2017 IEEE Winter Conference on Applications of Computer Vision

Deep Multi-Modal Vehicle Detection in Aerial ISR Imagery

Wesam Sakla Goran Konjevod T. Nathan Mundhenk

Computational Engineering Division
Lawrence Livermore National Laboratory
{sakla1,konjevod1,mundhenk1}@llnl.gov

Abstract

Since the introduction of deep convolutional neural networks (CNNs), object detection in imagery has witnessed substantial breakthroughs in state-of-the-art performance. The defense community utilizes overhead image sensors that acquire large field-of-view aerial imagery in various bands of the electromagnetic spectrum, which is then exploited for various applications, including the detection and localization of man-made objects. In this work, we utilize a recent state-of-the art object detection algorithm, faster R-CNN, to train a deep CNN for vehicle detection in multimodal imagery. We utilize the vehicle detection in aerial imagery (VEDAI) dataset, which contains overhead imagery that is representative of an ISR setting. Our contribution includes modification of key parameters in the faster R-CNN algorithm for this setting where the objects of interest are spatially small, occupying less than 1.5×10^{-3} *of the total image pixels. Our experiments show that (1) an appropriately trained deep CNN leads to average precision rates above* 93% *on vehicle detection, and (2) transfer learning between imagery modalities is possible, yielding average precision rates above* 90% *in the absence of fine-tuning.*

1. Introduction

Automatic Target Recognition (ATR) algorithms are highly sought by the defense community for intelligence, surveillance, and recoinnassance (ISR) applications. With the deployment of an array of high-resolution imaging sensors of various modalities, including visible band electro-optical imagery and infrared imagery, the defense community collects an abundance of data that can be exploited. Automated algorithms are needed that can reliably and quickly ingest imagery for target detection and localization. In many applications, the targets of interest are various types of land-based vehicles such as passenger cars, pickup trucks, and small vans. In recent years, deep convolutional neural networks [11],[10],[21], [7] are increasingly becoming the state-of-the-art in various computer vision applications, including image classification, object detection, and semantic segmentation.

In contrast to high-resolution imagery that is acquired at the ground-level and which represents many of the academic datasets in computer vision such as Pascal VOC [4] and ImageNet [3], aerial overhead imagery such as that in the VEDAI dataset [13] presents unique challenges that degrade object detection performance, including low spatial resolution (i.e., small number of pixels on target), variability in orientation, lighting/shadowing changes, specularities, and occlusion.

In this work, we focus on a specific algorithm, *faster R-CNN* [15], that utilizes deep convolutional networks for performing end-to-end object detection. We specifically address fundamental parameters of the faster R-CNN algorithm that influence the capability to detect small targets in overhead imagery such as that in the VEDAI dataset. We empirically validate that for certain sizes of objects, the default parameters that allow for learning region proposals via faster R-CNN are insufficient and require modification to detect small targets. We also explore the possibilities of *transfer learning* between image modalities. Often times, in ISR settings, image modalities are not collected simultaneously, so there is utility in applying a common object detection model across modalities. The remainder of this paper is organized as follows. In Section 2, we provide context related to object detection algorithms and the use of recent CNN-based methods. We review the faster R-CNN algorithm architecture in Section 3. The VEDAI dataset and annotations are described in Section 4. In Section 5, we discuss the proposed modifications to faster R-CNN for the detection of small vehicles. Experiments and results are detailed in Section 6, with concluding remarks and future work discussed in Section 7.

2. Background

Prior to the introduction of deep CNNs, sliding window detectors (e.g., [19]) were the state-of-the-art approach

978-1-5090-4822-9/17 $31.00 © 2017 IEEE
DOI 10.1109/WACV.2017.107

to object detection. Sliding window methods utilize both a specific hand-crafted feature representation such as histogram of gradients (HOG) (e.g., [2]) and a classifier such as a support vector machine (SVM) to independently binary classify all sub-windows of an image as belonging to an object or background. While these methods modeled the object as being a rigid template, newer methods such as the classic deformable parts-based model [5] modeled objects as combinations of spatially organized parts.

