Journal of Imaging Science and Technology® 66(4): 040405-1–040405-9, 2022.

Military Target Detection in Remote Sensing Imagery Based on YOLOv4-Faster

Bang-Jui Wang
Ph.D. Program in Maritime Science and Technology, National Kaohsiung University of Science and Technology, Taiwan

Chih-Bin Hsu
Department of Information Management, R.O.C. Naval Academy, Taiwan

Jen-Chun Lee and Shang-Jen Chuang
Department of Telecommunication Engineering, National Kaohsiung University of Science and Technology, Taiwan
E-mail: i923002@nkust.edu.tw

Chung-Hsien Chen
Metal Industries Research & Development Centre (MIRDC), Taiwan

Te-Ming Tu
Department of Telecommunication Engineering, National Kaohsiung University of Science and Technology, Taiwan

Abstract. *The continuing development of remote sensing has resulted in a rapidly increasing number of remote sensing applications. High-resolution remote sensing images are used in various fields in the military. We propose methods for object detection based on remote sensing images. We develop a signal processing method for normalizing remote sensing images to eliminate noise such as fog, haze, and poor lighting. This method further improves detection accuracy and reduces error rates. We develop YOLOv4-faster, an accelerated neural network model based on the YOLO (You-Only-Look-Once) object detection method. YOLOv4-faster outperforms existing networks in terms of execution time and detection performance. We conduct a series of experiments on two public datasets (TGRS-HRRSD and NWPU VHR-10) as well as a dataset containing six military target classes provided by IMINT & Analysis and collected from Google Earth. YOLOv4-faster improves efficiency by utilizing multi-scale operations for the accurate detection of objects of various sizes, especially small objects. The experimental results show improved mAP (mean average precision) performance of the proposed method for object detection in remote sensing images. We thus propose a novel system for automatic object detection for high-resolution remote sensing images.* © *2022 Society for Imaging Science and Technology.*
[DOI: 10.2352/J.ImagingSci.Technol.2022.66.4.040405]

1. INTRODUCTION

Remote sensing technology is a modern detection technology that emerged in the 1960s. Over the years, it has provided stable and detailed data for land use analysis, agricultural pest monitoring, urban planning, and other civil fields. It is important for military applications such as military target detection, battlefield environment simulation, and so on. Currently, remote sensing technology is capable of high spectral, high spatial resolution, and all-weather earth observation. As these three resolutions continue to improve, remote sensing data has witnessed explosive growth. Therefore, automatic information extraction of high-resolution remote sensing images has become a research focus in every country.

Since the success of deep learning in pattern recognition, we have witnessed huge advances in the field of computer vision. Object recognition is crucial for autonomous cars, security, surveillance, and industrial applications which use deep learning methods such as region-based convolutional neural networks (R-CNN) [1], single-shot MultiBox detectors (SSD) [2], You-Only-Look-Once (YOLO) [3], and deep residual networks (ResNet) [4]. In recent years, deep learning technology has been widely used to extract features from high-resolution remote sensing images, and significant achievements have been made in remote sensing scene classification, remote sensing target detection, remote sensing image description, segmentation, and other tasks. However, these methods still suffer from poor performance and heavy CPU requirements. In this paper, we explore remote sensing target detection. The research work and contributions of this paper mainly include the following aspects:

(1) We propose a new dehazing algorithm for remote sensing images that processes the image to facilitate either visualization or further analysis.

(2) As too much or too little feature extraction do not benefit remote sensing target detection, we propose a novel neural network architecture to replace the existing YOLO-based method.

Received Jan. 18, 2022; accepted for publication June 14, 2022; published online July 13, 2022. Associate Editor: Jia-Shing Sheu.

1062-3701/2022/66(4)/040405/9/$25.00

(3) To improve multi-scale target detection, we reduce the original layers of YOLOv4 [5] and modify the original two output layers of YOLOv4-tiny [6] to three output layers.
(4) The proposed neural networks can adjust the network structure dynamically based on the remote sensing target type.
(5) Compared with classical deep learning networks, the proposed algorithms reduce the network layers and weights to decrease the computing cost without lowering the target detection performance.

