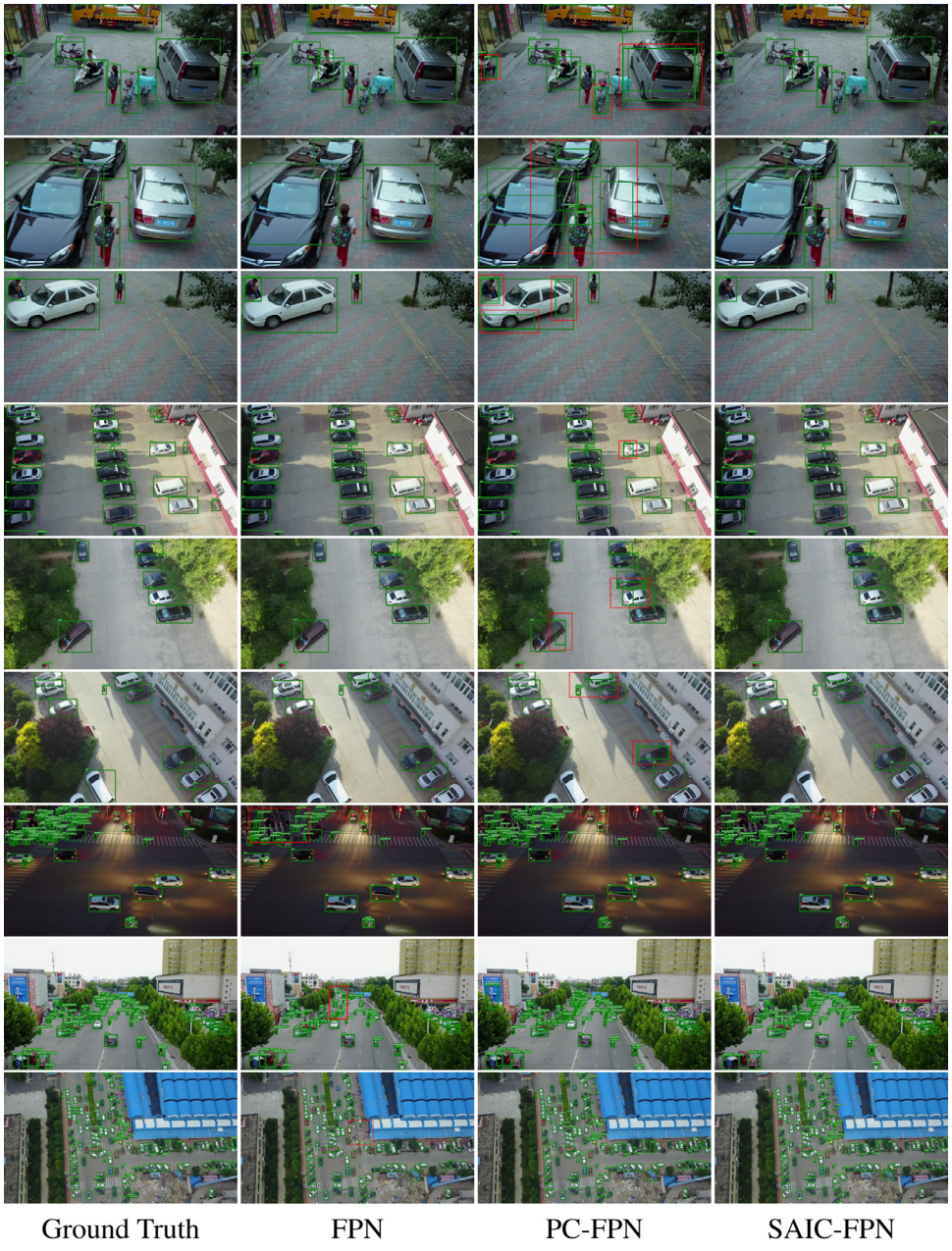


Fig. 9. Visualized comparison between FPN, PC-FPN, and SAIC-FPN.

greater than 0.5 are plotted. It can be seen that FPN and SAIC-FPN performance well for images with large and medium ORS, while PC-FPN introduces many false alarms due to multi-scale inference, shown in red regions. For images with small ORS, the accuracy of FPN is significantly reduced due to small objects challenge, in which a lot of object are missed, shown in red regions. Benefited by scale adaptive cropping, SAIC-FPN can recall most of small objects and simultaneously reduces false alarms.

