VEHICLE DETECTION IN REMOTE SENSING IMAGES USING DEEP NEURAL NETWORKS AND MULTI-TASK LEARNING

Min Cao[1], Hong Ji[2], Zhi Gao[2]*, Tincan Mei[1]

[1] School of Electronic Information, Wuhan University, 430072 Wuhan, China - (2015301220053@whu.edu.cn,mtcwlb@aliyun.com)
[2] School of Remote Sensing and Information Engineering, Wuhan University, 430079 Wuhan, China - (2013301220036@whu.edu.cn, gaozhinus@gmail.com)

Commission II, WG II/III

KEY WORDS: Vehicle detection, Remote sensing images, multi-scale feature fusion, hard example mining, homography augmentation, GAN, super-resolution

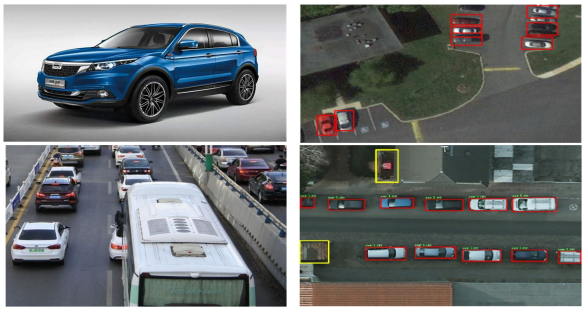
ABSTRACT:

Vehicle detection in remote sensing image has been attracting remarkable attention over past years for its applications in traffic, security, military, and surveillance fields. Due to the stunning success of deep learning techniques in object detection community, we consider to utilize CNNs for vehicle detection task in remote sensing image. Specifically, we take advantage of deep residual network, multi-scale feature fusion, hard example mining and homography augmentation to realize vehicle detection, which almost integrates all the advanced techniques in deep learning community. Furthermore, we simultaneously address super-resolution (SR) and detection problems of low-resolution (LR) image in an end-to-end manner. In consideration of the absence of paired low-/high-resolution data which are generally time-consuming and cumbersome to collect, we leverage generative adversarial network (GAN) for unsupervised SR. Detection loss is back-propagated to SR generator to boost detection performance. We conduct experiments on representative benchmark datasets and demonstrate that our model yields significant improvements over state-of-the-art methods in deep learning and remote sensing areas.

1. INTRODUCTION

Vehicle detection in remote sensing images has been widely applied in many fields and thus received much attention over past years. In spite of the tremendous efforts devoted to this task, the existing methods still require substantial improvement to address several challenges in this area. First, scale and direction variability make it more difficult to accurately locate the vehicle object. Second, complex background increases intraclass variability and interclass similarity. Third, some remote sensing images are captured in low resolution, which would definitely result in lacking sufficient detailed appearance to distinguish vehicle from similar objects. As shown in Figure 1, compared with the everyday images of vehicles, the remote sensing image captured from a perpendicular (or slightly oblique) viewpoint loses the 'face' of vehicle, and the vehicles typically display rectilinear structures. Thus, the presence of nonvehicle rectilinear objects such as trash bins, electrical units, air conditioning units on the tops of buildings, can complicate the task, causing many false alarms. Therefore, researchers are trying to exploit the state-of-the-art deep learning based object detection techniques to push the boundaries of the achievement in this regard.

Object detection can be split into two sub-tasks, localization and classification. Conventional methods addressing this problem are usually via three phases, image segmentation, feature extraction and training classifier. Particularly, saliency detection is utilized to generate region of interest (RoI) as positive samples. Then, low-level, handcrafted visual features (e.g., color histogram, texture, local pattern) are constructed on these samples to train classifiers (e.g., SVM, AdaBoost). However, due to complicate texture architecture and lacking pixel-level annotation, positive training data could be noisy and thus degenerates the subsequent classifier. Furthermore, predefined manual features are usually computationally expensive and can't access to high-level semantic representation of objects, rendering the detection performance has much room for improvement.



(a) (b)

Figure 1. Examples of everyday images of vehicles (a) and remote sensing image of vehicles (b). In remote sensing image, red boxes indicate the correctly detected vehicles and the false alarms are marked with yellow boxes.

Recently, convolutional neural networks (CNNs) exhibits strong feature learning capability and obtains state-of-the-art performance in a variety of classification and recognition tasks on benchmark datasets. Specific to the problem of object detection, great achievements have been made, which are usually driven by the success of region proposal methods and region-based CNNs, such as R-CNN (Girshick et al., 2014), Fast R-CNN (Girshick, 2015). On the basis of above networks, other advanced techno-