Rest of the paper is organized as follows. Section 2 briefly introduces related work. A detailed description of the proposed strategy for object detection and image processing are given in Section 3. The experimental results and comparisons with other object detection methods are discussed in Section 4, and Section 5 concludes and proposes future work.

2. RELATED WORK

The development of remote sensing target detection follows that of general object detection. Traditionally, object detection methods include two steps: region of interest (ROI) selection and independent feature extraction from each region for classification. Although it is a viable method, ROI generation with a sliding window strategy is redundant and inaccurate. In 2012, Alex Krizhevsky et al. [7] won the ILSVRC competition by using an AlexNet model with a 7-layer convolutional neural network (CNN), which resulted in an increased general interest in deep learning. In the past two years, deep learning based target detection on high-resolution remote sensing images has become a popular method. Compared with traditional target detection algorithms such as the Viola–Jones detector [8], histogram of oriented gradients-support vector machine (HOG-SVM) [9], and the famous deformable parts model (DPM) algorithm [10], CNN-based target detection algorithms have shown great success, especially in terms of speed and accuracy. Deep learning is now widely applied in fields such as speech recognition [11] and many object detection and recognition tasks [12, 13], where they outperform traditional methods.

Generally speaking, deep learning methods for object detection are dominated by CNN-based algorithms, which can be divided into two-stage models and one-stage models. Two-stage models such as faster R-CNN (region-based convolutional neural networks) [14] or mask R-CNN [15] are similar to traditional methods. They use a region proposal network (RPN) to generate ROIs in the first stage from which the detected objects are selected and CNN features extracted for object recognition. Two-stage models attain higher accuracy rates, but are typically slower when making predictions as compared to alternate models such as one-stage models that may be less accurate but are designed for real-time prediction. Based on global regression and classification, a one-stage model locates objects, directly mapping from image pixels to bounding box coordinates and class probabilities. One-stage architectures thus reduce computing costs and can be used in real-time applications. In addition, experimental results for one-stage models based on YOLO (You-Only-Look-Once) models [3] demonstrate optimal speed and accuracy, which facilitate the processing of streaming video in real time. After the YOLO model was developed, YOLO object detection models (YOLOv2 and YOLOv3) were proposed successively by Redmon et al. [16]. Note that YOLOv3 [17] uses darknet-53, a deeper architecture for the feature extractor, and it detects at three different scales to correct the main shortcomings of YOLO and YOLOv2. Detection at different scales makes it easier to detect small objects, especially in remote sensing images. Moreover, YOLOv3 improves accuracy using algorithms such as feature pyramid networks (FPNs), path aggregation network (PANets), and spatial pyramid pooling (SPP), making it more accurate than YOLO and YOLOv2, but slower.

In 2020, Bochkovskiy et al. proposed the YOLOv4 model, an improvement to YOLOv3 with improved accuracy [5]. They effectively integrated deep learning algorithms to create an efficient neural network for object detection. For better accuracy, they proposed CSPDarknet-53, which uses CSP (cross stage partial) connections along with darknet-53 from YOLOv3. It is also easier to train this neural network on a single GPU. After YOLOv4, a series of methods were proposed, including YOLOv5 [18], YOLOv4-tiny [6], and scales-YOLOv4 [19]. YOLO-based object detection is thus becoming a dominant trend, and YOLOv4 yields excellent results in object detection of natural images. However, previous methods such as SSD [2], YOLT [20], and YOLOv3 models are still used for target detection in remote sensing images. To the best of our knowledge, there is no study that uses the YOLOv4 model for target detection of remote sensing images.

The YOLOv4 methods claim to offer an optimal neural network architecture for object detection. For the 80 classes in the MS COCO dataset, YOLOv4-based object detectors are clearly faster and more accurate than other detectors. However, in practical applications it is not necessary to simultaneously detect 80 object categories. In the real world, the demand for specific target detection often includes only one or a few objects. Perhaps, then, using a YOLOv4 model to detect a few specific targets constitutes overengineering. Therefore, we seek to evaluate the feasibility and effectiveness of the proposed method to reduce and modify the YOLOv4 neural network. We propose a new YOLOv4-based neural network model for object detection. This model provides automatic target detection with high-resolution remote sensing images, especially for small targets.

