Computational Visual Media
https://doi.org/10.1007/s41095-019-0132-5 Vol. 5, No. 2, June 2019, 221–228
Research Article

Livestock detection in aerial images using a fully convolutional network

Liang Han[1] (✉), **Pin Tao**[2], **and Ralph R. Martin**[3]

Abstract In order to accurately count the number of animals grazing on grassland, we present a livestock detection algorithm using modified versions of U-net and Google Inception-v4 net. This method works well to detect dense and touching instances. We also introduce a dataset for livestock detection in aerial images, consisting of 89 aerial images collected by quadcopter. Each image has resolution of about 3000 × 4000 pixels, and contains livestock with varying shapes, scales, and orientations.

We evaluate our method by comparison against Faster RCNN and Yolo-v3 algorithms using our aerial livestock dataset. The average precision of our method is better than Yolo-v3 and is comparable to Faster RCNN.

Keywords livestock detection; segmentation; classification

1 Introduction

Forage–livestock balance is an important factor affecting the productivity of grassland. Having an appropriate number of livestock on grassland is important for sustainable animal husbandry. If the stocking rate is too high, the grassland will be overutilized, and the grassland ecology will deteriorate, which is not conducive to the production of livestock. Accurately determining the actual number of livestock present on grassland is necessary for macro-control, and is needed for government surveillance of overgrazing to prevent grassland ecological degradation. Therefore, an efficient and accurate method for detecting livestock is needed to obtain the actual number of animals grazing on the grassland.

In recent years, deep convolutional neural networks have been used widely for computer vision tasks, including image classification, object recognition, and semantic segmentation. They surpass traditional methods for many visual recognition tasks. However, the application of deep neural networks to visual recognition tasks cannot be separated from supporting datasets. The release of publicly available datasets is a factor driving the advance of deep convolutional neural networks in the field of computer vision, allowing researchers to develop, evaluate, and compare new algorithms. Many conventional target detection datasets exist, such as PASCAL VOC [1], COCO [2], ImageNet [3], LabelMe [4], etc. Many state-of-the-art target detection algorithms such as Faster R-CNN [5], Yolo [6], SSD [7], Mask R-CNN [8], etc., have been evaluated using these conventional datasets. However, object detectors based on conventional datasets do not perform well on aerial images, the main reason being that aerial images have their own particularities. Firstly, the views have a specific nature: aerial images are usually taken from high altitude, while conventional datasets are mostly ground-level views, so the appearance of the same target is quite different. Secondly, the target resolution is low. In aerial images, most targets occupy relatively few pixels, providing little feature information. The CNN pooling layer further reduces this information. After 4 pooling layers, a 20×20 pixel target has only about 1 pixel, making it difficult to detect small targets. Thirdly, orientation is arbitrary.

1 Department of Computer Technology and Application, Qinghai University, Xining, China. E-mail: hanl2010@qq.com (✉).
2 Department of Computer Science and Technology, Tsinghua University, Beijing, China. E-mail: taopin@tsinghua.edu.cn.
3 School of Computer Science and Informatics, Cardiff University, Cardiff, Wales, UK. E-mail: MartinRR@cardiff.ac.uk.

Manuscript received: 2018-12-12; accepted: 2019-01-27

Aerial images are taken from above, and the heading of the target is arbitrary, unlike conventional datasets, in which for example pedestrians are generally upright. The target detector thus needs to be insensitive to direction. For the above reasons, specialized aerial image datasets are needed for training and assessing aerial image target detection methods, and target detection algorithms need to take these characteristics into account.

In this paper, we first introduce an aerial livestock dataset for use in the design and assessment of algorithms to detect and count livestock on grassland. The dataset is divided into three parts according to the time, place, and difficulty of detection of livestock in the images. Secondly, we introduce a livestock detection algorithm using modified versions of U-net [9] and Google Inception-v4 net [10]. U-net is used to segment the aerial image to obtain regions of interest (ROIs). Since the feature map has the same resolution as the original image, feature information for small targets can be retained. The modified Inception-net is used to classify each ROI to accurately identify targets. We comparatively evaluate our algorithm against Faster RCNN [5] and Yolo-v3 [6] algorithms using our dataset. Experimental results show that image segmentation using a fully convolutional neural network is beneficial to small target recognition and the average precision of our method is better than that of Yolo-v3 and is comparable to that of Faster RCNN.