* Corresponding author

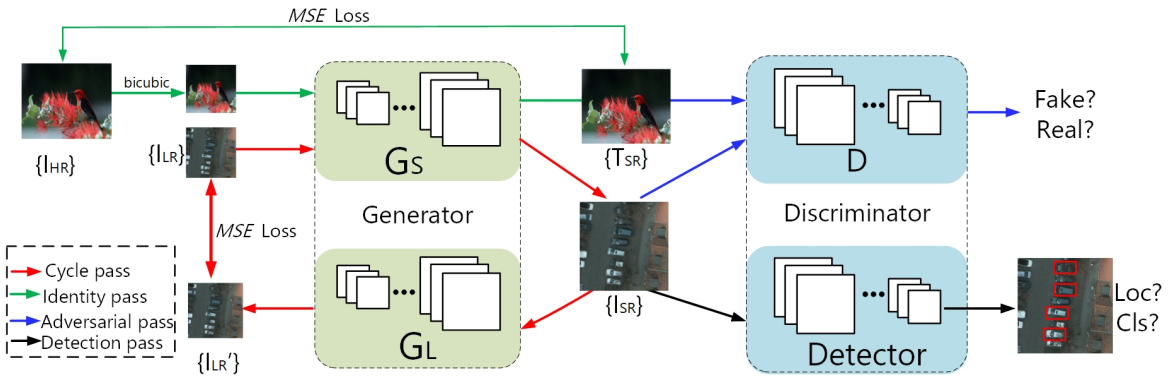


Figure 2. Illustration of our CycleGAN-based vehicle detection module(CVDM). I_{LR} is the input LR image, I_{SR} is the super-resolved image from I_{LR}, I'_{LR} is of LR generated from I_{SR}. T_{HR} is the HR image provided as reference from other high-quality dataset. T_{LR} is down-sampled version of T_{HR}. T_{SR} is the super-resolved HR image from T_{LR}. Colored arrows represent different parts in the whole framework.

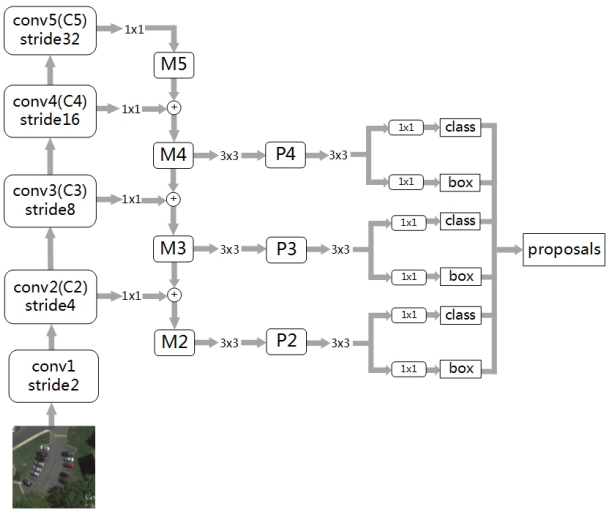


Figure 3. Illustration of our vehicle detection module(VDM). Bottom-up feature extraction and top-down multiscale feature fusion for region proposal generation. M5, M4, M3, M2 indicate the middle levels of intermediate operation. P4, P3, P2 indicate the pyramid levels. 1×1 and 3×3 represent the convolution layer with kernel size 1 and 3 respectively.

logies are employed to boost the detection performance. Feature pyramid network (FPN)(Lin et al., 2017) and Top-down module (TDM) (Shrivastava et al., 2016b) integrate multilayer features to cover objects with different scales. Deep residual networks are used as backbone ConvNet for better representations. Moreover, sample mining technique is applied to dig training data which makes more contributions to the optimization of networks. In remote sensing community, researchers have figured out many CNN-based methods for vehicle detection task. But few works make full use of all above advances in an unified framework, let alone any specific vehicle-oriented design that could be incorporated in CNNs to facilitate detection in remote sensing image.

Based on above observations, we focus on vehicle detection problem in remote sensing images. First, we take advantage of several advanced technologies in DNNs area to bridge the gap between deep learning and remote sensing vehicle detection communities. Particularly, we incorporate deep residual network ResNet50 (He et al., 2016) for feature extraction, multiscale feature architecture to make accurate predictions and hard example mining to facilitate network optimization. Plus, we exploit homograph-based augmentation method to boost overall detection performance. Second, in order to alleviate the problem of low-quality image detection task, which refers to LR image in this work, we leverage on CycleGAN model and multi-task learning, in which SR network is a generator and object detector is treated as a discriminator. Note that, due to lacking paired low-/high-resolution images, we investigate unsupervised learning regime for SR task. Our proposed framework is evaluated on several representative datasets and the results demonstrate that ours outperform state-of-the-art object detection approach Faster R-CNN++ in deep learning community, and other CNN-based methods in remote sensing area.

2. RELATED WORK

In this section, we are going to introduce several representative works in object detection and vehicle detection fields.

2.1 General object detection

Early works in object detection community mainly rely on hand-crafted features (Dalal, Triggs, 2005),(Lowe et al., 1999) and then train classifiers (Felzenszwalb et al., 2009). Since AlexNet (Krizhevsky et al., 2012) got champion in ILSVRC-2012 competition (Deng et al., 2012), plenty of neural network architectures have been proposed and showed powerful learning capability in image classification task (Szegedy et al., 2015),(Szegedy et al., 2016),(He et al., 2016). On the basis of these works, R-CNN (Girshick et al., 2014) takes wrapped potential regions that are provided by region proposal methods (Uijlings et al., 2013) as input and extracts CNNs features, which are utilized to train class-specific liner SVMs. In order to avoid redundant computational cost in R-CNN, Fast R-CNN (Girshick, 2015) forwards entire image through network only once, imposing those seriously overlapped proposals to share computation. Plus, CNNs itself takes responsibility of classification and location regression. Thus the whole detection framework is modeled in an end-to-end manner. Region proposal network (RPN) is proposed to replace region proposal methods in Faster R-CNN (Ren et al., 2017), which significantly accelerates image processing. Later, FPN (Lin et al., 2017), Mask R-CNN (He et al., 2017) follow the fashion of Faster R-CNN pipeline, with improvements in multi-scale training, feature fusion, multi-task

Methods	Feature	Size	Ratio	Number
Original RPN*	C5	128, 256, 512	0.5, 1, 2	9
RPN with FPN*	P2	32	0.5, 1, 2	3
	P3	64	0.5, 1, 2	3
	P4	128	0.5, 1, 2	3
	P5	256	0.5, 1, 2	3
	P6	512	0.5, 1, 2	3
Our RPN	P2	32, 48	0.5, 1, 2	6
	P3	64, 96	0.5, 1, 2	6
	P4	128, 192	0.5, 1, 2	6

* Original RPN and RPN with FPN represent the RPNs of Faster R-CNN and Faster R-CNN++ respectively.