The use of deep CNNs has lead to significant improvements in state-of-the-art performance on the ILSVRC and COCO 2015 competitions, taking first place on the tasks of ImageNet detection/localization and COCO detection [8]. While a handful of CNN-based object detection approaches have been developed over the last few years [16], [14], [12], our work focuses on the use of the faster R-CNN algorithm [15]. In lieu of using sliding windows to find objects in images, the original R-CNN framework, proposed by Girshick *et al.* uses object *proposals* generated externally by an algorithm such as *selective search* [18] or *edge boxes* [22] to fine-tune an ImageNet pre-trained CNN for object detection. Fast R-CNN [6] was developed to greatly reduce the computation of the forward pass for R-CNN proposals by sharing convolutional features and pooling object proposals from the last convolutional layer, however it still relies on the generation of external object proposals.

3. Faster R-CNN

To increase real-time object detection speeds and encapsulate the object detection pipeline in a unified end-to-end framework, the faster R-CNN algorithm eliminates the need for the generation of external object proposals by introducing a *region proposal network* (RPN) that learns region proposals using the CNN features. The learned region proposals are then fed upstream into the fast R-CNN object detection network. Faster R-CNN can detect objects at a frame rate of 5 *fps* on a single GPU using the VGG-16 model [17].

3.1. VGG-16 CNN Trunk

While faster R-CNN can utilize the convolutional layers from any CNN architecture, we have chosen the VGG-16 model [17] which has been pre-trained on ImageNet. VGG-16 has 5 groups of convolutional layers, where the first two groups contain two convolutional layers and the last three groups contain three convolutional layers. Each group, denoted by *conv1_x* - *conv5_x*, is separated by a max pooling layer with 2 × 2 kernels that progressively decreases the spatial resolution of the preceding activation maps by half along each dimension. The number of convolutional filters in each group are 64, 128, 256, 512, and 512, respectively. Figure 1 provides an illustration of the VGG-16 convolutional trunk.

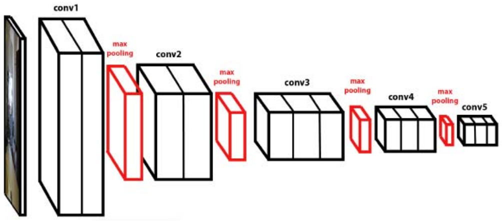

Figure 1. VGG-16 convolutional trunk.

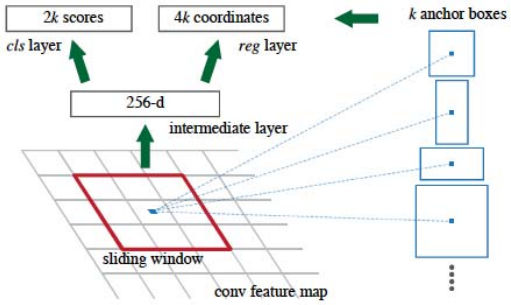

Figure 2. Faster R-CNN region proposal network [15].

3.2. Region Proposal Network

An RPN [15] is a fully convolutional network that simultaneously predicts rectangular object bounding boxes and *objectness* (i.e., membership to a set of object classes vs. background) scores. To generate region proposals, a mini-network is constructed on top of the activation map of the last shared convolutional layer. The input to this mini-network is an $n \times n$ (typically 3×3) spatial window of the input convolutional feature map. The outputs of this convolutional layer's sliding window are then mapped to a low dimensional (e.g., 256, 512) feature vector, followed by a *ReLU* nonlinearity. Finally, these low-dim features are fed into two fully connected layers – a bounding box regression layer (*reg*) and a box-classification layer (*cls*).

A total of k region proposals are predicted at each sliding window location, where k denotes the number of *anchors*. An anchor represents a canonical {scale, aspect ratio} configuration of a region proposal, and thus allows for multi-scale object detection. Hence, the *reg* layer has $4k$ outputs encoding the 4 coordinates of k bounding boxes, and the *cls* layer has $2k$ outputs, with 2 scores for each of the k boxes indicating how likely each of the k regions contains object or background. By default, faster R-CNN uses 3 scales and 3 aspect ratios, yielding $k = 9$ anchors at each sliding window position. The RPN is illustrated in Figure 2.