2 Related work

2.1 Aerial image dataset

Many new aerial image datasets have been recently produced and made publicly available. The targets in these datasets are mainly land vehicles, ships, aircraft, etc. The DOTA [11] dataset contains 2806 aerial images, each of size about 4000 × 4000 pixels. It contains objects having different scales, orientations, and shapes, in 15 common object categories. The fully annotated DOTA dataset contains 188,282 instances, each of which is labeled by an oriented (i.e., not axis-aligned) bounding box. Most instances are between 10 and 50 pixels in size. Similarly, NWPU VHR-10 [12] contains ten classes of objects, with about 3000 instances in total. The UCAS-AOD (Dataset of Object Detection in Aerial Images) [13] contains a vehicle dataset and a plane dataset collected from Google Earth aerial images. The former contains 310 images with 2819 vehicles; the latter contains 600 images with 3210 planes. Other aerial image datasets contain only one category and are fine-grained: TAS [14], VEDAI [15], COWC [16], and DLR 3K Munich Vehicle [17] focus on land vehicles, while HRSC2016 [18] contains ships. These aerial image datasets are appropriate for the detection of vehicles, ships, and aircraft, etc., but not for the detection of livestock. Therefore, we constructed an aerial livestock dataset for livestock detection.

2.2 Region detection

In 2010, Cheng et al. [19] proposed a method for finding and editing approximately repeated elements, which is based on contour detection; it can detect all image elements similar to a selected one. This method needs manual selection of one element; elements have obvious contour features. In 2012, Krizhevsky et al. [20] applied a convolutional neural network (CNN) to image classification for the first time, showing good results for both image classification and target location tasks. In 2014, Girshick et al. [21] proposed the Region CNN (RCNN) algorithm , which extracts candidate targets that may be objects; it identified them with an accuracy of 58.5%. Girshick [22] later proposed Fast R-CNN, which shortened the time to process an image to 2–3 s, achieving an accuracy of 78.8%. In 2015, Ren et al. [5] used a Region Proposal Network (RPN) for extraction of candidate target blocks, and shared the convolutional layer parameters of RPN and Fast R-CNN, leading to Faster RCNN. To further improve efficiency, Redmon et al. [6] proposed the Yolo (You only look once) network in 2016, which has an average accuracy of 64.3% on the VOC2007 dataset. SSD [7] uses both lower and upper feature maps for detection, with an average accuracy of 74.3% on the VOC2007 dataset. Yolo-v2 [23] achieves an average accuracy of 78.6% by adding batch normalization and increasing image resolution. Using multi-scale prediction and darknet-53, Yolo-v3 [24] further enhanced the ability to detect small targets and increased detection speed. In addition to pixel features, Zhang et al. [25] applied motion, saliency, and other information about targets to detect visual distractors using SegNet [26]. However, these methods have only been evaluated on conventional datasets such as Pascal VOC rather than aerial image datasets. Improved methods exist

that target and have been evaluated on aerial image datasets. Zhu et al. [13] used a coarse localization plus fine classification method on the UCAS-AOD dataset for small target detection, with good results, but most of the targets in this dataset are between 120×120 and 360×360 pixels, much larger than the targets in our dataset. Sakla et al. [27] used the modified Faster R-CNN on the VEDAI dataset to detect small vehicles, again achieving good results, but the targets in this dataset are sparse, and the targets in our dataset are much more dense. Therefore, we need to design an algorithm suitable for dense small target detection.

3 The image dataset

3.1 Images

Our aerial images of livestock were taken over grassland using a quadcopter. The dataset is divided into three parts according to the time, place, and difficulty of detection of livestock in the images. Most livestock in images in Part I of the dataset are black, and the background is relatively simple, with few confusing factors; it contains 29 images. Most livestock in Part II are also black, but there are geomorphological interference factors in the background; it contains 17 images. The livestock in Part III have different colors (black, white, gray, etc.) and sizes, and are close to each other and may even touch or overlap; there may also be confusing factors such as snow, a few houses, and landforms in the image. Thus Part III has a higher detection difficulty than the Parts I and II; it includes 43 images. Three examples from the dataset are shown in Fig. 1.