3. PROPOSED METHOD

Since the size of each remote sensing image is at least 12,000 × 12,000 pixels, the target detection computation time for large images is important. Targets must be detected in real time and with performance equal to the YOLOv4 model. This

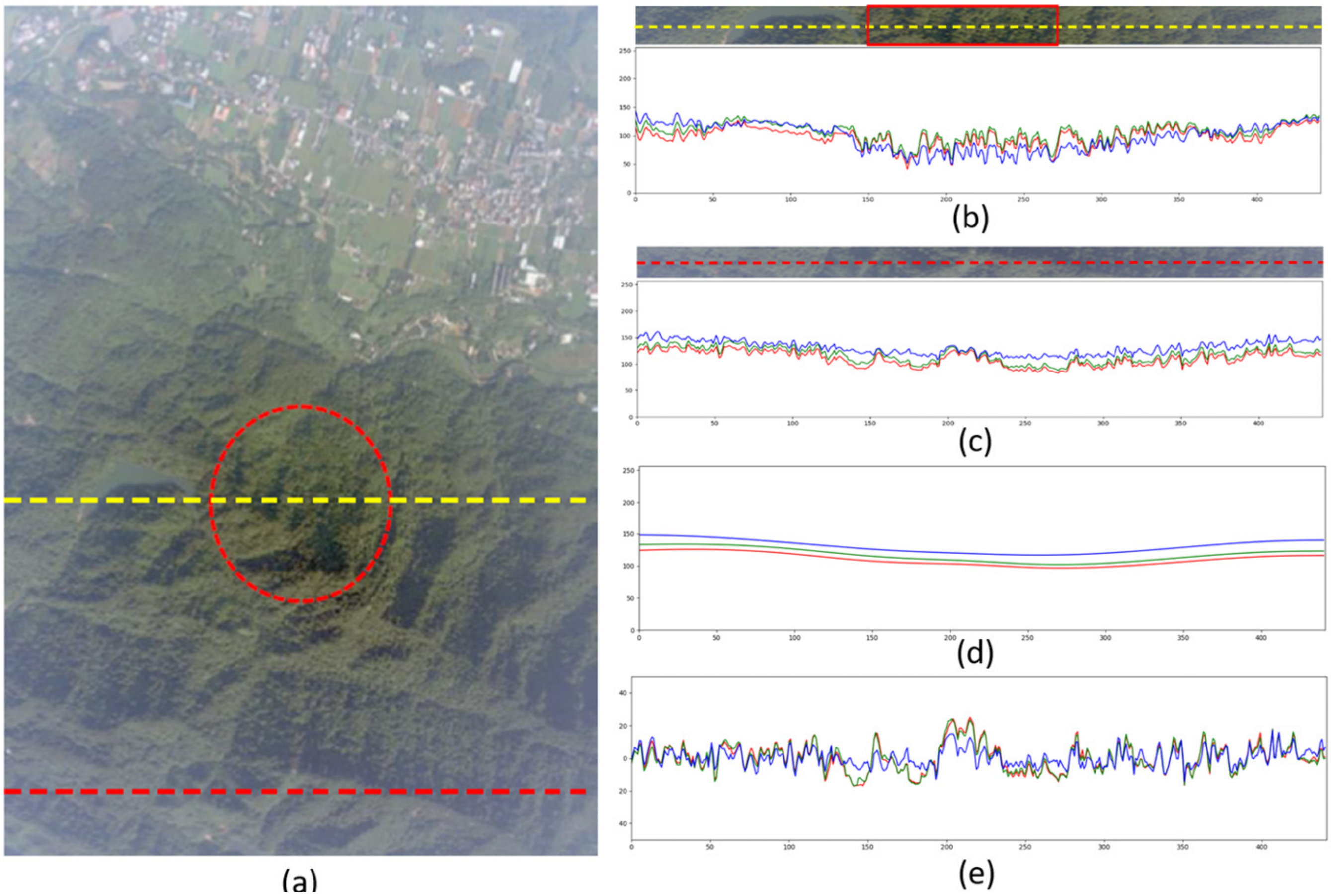

Figure 1. (a) Hazy remote sensing image; histogram of (b) yellow line and of (c) red line; (d) $f_{i,\mathrm{DC}}$ obtained using low-pass filter; (e) $f_{i,\mathrm{AC}}$ obtained using Eq. (1).

section describes the proposed framework for military target detection using high-resolution remote sensing images. The solutions and steps are described in detail below.