4. VEDI Dataset and Annotations

We perform all experiments using the VEDAI dataset [13], which contains aerial, ortho-normalized imagery from

Name	Size (pixels)	GSD (cmpp)	Channels
Large color (LCIs)	1024×1024	12.5×12.5	3
Small color (SCIs)	512×512	25.0×25.0	3
Large infrared (LIIs)	1024×1024	12.5×12.5	1
Small infrared (SIIs)	512×512	25.0×25.0	1

Table 1. The four subsets of images in the VEDAI dataset.

the publicly available Utah AGRC [1] database. The original large field-of-view satellite images have been partitioned into images of size 1024×1024 and contain a wide diversity of vehicles, backgrounds, and confuser objects. The imagery is available in three visible color channels and one near infrared channel. All images have been taken with the same distance to the ground, which is typical for ISR scenarios. Additionally, the images have been downsampled by two to obtain images with a spatial resolution of 512×512, providing imagery with smaller, more challenging targets for evaluation. Hence, between the color and infrared channels and the two unique spatial resolutions, four subsets of data are available for experimentation, as shown in Table 1. The original images have a ground sampling distance (GSD) of 12.5 *cm per pixel (cmpp)*, while the downsampled images have a GSD of 25.0 cmpp.

While the VEDAI dataset contains 9 classes of objects, we focus our attention on small vehicles, namely the *car*, *pickup*, and *van* classes. Examples of these classes and the diversity with which they appear in the imagery are shown in Figure 4. A total of 2349 instances across these three classes are present in 982 images of the VEDAI dataset. We randomly choose 80% of the images for training (2007 instances) and the remaining 20% (342 instances) for testing. The VEDAI dataset contains extensive ground-truth annotations, including the centroid, orientation, and coordinates of the four corners of each instance.

From these extensive annotations, we chose to annotate vehicle instances in the following manner: retain the centroids and generate 40×40 pixel square bounding boxes around the centroids for the 1024×1024 resolution imagery and 20×20 pixel bounding boxes for the 512×512 imagery. This was done for several reasons. The faster R-CNN algorithm generates bounding-box predictions with *orthogonal* axes, so it is not possible to use the orientation component of the annotations. Second, upon observation of the VEDAI annotations, we found some of the bounding box annotations to be erroneous. Third, the use of padded bounding box regions facilitates the introduction of context with the annotations, which allows for the algorithm to learn various backgrounds in which targets are situated. The size of a 40×40 bounding box was chosen empirically by inspecting various instances in the data and determining the smallest size square bounding box that could encapsulate the instance at any possible orientation in the imagery. Examples of the annotations are shown in Figure 3. As the results will show, this labeling scheme does not detract from the performance of the algorithm. Furthermore, from a ground truthing standpoint for ISR applications, the efficiency of simply generating a *GSD-dependent* fixed-size bounding box around a centroid is more practical when annotating large datasets.

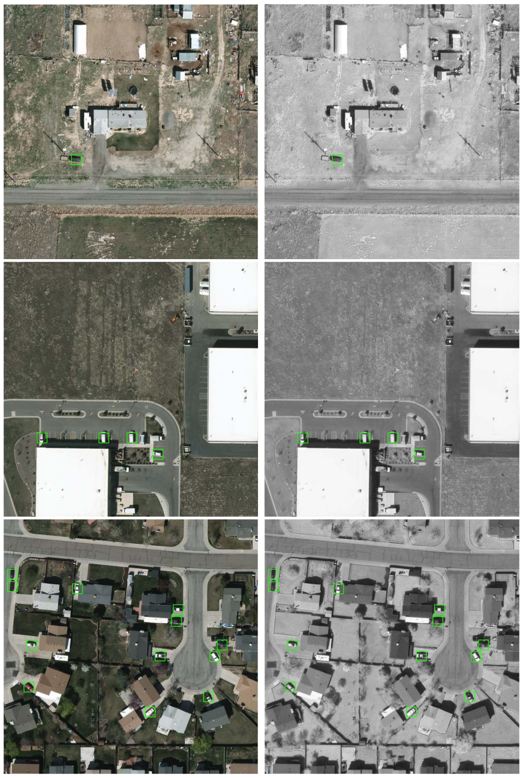

Figure 3. Coincident RGB (left) and IR (right) images from the 12.5 cmpp GSD VEDAI subset, illustrating the annotations overlaid on the vehicle classes of interest in several environments.