3.2 Targets and annotation

Each target has a resolution of between 20×20 and 40×40 pixels. They have varying colors and random directions. We do not consider the color, direction, or category of livestock, but just label each instance as livestock using an axis-aligned rectangular bounding box. Image information and bounding boxes are saved in the same format as for the Pascal VOC dataset. A total of 4996 instances were annotated across all images. We did not divide the data into a training set and test set for the datasets of Parts I and II, but did so for the dataset of Part III: 36 images were used as the training set, and 7 images were used as the test set. In this paper, we only used the dataset of Part III for training and testing, because it has a higher detection difficulty than Parts I and II.

4 Livestock detection

4.1 Network design

We use the method of region proposal and classification to detect livestock, following the same approach as R-CNN. We use U-net to perform pixel-level segmentation of the image, acting as a region proposal network (RPN) to generate regions of interest. We use U-net because the feature map it outputs has the same size as the original image, so features for small targets can be preserved as well as possible. We then send the regions of interest to Google Inception-v4 net [10] for classification. Inception-v4 net has faster convergence and higher precision than ResNet [25] and VGG Net [26]. Our pipeline is shown in Fig. 2.

4.1.1 Obtaining regions of interest

We use U-net to score each pixel in the image, and preserve regions with high scores as regions of interest. Instead of using a fixed threshold score for selecting ROIs, we map the score of each pixel to 0–255, as a grayscale image, and then use an adaptive threshold segmentation method to segment the ROIs.

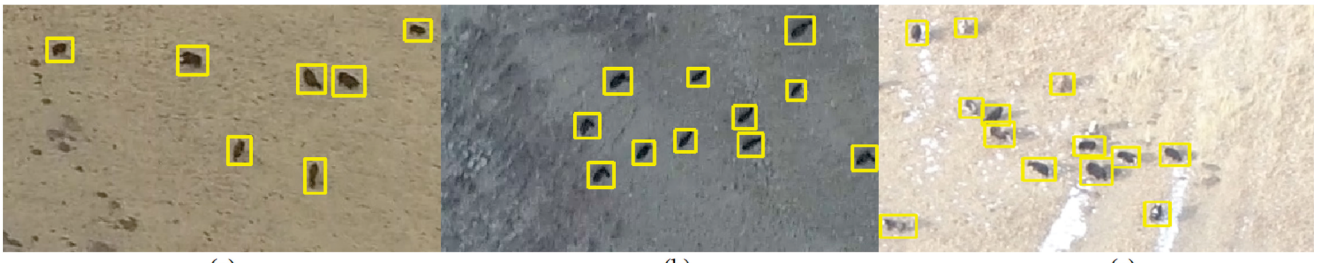


Fig. 1 Examples taken from the livestock dataset. (a)–(c) are cropped examples from Parts I–III of the dataset respectively.

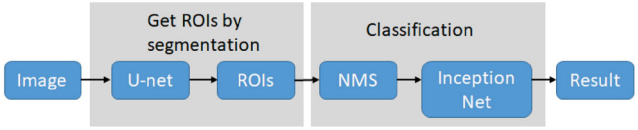


Fig. 2 Object detection pipeline.

4.1.2 Classification

Like Faster RCNN, we generate 9 anchors for each pixel in the segmentation results, with anchor size set to {20, 30, 40}. Non-maximum suppression (NMS) is then applied to the resulting anchors. The score of an anchor is the average of all pixel scores within it. The IOU threshold is set to 0.1, in order to remove redundant regions as much as possible. We map the remaining anchors to the input image, scale each captured image to the same size (50×50 pixels), and use the classification network (Inception net) to classify the proposed regions. We multiply the classification score by the anchor score to give the final score of each proposed region.

4.1.3 U-net modification

U-net [9] is a fully convolutional neural network originally used for biomedical image segmentation; it has also been used for other scenes. The original U-net uses valid padding in the convolutional layer because the boundary of the input image is mirrored and the input image size is changed from 388×388 to 572×572 pixels. It consists of a contraction path and an expansion path. Each convolution operation reduces the feature map in the contraction path, and then upsamples the feature map in the expansion path, concatenating the feature map cropped from the contraction path to the upsampled feature map in the expansion path. The purpose of this is to be able to predict cell edges more accurately in border regions of the image. However, we wish to use U-net for target segmentation instead of edge segmentation. Thus, we do not mirror the boundary of the input image, and change all convolutional layer padding methods to use the same padding. We concatenate the two features directly, without cropping features from the contraction path. The input size of the network is set to 224×384. Our modified U-net network structure is shown in Fig. 3.