3.1 *Image Preprocessing*

Image quality affects target detection results. Generally speaking, remote sensing images are characterized by atmospheric conditions such as fog, haze, and inconsistent lighting, which results in low image quality. Therefore, image dehazing is an important issue for this task. Before detecting targets in a remote sensing image, we must dehaze the image. Many methods use enhancement and conventional image processing techniques to remove haze from a single image, such as histogram-based [21] and contrast-based dehazing methods [22]. However, little information is available for single images, limiting the effects of dehazing. Miyazaki et al. [23] propose polarization-based methods to improve dehazing performance with multiple images, and He et al. [24] propose an empirical statistics-based dark channel prior (DCP) which uses statistics of haze-free images for single image dehazing. In addition to the DCP, many other prior-based methods have been proposed [25, 26]. Although these approaches are efficient and often produce superior dehazing effects, they do not perform well in all cases. In recent years, convolutional neural networks (CNNs) have shown great success in computer vision. Many CNN-based methods have also been developed for image dehazing. Gu et al. [27] proposed the dense attentive dehazing network (DADN) for remote sensing image dehazing. Although some methods have achieved great breakthroughs in image dehazing, some dehazing methods fail to remote sensing images since remote sensing images are different from normal images. We address this problem with a new method for remote sensing image dehazing using signal processing techniques.

Figure 1(a) is a reduced-size remote sensing image with haze. The clear area in the middle of Fig. 1(a) is caused by the "spotlight effect" and shows how haze affects the image quality. In Fig. 1(a), the yellow dotted line passes through the spotlight area and the red line passes through the hazy area. Using signal processing techniques, we use the changes in pixel values to observe variation between haze and no haze (the histograms in Fig. 1(b) and Fig. 1(c) correspond to the yellow and red lines, respectively). The B band is most affected by haze, as shown in Fig. 1(b). Hazy images show significant increases in the R and G bands; change in the B band is significantly reduced in clear images. Note that there is little change in the gray value for the hazy image, but the

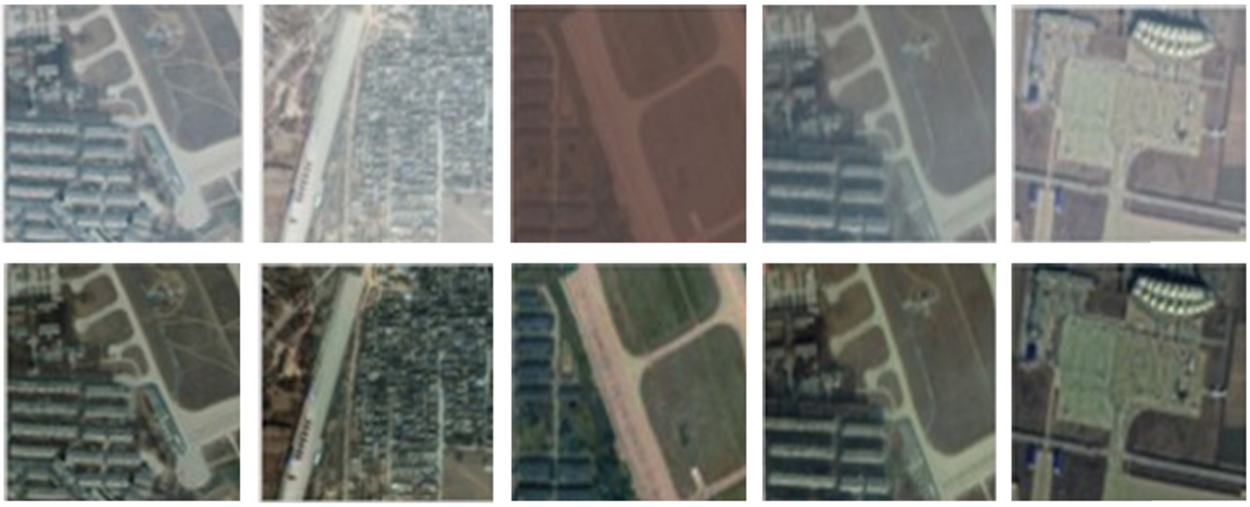

Figure 2. Remote sensing images before and after dehazing.

level does shift upward. That for the clear image, in contrast, changes greatly.