For performance evaluation, we use the standard *precision* and *recall* statistics to compute the *average precision* metric. The detection predictions are assigned as true/false positives based on the bounding box overlap with ground-truth instances. We use the PASCAL VOC [4] method for bounding box evaluation. To be considered a correct detection, the *intersection-over-union*, the area of overlap a_o between the predicted bounding box B_p and ground ground truth bounding box B_{gt}, must be greater than 0.5, as shown in eq. 1.

$$a_o = \frac{\text{area}(B_p \cap B_{gt})}{\text{area}(B_p \cup B_{gt})} \geqslant 0.5. \quad (1)$$

where $B_p \cap B_{gt}$ denotes the intersection of the predicted and ground truth bounding boxes and $B_p \cup B_{gt}$ their union. Each detection generated by faster R-CNN contains the predicted bounding box coordinates B_p as well as a confidence

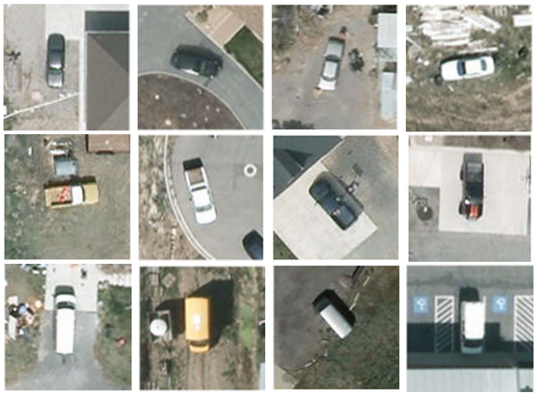

Figure 4. Examples of vehicle classes of interest from the VEDAI dataset. Notice the challenges present in this dataset due to diversity in target appearance, background variation, and presence of shadows and confuser objects. From top row to bottom row: car, pickup, and van.

score. Prior to the assignment of true/false positives via the overlap criterion in eq. 1, the detections are sorted by decreasing confidence output. Multiple detections of the same instance in an image are penalized, so if the algorithm generates three detections for the same object, only one may count as a true detection, while the other two will be considered false positives. Additionally, for a given image I and a particular minimum confidence threshold *t_conf*, we do not consider any detections d_I for which their corresponding confidence scores c_I are less than $t\_conf = 0.5$.

5. Faster R-CNN Modifications

This section discusses two important modifications to the faster R-CNN algorithm that we have explored for its successful application to overhead ISR imagery. The first modification involves the anchors used in the RPN module of faster R-CNN. The second modification involves the choice of which VGG-16 convolutional layer to use as input to the RPN.

5.1. RPN Anchor Sizes

As mentioned in section 3.2, faster R-CNN uses $k = 9$ anchors for the generation of region proposals. The default anchor scales are $\{128^2, 256^2, 512^2\}$ pixels, and the default aspect ratios are {2:1, 1:1, 1:2}, which cater to both the larger sizes of objects and more dramatic scale variances that are found in the iconic images present in traditional academic datasets such as PASCAL VOC. For aerial ISR image datasets such as VEDAI that contain targets on the order of tens of pixels, these scales are inadequate. Additionally, the use of several aspect ratios for scale invariance is unnecessary in the overhead ISR setting where the GSD is relatively fixed and can be estimated, given camera or GPS metadata or *a priori* known position for satellites. Accordingly, we have modified the RPN module of faster R-CNN to use a single anchor ($k = 1$), with a scale of 40^2 pixels for the 12.5 cmpp GSD imagery and a scale of 20^2 pixels for the 25 cmpp GSD imagery. The default settings and modifications are summarized in Table 2.

setting	scales (pixels)	aspect ratios
default	$\{128^2, 256^2, 512^2\}$	{2:1, 1:1, 1:2}
12.5 cmpp GSD	40^2	1:1
25.0 cmpp GSD	20^2	1:1

Table 2. RPN anchor settings for faster R-CNN. The first row lists the default anchor settings with $k = 9$, and the next two rows provide the proposed anchor settings with $k = 1$ for the subsets of VEDAI imagery.

5.2. RPN Shared Convolutional Layer

By default, the faster R-CNN algorithm using a VGG-16 network constructs the RPN by using the last convolutional layer, *conv5_3*, as input. However, depending on the sizes of the objects to be detected, this may not be ideal. For objects with a very low number of pixels on target, such as those found in VEDAI, and in contrast to those found in PASCAL VOC and Imagenet, it may prove more useful to use earlier convolutional layers as input to the RPN. As pointed out in [20], a single feature in the VGG-16 *conv5_3* layer activation map corresponds to 16 pixels in the input image. Hence, very small objects may incur a severe loss of information regarding the features used for region proposals. In the case of our VEDAI annotated small vehicles, only 2.5^2 and 1.25^2 pixels in the *conv5_3* feature map contribute to the vehicle feature representations in the large and small image subsets, respectively. Thus, similar to [20], we create faster R-CNN VGG-16 network models that input the *conv3_3* and *conv4_3* activation maps into the RPN module. However, in contrast to [20], for our application where scale invariance is not necessary, the choice of the selected convolutional layer is *fixed* for the entire training set.