4.1.4 Inception-net modification

A structure called Inception is used with GoogleNet to increase the depth and width of the DCNN and improve its performance. The Inception architecture

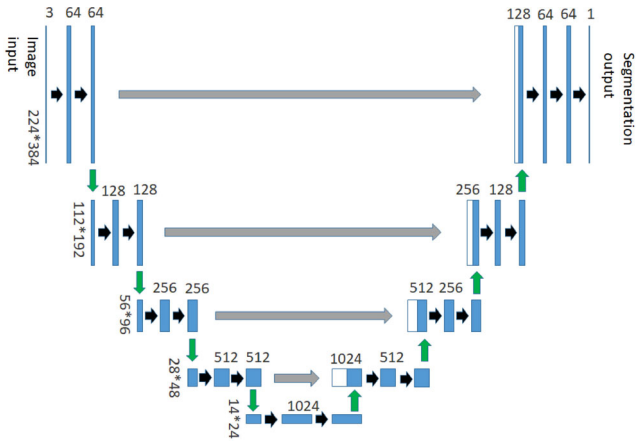


Fig. 3 Modified U-net network structure.

can achieve high performance with low computational cost. The latest Inception-v4 [10] network achieves a 3.8% top-5 error on the ImageNet dataset. Inception-v4 uses three main inception modules, called inception-A, inception-B, and inception-C, and uses reduction-A and reduction-B modules to reduce the feature map. The original Inception-v4 uses one stem module, four inception-A modules, one reduction-A module, seven inception-B modules, one reduction-B module, and three inception-C modules, and then a fully connected layer at the end of the network, giving scores of 1000 classes through a softmax layer. As the regions we wish to classify have low resolution (50×50 pixels) and only need to be divided into two classes, our classification task is simpler than the one in the LSVRC challenge, so we remove the stem module, and the image to be classified is directly input into the inception-A module. Only one inception-A, one inception-B, and one inception-C modules are used in the network structure. We retain the reduction-A and reduction-B modules, and change the number of output classes to 2. Our modified Inception network structure is shown in Fig. 4.

4.2 Training

4.2.1 Inception-net training

Data augmentation is critical to making the network invariant and robust. Positive samples are expanded by flipping vertically, horizontally, and across both axes. Generally, negative samples are selected randomly in the image, chosen so that they do not overlap positive samples. However, we also need to select some difficult negative samples to ensure high quality classification results. The strategy we adopt

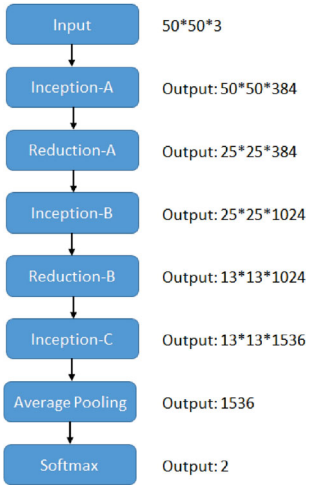


Fig. 4 Modified Inception network structure.

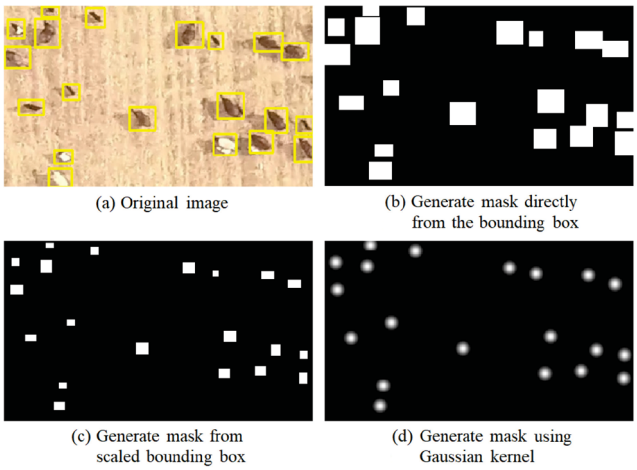


Fig. 5 Masks generated by three different methods.

is to use adaptive threshold segmentation to segment darker areas and brighter areas in the image which are similar to the livestock in color and shape. After thresholding segmentation, many negative samples are obtained, from which we randomly select some. We also randomly select some negative samples from other parts of the image too. A total of approximately 13,000 samples were obtained, 90% of which were randomly selected as the training set, and the remainder used as the test set. After training, the modified Inception-net achieved an accuracy of 94.76%.