With signal processing, each band can be written as

$$f_i = f_{i,\mathrm{DC}} + f_{i,\mathrm{AC}}, \tag{1}$$

where $i = R, G, B$. Fig. 1(d) applies a low-pass filter to Fig. 1(c) to yield $f_{i,\mathrm{DC}}$; the small signal of the zero level is $f_{i,\mathrm{AC}}$. However, in Fig. 1(b), the gray value in the spotlight area is lower than that in the other areas, allowing us to define how haze affects the image. We find the minimum value $\min(f_{i,\mathrm{DC}})$ in Fig. 1(d). Equation (2) reveals the amount of haze in each pixel: $H_{\mathrm{impact}} = 0$ describes a pixel without haze, and $H_{\mathrm{impact}} \neq 0$ indicates the amount of haze in each pixel:

$$H_{\mathrm{impact}} = f_{i,\mathrm{DC}} - \min(f_{i,\mathrm{DC}}). \tag{2}$$

In Fig. 1(e), $f_{i,\mathrm{AC}}$, to dehaze and enhance the image, we amplify hazy pixels and leave clear pixels alone. Therefore, we set a compensation factor K_i to

$$K_i = \frac{H_{\mathrm{impact}}}{255}. \tag{3}$$

Therefore, $f_{i,\mathrm{AC}}$ and $(1 + K_i)$ are multiplied to produce g_i, which can be written as

$$g_i = (1 + K_i) \cdot f_{i,\mathrm{AC}}. \tag{4}$$

Here, g_i dehazes and enhances the original image f_i. In Eq. (4), 1 is the original signal of $f_{i,\mathrm{AC}}$. However, the equation does not include smooth areas in the image ($f_{i,\mathrm{DC}}$), so we add the original $f_{i,\mathrm{DC}}$ as

$$Dehaze_i = (1 + K_i) \cdot f_{i,\mathrm{AC}} + f_{i,\mathrm{DC}}. \tag{5}$$

Figure 2 is the result of remote sensing image dehazing ($Dehaze_i$) using Eq. (5), which is proposed to adjust the image to facilitate visualization or further analysis.

3.2 *Architecture of Proposed Method*

Given the large size of each remote sensing image, the computational time for target detection is important because we require real-time target detection with YOLOv4-level performance. This section describes the proposed framework for military target detection with high-resolution remote sensing images. For the network structure we use the existing CNN architecture based on YOLO; YOLOv4-tiny is a simple network structure with CSPdarknet53-tiny as a backbone and which detects objects at two different scales to capture both big and medium objects. As remote sensing images consists mostly of small targets, YOLOv4-tiny cannot be used for target detection with remote sensing images; thus, we propose the use of YOLOv4-faster in this paper.

At the first level, YOLOv4-tiny downsamples to reduce computational costs, resulting in the initial loss of half the information. Despite the strengths of YOLOv4-tiny, it still does not extract the proper features for target detection, especially for small targets. YOLOv4-faster, by contrast, retains more information without downsampling at the first level, as shown in Figure 3. Next, the backbone of the proposed model uses the three residual networks (ResNet) and CSPdarknet53-tiny to extract more information. Finally, for cases in which most of the remote sensing images contain small targets, YOLOv4-faster uses three different scales to detect large, medium, and small targets, respectively, compared to YOLOv4-tinys two scales for detection of large and medium targets. Note that this modification is important for remote sensing images. YOLOv4, YOLOv4-tiny, and YOLOv4-faster have 245 MB, 23 MB, and 26 MB weights, respectively. The corresponding computational costs are 60.1 BFlops, 6.9 BFlops, and 9.26 BFlops.