Table 1. Anchor information of our RPN and the original RPN at each possible location

(a) (b)

Figure 4. Instances of both positive (a) and negative (b) hard example patches in our work.

learning. Aforementioned CNN-based detection methods are two-stage, where class-agnostic proposals are provided and then refined in bounding box coordinates and classified into specific classes. Another typical solution for object detection is one-stage method in which proposals are predicted only once. YOLO (Redmon, Farhadi, 2018) and SSD (Liu et al., 2016a) are representatives of such trend.

2.2 Vehicle detection in remote sensing image

Traditional methods addressing vehicle detection problem rely on shallow-learning features. Here we discuss some representatives of them. Early work (Zhao, Nevatia, 2003) chooses the boundary of the car body, the shadow and the boundary of the front windshield as the characteristics to consider the change of view and shadow. The framework in (Liu et al., 2016b) applies Gauss process (GP) classification and gradient based segmentation algorithm (GSEG) to realize vehicle probability estimation of each pixel. Histogram of directional gradient feature descriptor (HOG) (Dalal, Triggs, 2005) and linear support vector machine (SVM) are used in (Bougharriou et al., 2017),(Madhogaria et al., 2015). Work (Kembhavi et al., 2010) uses color probability maps, pixel pairs and HOG to depict the color and geometric structure properties. In (Elmikaty, Stathaki, 2014), gradients map is computed to filter out non-vehicle regions. Multiple descriptors, Histogram of Oriented Gradients (HOG), Fourier and truncated Pyramid Colour Self-Similarity (tPCSS) of selected regions are combined to train a SVM.

Recently, CNNs become the hottest fashion for vehicle detection in remote sensing field. CNN-based detection model combining two independent convolutional neural networks was proposed in work (Zhong et al., 2017). In (Uus, Krilavičius, 2019), a unified framework is proposed on the basis of YOLO to realize airplane detection in aerial images. Similarly, YOLO-like architecture is used to detect aerial vehicles (Carlet, Abayowa, 2017),(Lu et al., 2018). Moreover, many region-based methods are conducted to detect smaller aerial vehicles. In (Kyrkou et al., 2018), sliding-window incorporated with prior knowledge is utilized to generate vehicle-like proposals. Subsequently, CNNs is employed to complete classification task. Work (Ji et al., 2019) investigates improved Faster R-CNN framework for vehicle detection. For efficiency, in (Chen et al., 2013) parallel CNN architecture is applied to extract features of ROIs and produce detection results. Variable sizes of convolutional filter and max-pooling field are adopted to extract variable-scale features for vehicle detection in (Chen et al., 2014).

3. THE PROPOSED SYSTEM

In this section, we elaborate the details of our proposed framework. The whole system consists of two modules, vehicle detection module (VDM) and CycleGAN-based vehicle detection module (CVDM). VDM follows region-based detection pipeline, whose architecture is shown in Figure 3. As shown in Figure 2, CVDM incorporates VDM into an CycleGAN-like architecture, which aims to address detection problem in LR image. Architecture of $Detector$ is as VDM.

3.1 Vehicle detection module (VDM)

We model vehicle detection problem by region-based method, which forwards the entire image through a sequence of convolutional layers, extracts a set of feature maps corresponding to potential region proposals, and then produces detection results via two sibling branches. To train appropriate networks that complete these sub-tasks in an end-to-end fashion, our approach is composed of three main components. First, basic convolutional network generates feature maps. Second, hierarchical architecture constructs multilevel representations and predictions. Third, online hard example mining technique digs discriminative samples.

3.1.1 ConvNet and multilevel feature architecture Deep residual networks have proved to be effective for feature learning and achieved remarkable success in object detection task. Thus, we utilize ResNet50 for feature extraction. However, we observe that for remote sensing image, vehicle object may occupy relatively small area in the whole image. The stride of ResNet50 is 32, which results in losing vehicle information in the process of convolution and pooling operations. Commonly used countermeasure for this case is mutiscale training and testing, which is obviously time consuming and cannot guarantee the features are interpretable enough for final detection. To alleviate this problem, many feature fusion strategies are proposed to build hierarchical architecture and make predictions in multiple feature levels. As FPN obtains state-of-the-art results on canonical benchmark datasets, we adopt FPN-like architecture to construct appropriate multilevel features for our task.

ResNet50 has 5 blocks (each block consists of several convolutional layers, namely $c1$, $c2$, $c3$, $c4$, $c5$). For building semantic representation, feature maps of each block are filtered, upsampled and merged with previous block. Finally, there are 5 stages in proposed multilevel neural network, namely $p2$, $p3$, $p4$, $p5$. Original RPN generates region proposal on the last block of convolutional layers. FPN performs region proposal operation on each stage as well as another additional stage $p6$ at last for covering objects from 32^2 to 512^2. Taking into account of the size range of vehicle objects, we only utilize $p2$,$p3$,$p4$ for region proposal. Table 1 gives detail comparisons between naive RPN, FPN and proposed multilevel architecture. At the phase of subsequent detection network, original RPN projects RoIs on $c5$, while FPN on each stage based on areas of the RoIs.

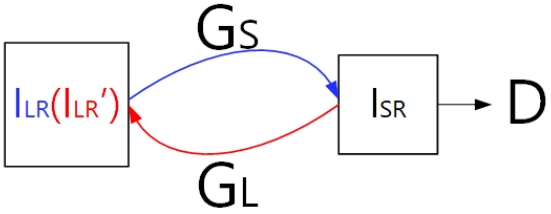


Figure 5. Pipeline of the generator of our CycleGAN network.