4.2.2 U-net training

Since the resolution of the original images is about 3000×4000 pixels, they are too large for training the U-net directly. So we split each original image into 224×384 pixel images. Such images that do not contain livestock are then removed, finally leaving 2716 images from the original training dataset and 348 images from the original test dataset.

Each instance in the aerial livestock dataset is only labeled with a rectangle, and does not have a precise outline, so it is necessary to generate a segmentation mask according to the bounding box. If the area of the bounding box is directly used as the foreground and the other area is used as the background, some masks of instances will touch or even overlap. Figure 5(b) shows an example of masks of instances generated directly from the bounding box. If the bounding box is properly scaled down and the reduced bounding box is used as the foreground area, the resulting masks will not touch, but the masks of some smaller instances can become so small that these instances are lost when segmenting. Figure 5(c) shows an example of generating masks of instances from scaled bounding boxes. In order to ensure that masks of all instances do not touch, and that smaller instances have similar masks to larger instances, we fill the bounding box using a Gaussian kernel whose center overlaps the center of the bounding box. This ensures that the masks of all instances in the image are separate, and the masks of smaller instances are not much different from the masks of larger instances. Figure 5(d) shows an example of generating masks using Gaussian kernels.

In this way, image segmentation is no longer a classification problem, but becomes a regression problem, so we use mean square error (MSE) as the loss function, which is defined as

$$L(y, \widehat{y}) = \frac{1}{n} \sum_{i=0}^{m} (y - \widehat{y})^2 \tag{1}$$

where y is the mask of the input image, and $\widehat{y}$ is the segmentation output of U-net.

5 Experiments

Figure 6 shows results of processing two examples from the test set from segmentation to recognition. It can be seen that our method detects most livestock instances. The livestock in the original image in Fig. 6(a) have varying colors, random directions, and different sizes. The left image of Fig. 6(a) has some snow patches in the background that look like white livestock. The livestock in the right image of Fig. 6(a) are very dense, and some livestock are even touching. Figure 6(b) shows the result of segmentation by U-net and Gaussian filtering. Most

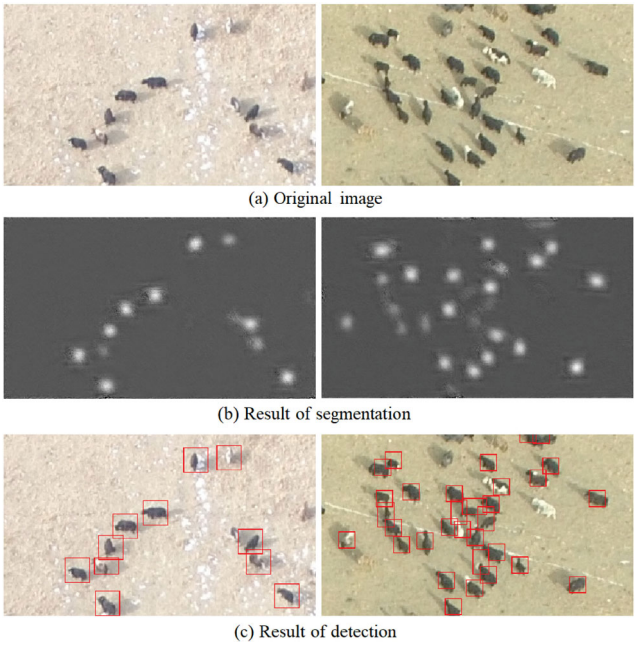


Fig. 6 Processing example images from segmentation to detection.

instances are correctly segmented. Some instances have obvious segmentation results, others have weak segmentation results, and touching instances are segmented with unclear boundaries. The segmentation results are used to generate anchors, and then classification is performed, so that instances touching each other can be separated. Figure 6(c) shows the results of detection. Some white and gray instances remain undetected, mainly because they have are been clearly segmented (see Fig. 5(b)). We also trained Faster RCNN and Yolo-v3 using the same training set and test set for comparison against our method. Our method provides better detection results for the densest instances and touching instances than Faster RCNN and Yolo-v3, as can be seen by comparing Fig. 6(c) and Fig. 7. Figure 8 shows the $P-R$ curve and performance for the three methods: again our method is better than the Yolo-v3 and Faster RCNN.