Table I describes each layer of the neural network framework for military target detection. A 55-layer convolutional architecture is the basis of YOLOv4-faster. The feature extractor starts with a standard convolutional layer with 32 filters of size 3 × 3. To improve the accuracy of small target detection, YOLOv4-faster adds 7 layers (layers

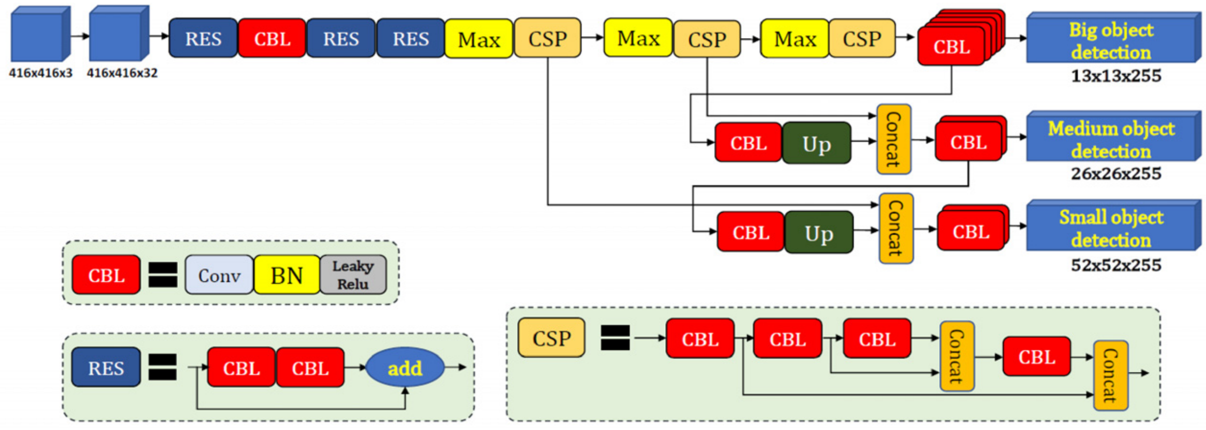

Figure 3. YOLOv4-faster structure.

49–55) to detect small targets. In addition, if only big targets are to be detected, such as bridges, storage tanks, and tennis courts, YOLOv4-faster can just use the first 41 layers. This increases the accuracy of big target detection and also reduces the computational costs. Therefore, for big targets we use a 41-layer network architecture instead of the entire YOLOv4-faster network.

4. EXPERIMENT RESULTS AND ANALYSIS

In this section, we evaluate YOLOv4-faster on two public remote sensing datasets and our collected military target dataset (described in Section 4.1). All experiments were implemented using the CUDA C++ API, on an i9 Intel CPU with 32GB RAM and a GTX 2080Ti GPU with 11GB on-chip memory. As performance metrics, we used average precision (AP) with an intersection over union (IOU) threshold of 0.5 and the average detection time per image. To verify the validity of our approach for remote sensing target detection, we compared YOLOv4-faster with the original YOLOv4, YOLOv4-tiny, and other state-of-the-art CNN algorithms on the three datasets. We implemented 100,000 training steps in this experiment. The learning rate of the model was decreased from 0.001 to 0.0001 after 80,000 steps and to 0.00001 after 90,000 steps. We used the same parameters for other comparison algorithms.

4.1 *Dataset*

Although publicly available remote sensing and aerial image datasets such as UC Merced Land Use [28], COWC [29], DOTA [30], and DIOR [31] have been proposed in the earth observation community, some are not available for download on the Internet, others contain only one or two categories, and others do not satisfy specific military requirements. Therefore, we used IMINT & Analysis [32] and Google Earth to collect military targets. Compiled in 2009, IMINT & Analysis provides the longitude and latitude of military bases in some countries. We also used Google Earths historical imagery to collect more military target images. We collected 12,148 military target images with a spatial resolution from 0.4 m to 1 m, and an image resolution of 800 × 800. The dataset contains a total of 95,740 military targets instances in 6 categories: fighter planes, helicopters, missile positions, military radar, battleships, and submarines. Most are small-scale targets. We partitioned the dataset into a training set (80%) and a testing set (20%). In addition, to ascertain the feasibility of our proposed method for non-military target detection of remote sensing images, we used two popular multi-class remote sensing image datasets: NWPU VHR-10 [33] and TGRS-HRRSD [34]. We summarize the two datasets as follows.