Due to vehicle objects generally with small size, we extract feature maps of proposal on the finest stage $p2$, which provides discriminative representations for classification and location.

3.1.2 Sub-detection network Feature maps of each proposal are pooled to 7×7 bins, followed by two continuous fully connected layers. Then the outputs are forwarded to sibling branches for classification and localization. For classification task, we apply standard multi-class cross entropy loss which can be formulated as Equation 1:

$$
\begin{aligned}
L(\{p_i\},\{t_i\}) &= \frac{1}{N_{cls}}\sum_i L_{cls}(p_i,l_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(\mathbf{t_i},\mathbf{t_i^*}) \\
L_{cls}(p_i,l_i^*) &= -log(p_{l_i^*}), \\
L_{reg}(\mathbf{t_i},\mathbf{t_i^*}) &= smooth_{L1}(\mathbf{t_i}-\mathbf{t_i^*}), \\
smooth_{L1}(\boldsymbol{x}) &= \begin{cases} 0.5\boldsymbol{x}^2 & \text{if } |\boldsymbol{x}|<1, \\ |\boldsymbol{x}|-0.5 & \text{otherwise.} \end{cases}
\end{aligned}
\tag{1}
$$

Here, i is the index of a region in a mini-batch and p_i is the predicted probability of region i being a vehicle. The ground-truth label l_i^* is 1 if the region is positive, and is 0 otherwise. $\mathbf{t_i}$ is a vector representing the 4 parameterized coordinates of the predicted bounding box, and $\mathbf{t_i^*}$ is that of the ground-truth box associated with a positive anchor. For more detailed discussion of this objective function and the recommended value of parameters, readers can refer to (Ren et al., 2017).

3.1.3 Online hard example mining Data mining aims to dig out samples that are not distinguishable enough for training and thus make the classifier more discriminative progressively. Especially in remote sensing area, background information is usually complex, implying high similarity between positive and negative samples. Randomly selecting training samples would miss useful information. Consequently, proceeding samples from simple to complex is proposed for solving this problem. Specifically, researchers make use of alternative learning strategy, incorporating influential samples gradually, totally training the classifier for several rounds. Selection criteria depends on confidence of previous detection model. We call this mining approach as offline manner. Later, in (Shrivastava et al., 2016a), researchers consider to complete this task with online manner and successfully embed the algorithm into Faster R-CNN, namely online hard example mining. In this manner, during each training forward pass, those proposals with high loss value are selected as hard examples to back-propagation for estimating weights. In implementation, backbone ConvNet is followed by two sub-detection networks, called readonly and standard modules respectively. The former branch is responsible for calculating loss value of proposals and the latter branch accounts for standard SGD operation combined with basic ConvNet. Readers can access detailed information of this algorithm in (Shrivastava et al., 2016a). We display some hard examples obtained of our implementation in Figure. 4.

3.2 CycleGAN-based vehicle detection module (CVDM)

In this section, we focus on detection task in LR image by simultaneous SR operation and object detection. Commonly solution for the problem is directly upsampling image by bicubic kernel, which definitely loses appearance details. Thus, we exploit SR method to enhance LR image. Existing methods model this problem with fully convolutional network (FCN) and pixel-level annotation, paired low-/high-resolution images, are essential for these models. However, in practise, it's difficult to obtain paired training data. To ease the burden of data collection, unsupervised learning regime is developed for domain translation. Our approach is inspired by two representatives, CycleGAN (Zhu et al., 2017) and Cycle-in-Cycle (Yuan et al., 2018), which realize unsupervised image translation by GAN. Our framework consists of generator and discriminator, in which G_S super-resolves LR image, G_L restores obtained SR image back to LR domain, $Detector$ realizes vehicle detection.

3.2.1 CycleGAN-based image super-resolution As shown in Figure 5, there are two generators for image SR component, where I_{LR}, I'_{LR} represent original LR image and its restored counterpart respectively, I_{SR} is corresponding super-resolved one. Cycle consistency loss is:

$$
\mathcal{L}_{cyc} = \mathbb{E}_{I_{LR}\sim P_{data}(I_{LR})}[||G_L(G_S(I_{LR})) - I_{LR}||_2] \tag{2}
$$

Where $||.||$ means MES loss. $I'_{LR} = G_S(I_{LR})$.

In order to preserve the color and quality of super-resolved image, we add identity loss to train the whole model. Its formulation can be seen in Equation 3, which also uses MSE loss. As we can't access to paired images in our target remote sensing data, here we refer to dataset that is for SR purpose and not related to our target data. T_{HR} means high-resolution reference, while T_{LR} represents its LR counterpart, which is downsampled by bicubic kernel.

$$
\mathcal{L}_{Idt} = \mathbb{E}_{T_{LR}\sim P_{data}(T_{LR})}[||G_S(T_{LR}) - T_{HR}||_2] \tag{3}
$$

We utilize adversarial loss for G_S and its discriminator D, which aims to distinguish high-resolution image T_{HR} from generated one I_{SR}. We present the objective as:

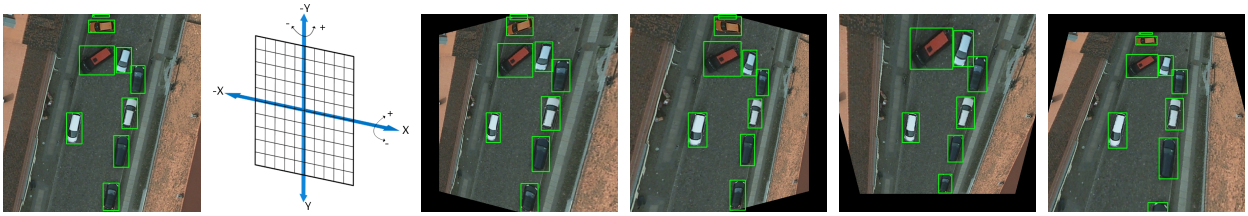
$$
\begin{aligned}
\mathcal{L}_{GAN} &= \mathbb{E}_{T_{HR}\sim P_{data}(T_{HR})}[\log(D(T_{HR}))] \\
&+\mathbb{E}_{I_{LR}\sim P_{data}(I_{LR})}[\log(1-D(G(I_{LR})))]
\end{aligned}
\tag{4}
$$

Now, the objective for SR module is:

$$
\mathcal{L}_{cycGAN} = \mathcal{L}_{GAN} + \lambda_1\mathcal{L}_{cyc} + \lambda_2\mathcal{L}_{Idt} \tag{5}
$$

Where λ_1 and λ_2 control the importance of consistency loss and identity loss in the whole model. The architectures of above two generators are shown in Table 2 and Table 3. Discriminator D is displayed in Table 4.

3.2.2 Discriminator network $Detector$ We embed proposed VDM as a discriminator in our CycleGAN-based framework, which takes generated I_{SR} as input and outputs detection result of vehicle object. So our CVDM is modeled in multi-task learning fashion, including super-resolution and object detection. Here, taking into account of the relationship between the



(a) (b) (c) (d) (e) (f)

Figure 6. Examples of homography data augmentation. (**a**) Original image cropped from available dataset. (**b**) Schematic illustration of rotation along x or y axis. (**c**)(**d**) Results of rotation along y axis with -15° and 15° respectively. (**e**)(**f**) Results of rotation along x axis with -15° and 15° respectively.

layer	conv	residual block $\times 16$	conv	element-wise sum	conv	pixelshuffle	conv	pixelshuffle	conv
kernel size	3	3	3	-	3	-	3	-	3
kernel num	64	64	64	-	256	-	256	-	64
stride	1	1	1	-	1	$\frac{1}{2}$	1	$\frac{1}{2}$	1

Table 2. Architecture of upsampling Generator G_S.

layer	conv	conv$\times 2$	residual block $\times 6$	conv$\times 2$	conv
kernel size	7	4	3	3	7
kernel num	64	64	64	64	3
stride	1	2	1	1	1

Table 3. Architecture of downsampling Generator G_L.

layer	conv	conv	BN	conv	BN	conv	BN	conv
kernel size	4	4	-	4	-	4	-	4
kernel num	64	128	-	256	-	512	-	1
stride	2	2	-	2	-	1	-	1

Table 4. Architecture of Discriminator D.

two tasks, we back-propagate detection loss to SR network, which guides the generator to produce image that is beneficial for detection purpose. In summary, the overall objective for CVDM is:

$$\mathcal{L} = \mathcal{L}_{GAN} + \lambda_1 \mathcal{L}_{cyc} + \lambda_2 \mathcal{L}_{Idt} + \lambda_3 \mathcal{L}_{Det} \tag{6}$$

where L_{Det} is detection loss, the same as Equation 1. Its loss weight is λ_3.

3.3 Implementation details

We first train VDM network, whose backbone is initialized by ResNet50 trained on ImageNet classification task. The model is trained by SGD optimizer and totally trained for 60k iterations. Initial learning rate is set to 2.5e-3 and reduced to 2.5e-4 after 40k iterations. Next, we train CycleGAN-based SR model, namely **CycGANSR**. G_S is initialized by the model released from (Lim et al., 2017). G_L and D are trained from scratch. All the networks are trained with Adam optimizer apart from $Detector$. Moreover, their initial learning rate is set to 1e-4 and reduced to 1e-5 after 40k iterations. The batchsize is 2 and the networks are totally trained for 80k iterations. As it's difficult to optimize generator and discriminator simultaneously, we leverage on alternative learning strategy. When training generators, the parameters of discriminator are fixed and objective function is shown as Eq (7), just without the classification loss (4th term) and localization loss (5th term). Here λ_1 and λ_2 are both set to 1. For training discriminator, we fix the generators and the objective function is shown as Eq (8), but without detector loss.

$$\begin{aligned}
\arg\min_{G^*} \frac{1}{N}\sum_i ||D(G_{UP}(I^i_{LR})) - 1||_2 + \\
\frac{1}{N}\sum_i \lambda_1 ||G_L(G_S(I^i_{LR})) - I^i_{LR}||_2 + \\
\frac{1}{N}\sum_i \lambda_2 ||G_L(G_S(T^i_{LR})) - T^i_{HR}||_2 + \\
\frac{1}{N}\sum_i -\lambda_3 \log(Det_{cls}(G_S(I^i_{LR}))) + \\
\frac{1}{N}\sum_i \lambda_3 [u^i \geq 1](Det_{reg}(G_S(I^i_{LR}), \mathbf{t}^{\mathbf{i}}_*)
\end{aligned} \tag{7}$$

$$\begin{aligned}
\arg\min_{D^*} \frac{1}{N}\sum_i (||D(G_S(I^i_{LR}))||_2 + ||D(T^i_{HR}) - 1||_2) + \\
\frac{1}{N}\sum_i -\omega \log(Det(G_S(I^i_{LR}))) + \\
\frac{1}{N}\sum_i \omega [u^i \geq 1](Det_{reg}(G_S(I^i_{LR}), \mathbf{t}^{\mathbf{i}}_*)
\end{aligned} \tag{8}$$

After training CycGANSR network and detection network, we train them jointly. Its training procedure is the same as CycGANSR and its objective functions are as Eq (7) and Eq (8) respectively. λ_3, ω are set to 0.01 and 0.1 respectively. For VDM, the scale of images for training is 800×800. For CVDM, input image is 200×200. Upsampling factor is 4.