5.1 Faster RCNN

In order to improve the detection results of Faster RCNN, we made some adjustments to its parameters. The default anchors are {128, 256, 512} pixels , and the default aspect ratios are {0.5, 1, 2}, which cater for larger objects. As in our dataset, the resolution of the targets is between 20×20 and 40×40 pixels, we modified the anchor size to {32}. We also changed the number of classes to 2 (livestock and background). After training and testing, the precision–recall curve is shown in Fig. 8. Figure 7(a) shows an example of an image processed by Faster RCNN.

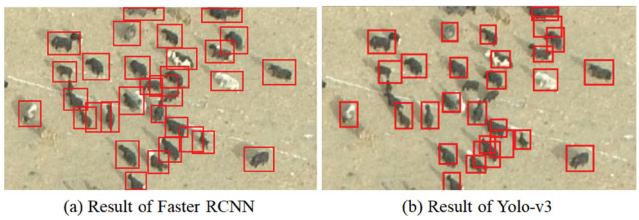


Fig. 7 Example results from Fast RCNN and Yolo-v3.

5.2 Yolo-v3

We also tuned the parameters of the Yolo-v3 network before training. Anchors are again used. We use the k-means [24] algorithm to obtain 9 anchors for our dataset, and the output class is modified to only be livestock. After training and testing, the $P-R$ curve is shown in Fig. 8. Figure 7(b) shows an example of an image processed by Yolo-v3.

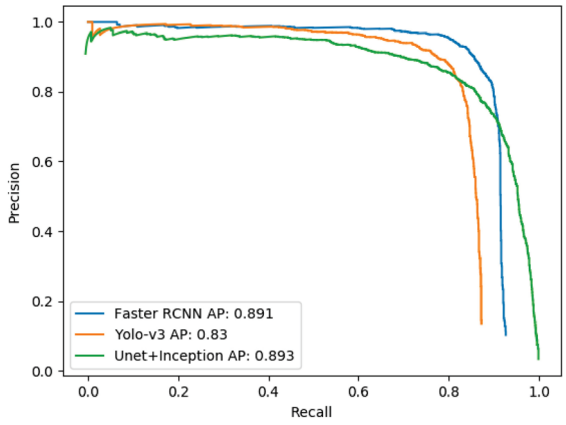


Fig. 8 Performance comparison for Faster RCNN, Yolo-v3, and our method.

6 Conclusions

This work has presented a dataset of aerial images of livestock, and used a method based on U-net and Inception to obtain better detection results than Yolo-v3 and Faster RCNN. However, it still has some shortcomings. Light-colored (white and gray) instances are not well segmented as there are few in the dataset used for training. As a result our method has lower precision than Yolo-v3 and Faster RCNN. The two neural networks in the algorithm need to be trained separately, and new datasets need to be generated for the two networks from the original dataset.

Our aerial livestock datasets are publicly available, and can be downloaded from `https://github.com/hanl2010/Aerial-livestock-dataset/releases`.

Acknowledgements

This work was supported by the Scientific and Technological Achievements Transformation Project of Qinghai, China (Project No. 2018-SF-110), and the National Natural Science Foundation of China (Projects Nos. 61866031 and 61862053).

References

[1] Everingham, M.; van Gool, L.; Williams, C. K. I.; Winn, J.; Zisserman, A. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision* Vol. 88, No. 2, 303–338, 2010.

[2] Lin, T. Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C. L. Microsoft COCO: Common objects in context. In: *Computer Vision—ECCV 2014. Lecture Notes in Computer Science, Vol. 8693*. Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars, T. Eds. Springer Cham, 740–755, 2014.

[3] Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 248–255, 2009.

[4] Russell, B.; Torralba, A.; Murphy, K.; Freeman, W. LabeMe: A database and web-based tool for image annotation. *International Journal of Computer Vision* Vol. 77, 157–173, 2008.

[5] Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In: Proceedings of the International Conference on Neural Information Processing Systems, 91–99, 2015.

[6] Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 779–788, 2016.

[7] Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A. C. SSD: Single shot multibox detector. In: *Computer Vision—ECCV 2016. Lecture Notes in Computer Science, Vol. 9905*. Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 21–37, 2016.

[8] He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, 2961–2969, 2017.

[9] Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional networks for biomedical image segmentation. In: *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015. Lecture Notes in Computer Science, Vol. 9351*. Navab, N.; Hornegger, J.; Wells, W.; Frangi, A. Eds. Springer Cham, 234–241, 2015.