4.5. Inference time

We measure the inference time of various methods on a single GTX 1080Ti GPU. The inference time of ORSC is about 0.012 s. The inference time of SRGAN is about 0.395 s. The time of image cropping and result merging is so short that can be ignored. The total time of processing one image by various methods is shown in Table 3.

Table 3
Inference time comparison.

Method	Inference time per image
FPN	0.240 s
SC-FPN_s	0.612 s
SC-FPN_m	0.960 s
SC-FPN_l	1.348 s
PC-FPN	2.920 s
SAIC-FPN w/o SRGAN	0.252s ∼ 2.172 s
SAIC-FPN(DE-FPN)	0.252s ∼ 2.568 s

Since the vanilla FPN does not need to crop the image and processes the entire image at once, it has the shortest inference time. For a single scale crop, the inference time increases as the image scale increases. Cropping the image in image pyramid significantly increases the number of inference for one image, leading to the longest inference time. SAIC-FPN resizes and crops images based

Table 4
Compared with ClusDet, where * denotes the multi-scale inference and bounding box voting are utilized in test phase.

Method	AP[%]	AP_{50}[%]	AP_{75}[%]
ClusDet [31]	28.4	53.2	26.4
ClusDet * [31]	32.4	56.2	31.6
SAIC-FPN w/o SRGAN	35.13	61.98	34.53
SAIC-FPN(DE-FPN)	**35.69**	**62.97**	**35.08**

Table 5
Comparative evaluation of SAIC-FPN (DE-FPN) on VisDrone 2018 DET test set.

Method	AP[%]	AP_{50}[%]	AP_{75}[%]
HAL-Retina-Net	31.88	46.18	32.12
DPNet	30.92	54.62	31.17
SAIC-FPN(DE-FPN)	**27.10**	**48.72**	**26.58**
CFE-SSDv2	26.48	47.30	26.08
RD^{4}MS	22.68	44.85	20.24
L-H RCNN+	21.34	40.28	20.42
Faster R-CNN2	21.34	40.18	20.31
RefineDet+	21.07	40.98	19.65
DDFPN	21.05	42.39	18.70
YOLOv3_DP	20.03	44.09	15.77
MFaster-RCNN	18.08	36.26	16.03
MSYOLO	16.89	34.75	14.30
DFS	16.73	31.80	15.83
FPN2	16.15	33.73	13.88
YOLOv3+	15.26	30.06	12.50
IITH DODO	14.04	27.94	12.67
FPN3	13.94	29.14	11.72
SODLSY	13.61	28.41	11.66
FPN	13.36	27.05	11.81

on the ORS level, so the number of detection is uncertain, leading to the uncertain inference time from 0.252s to 2.568 s.

4.6. Comparative experiments

Compared with content-based cropping methods. Four content-based cropping methods [28–31] are reviewed in related work. Cropping methods in YOLT [28] and ClusterNet [29] are tightly integrated with a detector so that they may not work well for new architectures. Data enhancement methods should be general, treating the detector as a black box and optimizing its accuracy.

There are some general methods [30,31] in the literature but their codes are not publicly available. Any mistake in third party implementation will lead to an unfair comparison. Fortunately, Yang et al. [31] conduct their experiments on Visdrone 2018 DET validation set, using the same detector with the same backbone as ours. Therefore, their methods can be fairly compared with ours, whose results are shown in Table 4. Our methods surpass ClusDet by 3.29 (10.15% relatively), probably because they focus on mitigating the scale differences inside the image. However, object scale diversity in aerial images is mainly caused by scale differences between images.

Compared with state-of-the-art methods. Thanks to the literature [3], we obtain the performance of SAIC-FPN (or data enhanced FPN, DE-FPN) on VisDrone 2018 DET test set, which is compared with other state-of-the-art methods in Table 5.