6. Experiments

All experiments were carried out by modifying the Python re-implementation of the faster R-CNN repository using the CAFFE [9] framework. We use an initial learning rate of 0.001 with a *step* learning policy and decrease it by a value of 0.1 after 20k iterations. We use a momentum value of 0.9 and a weight decay of 0.0005 and train the models for a total of 60k iterations.

In our experiments, we train and test faster R-CNN models for each of the image subsets in Table 1. Per the faster R-CNN standard protocol, we use the image-centric sampling strategy, whereby 128 positive and

parameter	value
RPN_ANCHOR_BASE_SIZE	8 pixels
RPN_POSITIVE_OVERLAP	0.7
RPN_NEGATIVE_OVERLAP	0.3
RPN_BATCHSIZE	128 proposals
RPN_MIN_SIZE	40 pixels
RPN_NMS_THRESH	0.7
RPN_PRE_NMS_TOP_N	1000 proposals
RPN_POST_NMS_TOP_N	300 proposals

Table 3. Faster R-CNN RPN-specific training parameters.

Model	Precision	Recall	AP	F-measure
FT layers 3-5	0.9429	0.9669	0.9610	0.9546
FT layers 1-5	**0.9504**	**0.9705**	**0.9667**	**0.9604**

Table 4. Influence of number of fine-tunable layers of VGG-16 faster R-CNN model on VEDAI 1024 × 1024 color imagery detection performance. Values shown represent mean values of metrics taken after training for 60k total iterations at 5k snapshots.

negative object proposals are randomly sampled from one image to form a mini-batch. The positive object proposals are selected from all proposals that have greater than *RPN_POSITIVE_OVERLAP* with any positive ground-truth annotation.The negative object proposals are selected from all proposals that have less than *RPN_NEGATIVE_OVERLAP* with any positive ground-truth annotation. Data augmentation is utilized via horizontal image flipping. Table 3 provides the RPN-specific parameters during training of faster R-CNN. During testing, we use 100 RPN proposals and a non-maximum supression (NMS) value of 0.4 to eliminate redundant detections. For the 25 cmpp GSD imagery, we set *RPN_MIN_SIZE* to 20 pixels.

6.1. Influence of Number of Layers to Fine-tune

In the first experiment, we investigate the influence of the number of fine-tunable layers on detection performance. We take a VGG-16 network that has been pre-trained on Imagenet and fine-tune two separate faster R-CNN models on the 1024 × 1024 color imagery. The first model is trained by freezing the first two VGG-16 convolutional layers and fine-tuning layers $3-5$. The second model is trained by fine-tuning all five convolutional layers. We train faster R-CNN for 60k iterations, take model snapshots every 5k iterations, and compute the metrics discussed in Section 4. As shown in Table 4, taking the mean values of the metrics across iterations shows improved performance when fine-tuning all the layers of VGG16, which is consistent with empirical observations on other datasets reported in the literature. Hence, for future experiments, we fine-tune *all* layers of an Imagenet pre-trained VGG-16 model.

Conv Layer	Mean AP	Max AP
conv5_3	0.9667	0.9689
conv4_3	**0.9740**	**0.9819**
conv3_3	0.9716	0.9796

Table 5. LCI VEDAI subset. Influence of RPN convolutional layer on detection performance.

Conv Layer	Mean AP	Max AP
conv5_3	0.7313	0.7546
conv4_3	0.9029	0.9233
conv3_3	**0.9493**	**0.9583**

Table 6. SCI VEDAI subset. Influence of RPN convolutional layer on detection performance.

Conv Layer	Mean AP	Max AP
conv5_3	0.9418	0.9486
conv4_3	0.9477	0.9523
conv3_3	**0.9592**	**0.9697**

Table 7. LII VEDAI subset. Influence of RPN convolutional layer on detection performance.

Conv Layer	Mean AP	Max AP
conv5_3	0.7093	0.7265
conv4_3	0.8878	0.8977
conv3_3	**0.9345**	**0.9442**

Table 8. SII VEDAI subset. Influence of RPN convolutional layer on detection performance.