4. EXPERIMENTS

In this section, we first introduce experiment setup, including data preparation and augmentation. Then we present results and compare ours with other state-of-the-art methods.

4.1 Experiment setup

4.1.1 Datasets and metrics We conduct elaborate experiments on four datasets. 1) **Potsdam** (Rottensteiner et al., 2012) dataset consists of 38 ortho-rectified aerial IR-RGB images, 24 of which are labeled for semantic segmentation, including vehicle category. Its ground sampling distance (GSD) is 5cm. 2) **VEDAI** dataset (Razakarivony, Jurie, 2016) is from the Utah

Methods	Input info*	AP	AP@0.5	AP@0.75	mRecall
YOLOv3	600*600	0.309	0.696	0.189	0.382
YOLOv3	800*800	0.259	0.624	0.141	0.330
FASR	800*800	0.409	0.764	0.372	0.529
FASR	800*800, HF	0.424	0.823	0.369	0.541
FASR	1200*1200	0.241	0.627	0.100	0.349
FASR	1200*1200, HF	0.247	0.669	0.144	0.361
RVD	-	-	0.502	-	-
FVD	-	-	0.66	-	-
YVD	-	-	0.767	-	-
DVD	-	-	0.817	-	-
VDM	800*800	0.438	0.794	0.403	0.529
VDM	800*800, HF	0.449	**0.856**	0.402	0.548
VDM	1200*1200	0.449	0.848	0.427	0.523
VDM	1200*1200, HF	**0.458**	0.835	**0.457**	**0.573**

* This column indicates the resolution of input testing image. HF means the input includes its horizontal flipping. FASR represents Faster R-CNN++. RVD, FVD, YVD, DVD represent works (Zhong et al., 2017), (Carlet, Abayowa, 2017), (Lu et al., 2018), (Uus, Krilavičius, 2019) respectively

Table 5. Results on VEDAI dataset

AGRC image collection, with 12.5cm GSD. We choose its half-resolution version of 512×512. Thus, the vehicle in this set is smaller than other datasets, and the car is typically about 10×8 pixels. 3) **DLR Munich** dataset (Liu, Mattyus, 2015) is captured at about 1000m above the ground over the area of Munich, Germany, using DLR 3K camera system. There are totally 20 images (of resolution 5616×3744 pixels), with approximate 13cm GSD. 4) **UCAS-AOD** dataset (Zhu et al., 2015) consists of 510 satellite images with resolution 659×1280 pixels, including 410 training images and 100 testing ones. Due to influence of environment and equipment, the sizes of vehicles in this dataset are usually larger than that of the Munich dataset. However, the quality of this dataset is much poorer.

We apply average precision (**AP**) and mean recall rate (**mRecall**, which is mean value of the recalls from IoU 0.5 to 0.95, with 0.05 stride) for comparing ours with other methods in the deep learning community. Furthermore, **F1** score, **precision** and **recall** (with IoU 0.5) are adopted for comparison with method in remote sensing area. Their definitions are:

$$recall = \frac{TP}{TP + FN} \tag{9}$$

$$precision = \frac{TP}{TP + FP} \tag{10}$$

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{11}$$

where TP, FP,FN represent true positive, false positive and false negative respectively.

4.1.2 Homography-based data augmentation We define the ground as a plane, which is usually not perpendicular to the main optical axis of camera. Thus, it makes deformation of vehicle targets and difficulty for detection task. To alleviate this problem, we apply homography transformation on training data to simulate the data captured from a more oblique viewpoint. We display some examples in Figure 6. Each image is rotated along x, y axis, with rotation angle $-15°$ and $15°$ respectively. Given the rotation angle, the homography matrix can be estimated to calculate the coordinates of transformed bounding box. As shown in Figure 6(c)(d)(e)(f), we obtain the transformed vehicles with remarkable appearance variances, together with their bounding boxes.

4.2 Results and comparisons

4.2.1 Results of VDM In computer vision field, we compare our results with Faster R-CNN++ (we improve naive Faster R-CNN by replacing VGG16 ConvNet with ResNet50 and utilizing FPN architecture for feature fusion) and YOLOv3, which are representatives in two-stage and one-stage trends respectively. Notice that we use advanced version of Faster R-CNN for fair comparison. In remote sensing community, we directly report the results that on such datasets. We clarify that YOLOv3 is trained with multi scales (including 600 × 600), while other results are trained with 800 × 800.

Methods	Input info*	AP	AP@0.5	AP@0.75	mRecall
YOLOv3	600*600	0.624	0.903	0.779	0.671
YOLOv3	800*800	0.627	**0.904**	0.758	0.682
FASR	800*800	0.630	0.888	0.764	0.721
FASR	800*800, HF	0.634	0.884	0.756	0.725
FASR	1200*1200	0.541	0.791	0.651	0.642
FASR	1200*1200, HF	0.547	0.862	0.640	0.660
VDM	800*800	0.662	0.902	**0.793**	0.735
VDM	800*800, HF	**0.668**	0.902	0.791	**0.740**
VDM	1200*1200	0.657	0.897	0.782	0.724
VDM	1200*1200, HF	0.655	0.897	0.781	0.731

Table 6. Results on Potsdam dataset

Figure 7. Examples of the remote sensing images from VEDAI Munich (1st row), Potsdam (2nd row), DLR (3rd row), UCAS-AOD (4th row) datasets respectively. The vehicle detection results of our method are marked with green boxes. Blue and red boxes indicate missing and false alarm respectively.