[10] Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. A. Inception-v4, Inception-ResNet and the impact of residual connections on learning. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 4278–4284, 2016.

[11] Xia, G.-S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A large-scale dataset for object detection in aerial images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3974–3983, 2018.

[12] Cheng, G.; Zhou, P.; Han, J. Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing* Vol. 54, No. 12, 7405–7415, 2016.

[13] Zhu, H.; Chen, X.; Dai, W.; Fu, K.; Ye, Q.; Jiao, J. Orientation robust object detection in aerial images using deep convolutional neural network. In: Proceedings of the IEEE International Conference on Image Processing, 3735–3739, 2015.

[14] Heitz, G.; Koller, D. Learning spatial context: Using stuff to find things. In: *Computer Vision—ECCV 2008. Lecture Notes in Computer Science, Vol. 5302*. Forsyth, D.; Torr, P.; Zisserman, A. Eds. Springer Berlin Heidelberg, 30–43, 2008.

[15] Razakarivony, S.; Jurie, F. Vehicle detection in aerial imagery: A small target detection benchmark. *Journal of Visual Communication and Image Representation* Vol. 34, 187–203, 2016.

[16] Mundhenk, T. N.; Konjevod, G.; Sakla, W. A.; Boakye, K. A large contextual dataset for classification, detection and counting of cars with deep learning. In: *Computer Vision—ECCV 2016. Lecture Notes in Computer Science, Vol. 9907*. Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 785–800, 2016.

[17] Liu, K.; Mattyus, G. Fast multiclass vehicle detection on aerial images.*IEEE Geoscience and Remote Sensing Letters* Vol. 12, No. 9, 1938–1942, 2015.

[18] Liu, Z.; Wang, H.; Weng, L.; Yang, Y. Ship rotated bounding box space for ship extraction from high-resolution optical satellite images with complex backgrounds. *IEEE Geoscience and Remote Sensing Letters* Vol. 13, No. 8, 1074–1078, 2017.

[19] Cheng, M.-M.; Zhang, F.-L.; Mitra, N. J.; Huang, X.; Hu, S.-M. RepFinder: Finding approximately repeated scene elements for image editing. *ACM Transactions on Graphics* Vol. 29, No. 4, Article No. 83, 2010.

[20] Krizhevsky, A.; Sutskever, I.; Hinton, G. E. ImageNet classification with deep convolutional neural networks. In: Proceedings of the International Conference on Neural Information Processing Systems, 1097–1105, 2012.

[21] Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 580–587, 2014.

[22] Girshick, R. Fast R-CNN In: Proceedings of the IEEE International Conference on Computer Vision, 1440–1448, 2015.

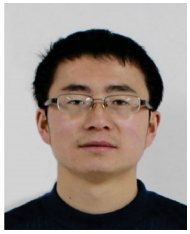
[23] Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 7263–7271, 2017.

[24] Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. *arXiv preprint* arXiv:1804.02767, 2018.

[25] Zhang, F.-L.; Xian, W.; Li, R.-L.; Zheng, Z.-H.; Wang, J.; Hu, S.-M. Detecting and removing visual distractors for video aesthetic enhancement. *IEEE Transactions on Multimedia* Vol. 20, No. 8, 1987–1999, 2018.

[26] Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3431–3440, 2015.

[27] Sakla, W.; Konjevod, G.; Mundhenk, T. N. Deep multi-modal vehicle detection in aerial ISR imagery. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision, 916–923, 2017.



Liang Han is a lecturer in the Department of Computer Technology and Application, Qinghai University. He received his bachelor and master degrees from Lanzhou University in 2010 and 2012 respectively. His research interests include computer vision and machine learning.

Pin Tao is an associate professor in the Computer Science and Technology Department of Tsinghua University. He received his B.S. degree from the Computer Science and Technology Department of Tsinghua University in 1997. In 1999 and 2002, he received his M.S. and Ph.D. degrees in computer applications from Tsinghua University. His research interests are in embedded media processing.

Ralph R. Martin is an emeritus professor of Cardiff University. He has served on the editorial boards of various journals including *Computer-Aided Design*, *Computer Aided Geometric Design*, and *Geometric Models*. In 2014, he was awarded the Friendship Award, China's highest award for foreign nationals. In 2016, he was awarded the title of Solid Modeling Pioneer by the Solid Modeling Association.