6.2. Influence of Selected Convolutional Layer for Region Proposals

Next, we outline the results of modifying the VGG-16 faster R-CNN network to use either *conv3_3*, *conv4_3*, or *conv5_3* as input to the RPN. In all of our models, we use a *region-of-interest* (ROI) pooling [6] size of 7 × 7. As before, we train faster R-CNN for 60k iterations, take model snapshots every 5k iterations, and report the mean of the average precision across all model snapshots as well as the model snapshot yielding the maximum average precision.

For the LCI subset results shown in Table 5, we find that the use of *conv4_3* provides a slight increase in the mean and max AP. For the smaller 20 × 20 objects in the SCI subset (Table 6), the use of both *conv4_3* and *conv3_3* provide substantial improvements in performance, with *conv3_3* yielding over a 20% increase in AP over the use of *conv5_3* RPN proposals. Figures 5 and 7 illustrate these impacts on detection performance for the LCI and SCI imagery, respectively.

The results on the infrared imagery are shown in Tables 7 and 8. In this case, the use of and *conv4_3* and *conv3_3* both provide improvements in AP over *conv5_3*, with *conv3_3*

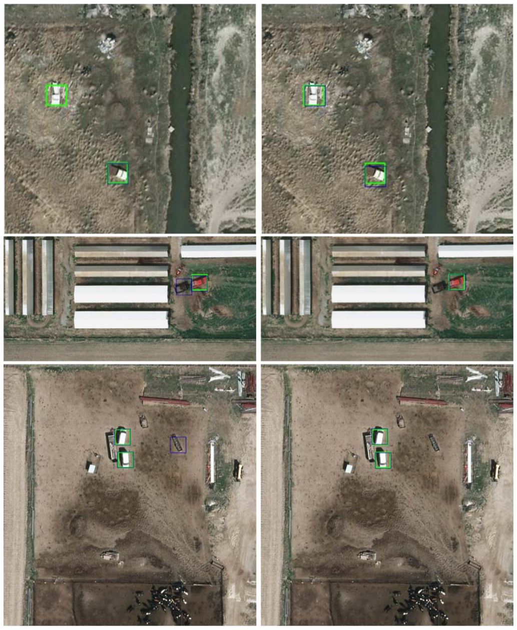

Figure 5. Sample detections from image regions in the LCI subset, illustrating differences between the use of *conv5_3* RPN proposals (left column) and *conv4_3* RPN proposals (right column). Green boxes denote ground-truth annotations, and blue boxes denote model predictions. The use of *conv4_3* RPN proposals leads to the detection of a missed true positive in the top row and the elimination of a false positive in the middle and bottom rows.

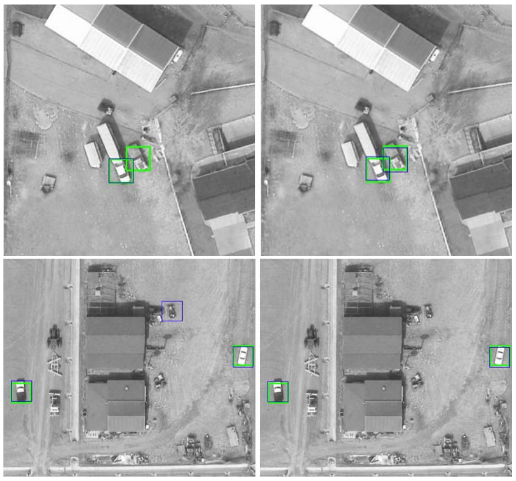

Figure 6. Sample detections from image regions in the LII subset, illustrating differences between the use of *conv5_3* RPN proposals (left column) and *conv3_3* RPN proposals (right column). The use of *conv3_3* RPN proposals leads to the detection of a missed true positive in the top row and the elimination of a false positive in the bottom row.

providing the best performance. As was the case for the smaller targets in color imagery, the performance improvement is substantial for the smaller targets in infrared imagery, with over a 22% increase in AP over the use of *conv5_3* RPN proposals. Figure 6 illustrates these impacts on detection performance for the LII imagery.