In Table 5 and 1st row of Figure 7, we report and display the results on **VEDAI** dataset. We first discuss results inferred on training scale level 800×800, where ours outperforms YOLOv3 and Faster R-CNN++ by 13 and 3 points in AP, respectively. On higher IoU threshold 0.75, result of YOLOv3 reduces sharply to 18.9%, in comparison with proposed VDM 40.3% and Faster R-CNN++ 37.2%, which implies region-based methods are more robust to small vehicles. mRecall value also verifies this conclusion. As augmented testing can boost overall performance, we apply horizontal flip augmentation on testing data. It can be seen that results of VDM and Faster R-CNN++ are both im-

Methods	Input info	Recall	Precision	F1-score
(Audebert et al., 2017)	800*800	0.907	0.841	0.870
VDM	800*800	0.918	**0.935**	**0.926**
VDM	800*800, HF	**0.926**	0.911	0.918

Table 7. Results on Potsdam vehicle dataset(IoU=0.5), compared with method in remote sensing.

Methods	AP	AP@0.5	AP@0.75	mRecall
R-FCN	0.321	0.613	0.303	0.396
SSD	0.249	0.521	0.212	0.258
YOLOv3	0.262	0.574	0.186	0.273
FASR	0.342	0.691	0.292	0.362
FASR+Bicubic	0.487	0.795	0.571	0.554
FASR+EDSR	0.450	0.784	0.530	0.538
FASR+CycGANSR	0.541	0.801	0.658	0.628
CVDM	**0.599**	**0.889**	**0.684**	**0.648**

Table 8. Results on Munich DLR dataset

proved slightly. To demonstrate the effectiveness of our VDM, we test their ability on other scales. For YOLOv3 and Faster R-CNN++, their results are much poorer than that on original scale. Especially the latter dropped more than 15% points for all metrics when testing on 1200×1200 scale. Reversely, our results are much better than before. mRecall rate of the last row exceeds that of Faster R-CNN++ by more than 20 points, which fully explains the robustness and generality of our VDM.

In Table 6, 7 and 2nd row of Figure 7, we report and display the results on **Potsdam** dataset. As the GSD of **Potsdam** is the smallest, the objects have the best appearance quality compared to other datasets, and all methods report better results. However, apart from a small drop at AP with 0.5 IoU threshold, our method achieves best results on other three metrics. For augment testing, proposed VDM is still robust, when Faster R-CNN++ behaves badly.

4.2.2 Results of CVDM We conduct experiments on **Munich DLR** and **UCAS-AOD** datasets. To well illustrate proposed CVDM, we downsample the training image to 200×200 as input of our model. Here, we also compare ours with R-FCN (Dai et al., 2016) and SSD (Liu et al., 2016a), which are both competitive methods in one-stage filed. Table 8 and 9 show results on Munich DLR and UCAS-AOD datasets. First, we test the detection performance of R-FCN, SSD, YOLOv3, and Faster R-CNN++ on the input LR images (without any SR operation), and the results are poor, which demonstrates that low-quality image limits detection performance, both on one-stage and two-stage methods. Next, we study the influence of different upsampling methods, bicubic interpolation, EDSR (pretrained model), CycleGAN-based SR (which does not incorporate detector) and our CVDM. The results of upsampled image are all better than that of LR image. It's clear that our method achieves the best results on all metrics, outperforming the second best result with about 5%. Although EDSR method uses neural network for SR, it obtains similar results with bicubic interpolation method because it is trained on dataset that is quite different with our target data. Some examples of the detection results on these two datasets are shown in row 3 and 4 of Figure 7.

Methods	AP	AP@0.5	AP@0.75	mRecall
R-FCN	0.316	0.605	0.297	0.391
SSD	0.264	0.566	0.188	0.286
YOLOv3	0.281	0.593	0.196	0.311
FASR	0.337	0.682	0.288	0.362
FASR+Bicubic	0.481	0.805	0.526	0.569
FASR+EDSR	0.486	0.804	0.526	0.559
FASR+CycGANSR	0.516	0.804	0.611	0.594
CVDM	**0.572**	**0.885**	**0.637**	**0.653**

Table 9. Results on UCAS-AOD dataset

In this work, we implement our experiments on PyTorch and NVIDIA GeForce GTX1080Ti with 12 GB on-board memory.

5. CONCLUSION

In this paper, we have investigated advanced deep learning techniques, which include better backbone ConvNet, multilevel feature fusion and sample mining, to realize vehicle detection in remote sensing image. Homography data augmentation is proposed to address multi-angle problem in data collection stage. Furthermore, we leverage on CycleGAN-like architecture to realize simultaneous SR and object detection for LR image, where SR task relies on unsupervised learning regime and is guided by detection task. Our experiments show that our system surpasses state-of-the-art methods. In future, we plan on realizing instance segmentation of vehicle in remote sensing image.

REFERENCES

Audebert, N., Le Saux, B., Lefèvre, S., 2017. Segment-before-detect: Vehicle detection and classification through semantic segmentation of aerial images. *Remote Sensing*, 9(4), 368.

Bougharriou, S., Hamdaoui, F., Mtibaa, A., 2017. Linear svm classifier based hog car detection. *2017 18th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, IEEE, 241–245.

Carlet, J., Abayowa, B., 2017. Fast vehicle detection in aerial imagery. *arXiv preprint arXiv:1709.08666*.

Chen, X., Xiang, S., Liu, C.-L., Pan, C.-H., 2013. Vehicle detection in satellite images by parallel deep convolutional neural networks. *2013 2nd IAPR Asian Conference on Pattern Recognition*, IEEE, 181–185.