Without any major modification (just remove the P6 from FPN), SAIC-FPN achieves the 3rd rank on VisDrone 2018 DET test set. Note that SAIC-FPN surpasses the baseline FPN (given by the literature [3]) by 13.74% (102% relatively), showing the huge potential of SAIC. SAIC is a general data enhanced method for UAV object detection, which means it can be combined with other state-of-the-art detectors and help them to get further accuracy improvement.

5. Conclusion

We present SAIC, a scale adaptive data enhancement method, for handling severe scale challenges in UAV object detection. Based on the observation that object scale diversity in aerial images is mainly caused by scale differences between images, SAIC first classifies ORS level of images, then resizes images based on the estimated ORS level, and finally crops the resized images. In ORS estimation step, we proposed NAORS as an category aware indicator for image ORS level. Also, a well-designed classification network is proposed in which an adaptive receptive structure is introduced to handle scene scale problems. In image resizing step, we adopt SRGAN to up-sample images with the smallest ORS level, making detecting low-resolution objects easier. Without any major model modification, FPN trained with SAIC achieves state-of-the-arts performance on VisDrone 2018 DET.

However, SAIC-FPN is a bit time-consuming in inference stage (about 0.252 s $\sim$ 2.568 s per image), even we use different cropping strategies for different ORS levels. In future work, we will explore a more flexible cropping method and also hope to combine the super-resolution task with detection task, so that the whole framework can be trained end-to-end.

Declaration of Competing Interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