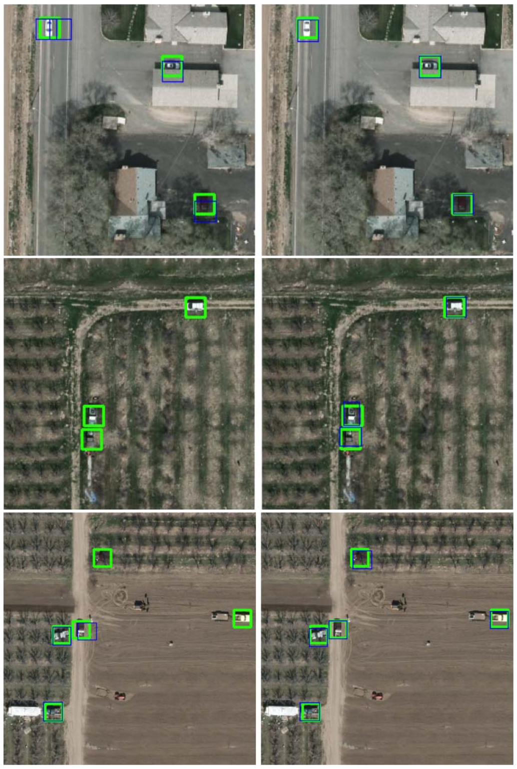

Figure 7. Sample detections from image regions in the SCI subset, illustrating differences between the use of *conv5_3* proposals (left column) and *conv3_3* proposals (right column). The use of *conv3_3* RPN proposals leads to the elimination of false positives and the detection of all true positives in the top row. In the middle row, all three true positives were missed using *conv5_3* RPN proposals but were all detected using *conv3_3* proposals. In the bottom row, the use of *conv3_3* proposals leads to the detection of all five true positives and more accurate bounding box predictions for targets in close proximity.

6.3. Cross-modality Transfer Learning

In this section, we conduct experiments that explore the transfer learning capabilities of faster R-CNN on the VEDAI dataset. In real-world scenarios, multi-modal imagery may not be simultaneously collected. Here, we outline two scenarios that would facilitate transfer learning of trained object detection models across modalities (e.g., COLOR, IR). In the first scenario, we assume that *labeled* imagery is first collected for a *source* modality, $\mathbf{M_S}$ and unlabeled imagery is then collected for a *target* modality, $\mathbf{M_T}$. Because no labels are present for $\mathbf{M_T}$, we may directly apply the model trained on $\mathbf{M_S}$ imagery to imagery from $\mathbf{M_T}$. In the second scenario, we assume that labeled imagery exists for both $\mathbf{M_S}$ and $\mathbf{M_T}$. In this scenario, we

$\mathbf{M_S}$	$\mathbf{M_T}$	Conv Layer	AP_S	AP_T	Δ_{AP}
C	IR	*3_3*	0.9796	0.8111	−16.85
C	IR	***4_3***	**0.9819**	**0.9088**	**−7.31**
C	IR	*5_3*	0.9689	0.8781	−9.08
IR	C	*3_3*	0.9697	0.6877	−28.20
IR	C	*4_3*	0.9523	0.9122	−4.01
IR	C	***5_3***	**0.9486**	**0.9328**	**−1.58**

Table 9. Cross-modal transfer learning, 1024 × 1024 imagery, no fine-tuning.

$\mathbf{M_S}$	$\mathbf{M_T}$	Conv Layer	AP_S	AP_T	Δ_{AP}
C	IR	*3_3*	0.9796	0.9650	−1.46
C	IR	***4_3***	**0.9819**	**0.9719**	**−1.00**
C	IR	*5_3*	0.9689	0.9672	−0.17
IR	C	*3_3*	0.9697	0.9516	−1.81
IR	C	***4_3***	**0.9523**	**0.9686**	**+1.63**
IR	C	*5_3*	0.9486	0.9630	+1.44

Table 10. Cross-modal transfer learning, 1024 × 1024 imagery, with fine-tuning.

are able to *fine-tune* models trained on $\mathbf{M_S}$ imagery with labeled images from $\mathbf{M_T}$.

In these experiments, we use the models with the max AP for $\mathbf{M_S}$ (see Tables 5 and 7) and also examine the effect of using models trained with different convolutional layers as input to the RPN proposal network, as discussed in Section 6.2. Tables 9 and 10 provide the transfer learning results using the 12.5 cmpp GSD imagery for the scenarios of no fine-tuning and fine-tuning, respectively. As shown in Table 9, in the absence of any fine-tuning for the target domain $\mathbf{M_T}$, AP rates above 90% and 93% are achieved for $\mathbf{M_S} = C$ and $\mathbf{M_S} = IR$, respectively, given the proper choice RPN convolutional layer (*conv4_3* and *conv5_3*, respectively). In the presence of fine-tuning, as shown in Table 10, the discrepancies between the AP values of $\mathbf{M_S}$ and $\mathbf{M_T}$ narrow drastically, independent of RPN convolutional layer.