Chen, X., Xiang, S., Liu, C.-L., Pan, C.-H., 2014. Vehicle detection in satellite images by hybrid deep convolutional neural networks. *IEEE Geoscience and remote sensing letters*, 11(10), 1797–1801.

Dai, J., Li, Y., He, K., Sun, J., 2016. R-fcn: Object detection via region-based fully convolutional networks. *Advances in neural information processing systems*, 379–387.

Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, 1, IEEE, 886–893.

Deng, J., Berg, A., Satheesh, S., Su, H., Khosla, A., Fei-Fei, L., 2012. ILSVRC-2012, 2012. *URL http://www. image-net. org/challenges/LSVRC*.

Elmikaty, M., Stathaki, T., 2014. Car detection in high-resolution urban scenes using multiple image descriptors. *2014 22nd International Conference on Pattern Recognition*, IEEE, 4299–4304.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., Ramanan, D., 2009. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9), 1627–1645.

Girshick, R., 2015. Fast r-cnn. *Proceedings of the IEEE international conference on computer vision*, 1440–1448.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 580–587.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask r-cnn. *Proceedings of the IEEE international conference on computer vision*, 2961–2969.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Ji, H., Gao, Z., Mei, T., Li, Y., 2019. Improved Faster R-CNN With Multiscale Feature Fusion and Homography Augmentation for Vehicle Detection in Remote Sensing Images. *IEEE Geoscience and Remote Sensing Letters*, 16(11), 1761–1765.

Kembhavi, A., Harwood, D., Davis, L. S., 2010. Vehicle detection using partial least squares. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(6), 1250–1265.

Krizhevsky, A., Sutskever, I., Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 1097–1105.

Kyrkou, C., Timotheou, S., Kolios, P., Theocharides, T., Panayiotou, C. G., 2018. Optimized vision-directed deployment of uavs for rapid traffic monitoring. *2018 IEEE International Conference on Consumer Electronics (ICCE)*, IEEE, 1–6.

Lim, B., Son, S., Kim, H., Nah, S., Mu Lee, K., 2017. Enhanced deep residual networks for single image super-resolution. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 136–144.

Lin, T.-Y., Dollár, P., Girshick, R. B., He, K., Hariharan, B., Belongie, S. J., 2017. Feature pyramid networks for object detection. *CVPR*, 1number 2, 4.

Liu, K., Mattyus, G., 2015. Fast Multiclass Vehicle Detection on Aerial Images. *IEEE Geosci. Remote Sensing Lett.*, 12(9), 1938–1942.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A. C., 2016a. Ssd: Single shot multibox detector. *European conference on computer vision*, Springer, 21–37.

Liu, Y., Monteiro, S. T., Saber, E., 2016b. Vehicle detection from aerial color imagery and airborne lidar data. *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, IEEE, 1384–1387.

Lowe, D. G. et al., 1999. Object recognition from local scale-invariant features. *iccv*, 99number 2, 1150–1157.

Lu, J., Ma, C., Li, L., Xing, X., Zhang, Y., Wang, Z., Xu, J., 2018. A Vehicle Detection Method for Aerial Image Based on YOLO. *Journal of Computer and Communications*, 6(11), 98–107.

Madhogaria, S., Baggenstoss, P. M., Schikora, M., Koch, W., Cremers, D., 2015. Car detection by fusion of HOG and causal MRF. *IEEE Transactions on Aerospace and Electronic Systems*, 51(1), 575–590.

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

Redmon, J., Farhadi, A., 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Ren, S., He, K., Girshick, R., Sun, J., 2017. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 1137–1149.

Rottensteiner, F., Sohn, G., Jung, J., Gerke, M., Baillard, C., Benitez, S., Breitkopf, U., 2012. The ISPRS benchmark on urban object classification and 3D building reconstruction. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci*, 1(3), 293–298.

Shrivastava, A., Gupta, A., Girshick, R., 2016a. Training region-based object detectors with online hard example mining. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 761–769.

Shrivastava, A., Sukthankar, R., Malik, J., Gupta, A., 2016b. Beyond skip connections: Top-down modulation for object detection. *arXiv preprint arXiv:1612.06851*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2016. Rethinking the inception architecture for computer vision. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

Uijlings, J. R., Van De Sande, K. E., Gevers, T., Smeulders, A. W., 2013. Selective search for object recognition. *International journal of computer vision*, 104(2), 154–171.

Uus, J., Krilavičius, T., 2019. Detection of different types of vehicles from aerial imagery. *CEUR Workshop proceedings [electronic resource]: IVUS 2019, International conference on information technologies, Kaunas, Lithuania, 25 April, 2019. Aachen: CEUR-WS, 2019, Vol. 2470*.

Yuan, Y., Liu, S., Zhang, J., Zhang, Y., Dong, C., Lin, L., 2018. Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 701–710.

Zhao, T., Nevatia, R., 2003. Car detection in low resolution aerial images. *Image and Vision Computing*, 21(8), 693–703.

Zhong, J., Lei, T., Yao, G., 2017. Robust vehicle detection in aerial images based on cascaded convolutional neural networks. *Sensors*, 17(12), 2720.

Zhu, H., Chen, X., Dai, W., Fu, K., Ye, Q., Jiao, J., 2015. Orientation robust object detection in aerial images using deep convolutional neural network. *Image Processing (ICIP), 2015 IEEE International Conference on*, IEEE, 3735–3739.

Zhu, J.-Y., Park, T., Isola, P., Efros, A. A., 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. *Proceedings of the IEEE international conference on computer vision*, 2223–2232.