References

[1] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, L. Fei Fei, Imagenet: a large-scale hierarchical image database, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.
[2] T.Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll'ar, C.L. Zitnick, Microsoft coco: common objects in context, in: Proceedings of the European Conference on Computer Vision (ECCV), 2014.
[3] P. Zhu, L. Wen, X. Bian, L. Haibin, Q. Hu, Vision meets drones: a challenge, arXiv:1804.07437 (2018).
[4] T. Kong, A. Yao, Y. Chen, F. Sun, Hypernet: towards accurate region proposal generation and joint object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[5] J. Mao, T. Xiao, Y. Jiang, Z. Cao, What can help pedestrian detection? in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
[6] W. Liu, A. Rabinovich, A.C. Berg, Parsenet: looking wider to see better, arXiv:1506.04579 (2015).
[7] S. Bell, C. Lawrence Zitnick, K. Bala, R. Girshick, Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.Y. Fu, A.C. Berg, SSD: single shot multibox detector, in: Proceedings of the European Conference on Computer Vision (ECCV), 2016.
[9] C.Y. Fu, W. Liu, A. Ranga, A. Tyagi, A.C. Berg, Dssd: deconvolutional single shot detector, arXiv:1701.06659 (2017).
[10] Z. Cai, Q. Fan, R.S. Feris, N. Vasconcelos, A unified multi-scale deep convolutional neural network for fast object detection, in: Proceedings of the European Conference on Computer Vision (ECCV), 2016.
[11] T.Y. Lin, P. Doll'ar, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
[12] T.Y. Lin, P. Goyal, R. Girshick, K. He, P. Doll'ar, Focal loss for dense object detection, in: Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018.
[13] S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path aggregation network for instance segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
[14] J. QiongYan, Y. LiXu, Accurate single stage detector using recurrent rolling convolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
[15] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[16] X. Zeng, W. Ouyang, B. Yang, J. Yan, X. Wang, Gated bi-directional CNN for object detection, in: Proceedings of the European Conference on Computer Vision (ECCV), 2016.
[17] X. Zeng, W. Ouyang, J. Yan, H. Li, T. Xiao, K. Wang, Y. Liu, Y. Zhou, B. Yang, Z. Wang, et al., Crafting GBD-net for object detection, IEEE Trans. Pattern Anal. Mach. Intell. 40 (9) (2018) 2109–2123.
[18] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, Deformable convolutional networks, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017.
[19] J. Gu, H. Hu, L. Wang, Y. Wei, J. Dai, Learning region features for object detection, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018.
[20] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
[21] I. Ševo, A. Avramović, Convolutional neural network based automatic object detection on aerial images, IEEE Geosci. Remote Sens. Lett. 13 (5) (2016) 740–744.
[22] N. Audebert, B. Le Saux, S. Lefèvre, Segment-before-detect: vehicle detection and classification through semantic segmentation of aerial images, Remote Sens. 9 (4) (2017) 368.
[23] Z. Deng, H. Sun, S. Zhou, J. Zhao, H. Zou, Toward fast and accurate vehicle detection in aerial images using coupled region-based convolutional neural networks, IEEE J. Select. Top. Appl. Earth Observ. Remote Sens. 10 (8) (2017) 3652–3664.
[24] L.W. Sommer, T. Schuchert, J. Beyerer, Fast deep vehicle detection in aerial images, in: Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), 2017.
[25] R. Girshick, Fast R-CNN, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015.
[26] S. Zhang, G. He, H.-B. Chen, N. Jing, Q. Wang, Scale adaptive proposal network for object detection in remote sensing images, IEEE Geosci. Remote Sens. Lett. (2019).
[27] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The cityscapes dataset for semantic urban scene understanding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[28] A. Van Etten, You only look twice: rapid multi-scale object detection in satellite imagery, arXiv:1805.09512 (2018).
[29] R. LaLonde, D. Zhang, M. Shah, Clusternet: detecting small objects in large scenes by exploiting spatio-temporal information, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
[30] M. Gao, R. Yu, A. Li, V.I. Morariu, L.S. Davis, Dynamic zoom-in network for fast object detection in large images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
[31] F. Yang, H. Fan, P. Chu, E. Blasch, H. Ling, Clustered object detection in aerial images, arXiv:1904.08008 (2019).
[32] C. Dong, C.L. Chen, K. He, X. Tang, Learning a deep convolutional network for image super-resolution, in: Proceedings of the European Conference on Computer Vision (ECCV), 2014.
[33] Y. Chen, Y. Tai, X. Liu, C. Shen, J. Yang, FSRNET: end-to-end learning face super-resolution with facial priors, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
[34] W. Shi, J. Caballero, F. Huszar, J. Totz, A.P. Aitken, R. Bishop, D. Rueckert, Z. Wang, Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[35] J. Kim, J.K. Lee, K.M. Lee, Accurate image super-resolution using very deep convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[36] B. Lim, S. Son, H. Kim, S. Nah, K. Mu Lee, Enhanced deep residual networks for single image super-resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2017.
[37] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, Photo-realistic single image super-resolution using a generative adversarial network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
[38] B. Singh, L.S. Davis, An analysis of scale invariance in object detection - snip, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
[39] B. Singh, M. Najibi, L.S. Davis, Sniper: efficient multi-scale training, in: Proceedings of the Conference on Neural Information Processing Systems (NIPS), 2018.
[40] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[41] N. Bodla, B. Singh, R. Chellappa, L.S. Davis, Soft-NMS improving object detection with one line of code, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017.
[42] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, K. He, Detectron, https://github.com/facebookresearch/detectron (2018).

Jingkai Zhou received the B.E. degree in Software Engineering from South China University of Technology in 2015. He is currently a Ph.D. candidate under Professor Qiong Liu. His research interests include object detection and object tracking.

Chi-Man Vong received the M.S. and Ph.D. degrees in Software Engineering from the University of Macau in 2000 and 2005, respectively. He is currently an Associate Professor with the Department of Computer and Information Science, University of Macau. His research interests include machine learning methods and intelligent systems.

Qiong Liu received the B.E. degree in Automation from Tsinghua University in 1982, the M.S. degree in Automation from Chongqing University in 1988, and the Ph.D. degree in Biomedical Engineering from Chongqing University in 1996. She is currently a Professor with the School of Software, South China University of Technology. Her research interests include object detection, object tracking, panoptic segmentation, and model compression.

Zhenyu Wang received the B.S. degree in computer science from Xiamen University in 1987, and the M.S. and Ph.D. degrees in computer science from the Harbin Institute of Technology in 1990 and 1993, respectively. He is currently a Professor and the Dean of the School of Software, South China University of Technology. His research interests include distributed computing and SOA, operating systems, software engineering, and large-scale application design and development.