6.4. Comparison to State-of-the-Art

Along with creation of the VEDAI dataset, Razakarivony *et al.* perform experiments with a variety of state-of-the-art sliding window detectors to serve as a baseline for comparison. In their experiments, they observed that a combination of HOG+LBP features with an SVM classifier provided the highest mAP [13]. Here, we compare the results of our faster R-CNN detection models to the performance of the HOG+LBP+SVM models in [13]. We emphasize that this is not a perfectly equivalent comparison for two reasons. In our experiments, we perform vehicle detection with respect to the *car*, *pickup*, and *van* classes, while the performance we compare against consists of results on the *small land vehicles* meta-class, which additionally includes the *tractor* class of objects. Additionally, the evaluation criterion that we outlined in section 4 regarding the scoring of true/false positives is different, with the evaluation protocol in [13] utilizing *ellipses* (which incorporate orientation annotation information) rather than rectangular regions. That being said, the average precision comparison results are summarized in Table 11. The faster R-CNN algorithm tuned for detecting small objects yields a substantial increase of approximately 17% in average precision across all the VEDAI subsets of imagery.

Image Subset	HOG+LBP+SVM [13]	Faster R-CNN
LCI	76.8 ± 1.5	**97.40**
LII	77.0 ± 1.6	**95.92**
SCI	74.9 ± 2.5	**94.93**
SII	75.0 ± 2.2	**93.45**

Table 11. Comparison of average precision of a state-of-the-art sliding window detector (HOG+LBP+SVM [13]) to faster R-CNN on the VEDAI subsets of imagery.

7. Conclusion

In this work, we have modified faster R-CNN, a state-of-the-art CNN-based object detection algorithm, to train models for localizing small vehicles in VEDAI, a challenging dataset of overhead aerial imagery. We have shown that the faster R-CNN parameters can be appropriately adjusted to handle the small, challenging targets that are present in VEDAI. By adjusting the configurations of the anchors present in the RPN module and providing selectivity of the VGG-16 convolutional feature map that is input to the RPN, we have demonstrated substantial improvements in mAP, relative to a prior state-of-the-art template-based sliding window method. We have also demonstrated that transfer learning is feasible and useful for sharing trained models across modalities. In the absence of fine-tuning on a target imagery modality, the faster R-CNN-based models yield AP rates above 90% and 93%, depending on the source image modality. Future work will explore methods for performing early and late decision fusion of coincident multi-modal image datasets such as VEDAI using deep CNN-based models tailored for object detection. We will also investigate other CNN-based object detection paradigms such as YOLO [14] and SSD [12] that do not require a region proposal step.

8. Acknowledgments

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

References

[1] Utah agrc website, 2012. 3

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 886–893, 2005. 2

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: a large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 1

[4] M. Everingham, L. Gool, C. Williams, J. Winn, and A. Zisserman. The pascal voc challenge. *Int. J. Comput. Vision*, 88:303–338, 2010. 1, 3

[5] P. F. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645, sep 2010. 2

[6] R. Girshick. Fast R-CNN. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015. 2, 5

[7] I. Goodfellow, Y. Bengio, and A. Courville. Deep learning. Book in preparation for MIT Press, 2016. 1

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2

[9] Y. Jia, E. Shelhamer, J. D. S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 4

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1

[11] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. 1

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision (ECCV) (to appear)*, 2016. 2, 7

[13] S. Razakarivony and F. Jurie. Vehicle detection in aerial imagery: A small target detection benchmark. *Visual Communication and Image Representation*, 34:187–203, January 2016. 1, 2, 7

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2, 7

[15] S. Ren, K. He, and R. G. J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Neural Information Processing Systems (NIPS)*, 2015. 1, 2

[16] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. Lecun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *International Conference on Learning Representations*, 2014. 2

[17] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *coRR*, abs/1409.1556, 2014. 2

[18] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 2013. 2

[19] P. Viola and M. J. Jones. Robust real-time face detection. *Int. J. Comput. Vision*, 57(2):137–154, may 2004. 1

[20] F. Yang, W. Choi, and Y. Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2129–2137, June 2016. 4

[21] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional neural networks. In *European Conference on Computer Vision (ECCV)*, pages 818–833, 2014. 1

[22] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, pages 391–405, 2014. 2