The TGRS-HRRSD dataset contains 21,761 images acquired from Google Earth and Baidu Map with spatial resolutions from 0.15 m to 1.20 m. The dataset contains a total of 55,740 target object instances in 13 categories, and is divided into three subsets: a training set, a validation set, and a test set. The train-val set (training+validation) and test set each account for 50% of the total dataset. The second dataset is NWPU VHR-10, which consists of 800 very high resolution (VHR) optical remote sensing images that include 10 object categories for research purposes only. The dataset contains two folders: one is a positive image set that includes 650 images, and the other is a negative image set that includes 150 images. Of these, 715 images were cropped from Google Earth with a spatial resolution from 0.5 m to 2.0 m, and 85 pan-sharpened color infrared images were acquired from the Vaihingen dataset with a spatial resolution of 0.08 m. All images in the positive image set were manually annotated by experts. To evaluate the proposed algorithm, we randomly split the NWPU VHR-10 dataset into training and testing sets. In this paper, we used an 80:20 ratio for the training and test data, and used five-fold cross validation.

4.2 *Validation of Proposed Method*

To properly evaluate the effect of the proposed method, we conducted an experiment on the original and enhanced (dehazed) datasets. For the six military target categories, we also verified the target detection performance. In our dataset, fighter planes, helicopters, military radars, submarines and some battleships are regarded as small targets, other

Table I. YOLOv4-faster network architecture.

Layer	Operation type	Input	Filter	Size/stride	Output
0	Convolution	416 × 416 × 3	32	3 × 3/1	416 × 416 × 32
1	Convolution	416 × 416 × 32	64	3 × 3/2	208 × 208 × 64
2-4	Residual network	208 × 208 × 64	–	–	208 × 208 × 64
5	Convolution	208 × 208 × 64	128	3 × 3/2	104 × 104 × 128
6–8	Residual network	104 × 104 × 128	–	–	104 × 104 × 128
9–11	Residual network	104 × 104 × 128	–	–	104 × 104 × 128
12	Max pooling	104 × 104 × 128	–	2 × 2/2	52 × 52 × 128
13–19	CSPNet	52 × 52 × 128	–	–	52 × 52 × 256
20	Max pooling	52 × 52 × 256	–	2 × 2/2	26 × 26 × 256
21-27	CSPNet	26 × 26 × 256	–	1	26 × 26 × 512
28	Max pooling	26 × 26 × 512	–	2 × 2/2	13 × 13 × 512
29–35	CSPNet	13 × 13 × 512	–	1	13 × 13 × 1024
36	Convolution	13 × 13 × 1024	512	1 × 1/1	13 × 13 × 512
37	Convolution	13 × 13 × 512	512	3 × 3/1	13 × 13 × 512
38	Convolution	13 × 13 × 512	256	1 × 1/1	13 × 13 × 256
39	Convolution	13 × 13 × 256	512	3 × 3/1	13 × 13 × 512
40	Convolution	13 × 13 × 512	255	1 × 1/1	13 × 13 × 255
41			Large object detection		
42	Route	38	–	–	
43	Convolution	13 × 13 × 256	128	1 × 1/1	13 × 13 × 128
44	2 × Upsampling	26 × 26 × 128	–	–	26 × 26 × 128
45	Concatenation	26, 44	–	–	26 × 26 × 384
46	Convolution	26 × 26 × 384	256	3 × 3/1	26 × 26 × 256
47	Convolution	26 × 26 × 255	255	1 × 1/1	26 × 26 × 255
48			Medium object detection		
49	Route	46			26 × 26 × 256
50	Convolution	26 × 26 × 256	64	1 × 1/1	26 × 26 × 64
51	2 × Upsampling	26 × 26 × 64	–	–	52 × 52 × 64
52	Concatenation	18, 46	–	–	52 × 52 × 192
53	Convolution	52 × 52 × 192	128	3 × 3/1	52 × 52 × 128
54	Convolution	52 × 52 × 128	255	1 × 1/1	52 × 52 × 255
55			Small object detection		

battleships (aircraft carriers) regarded as medium targets, and missile positions are regarded as big targets, as shown in Figure 4.

In this work, these images were resized into squares of three different sizes: 416 × 416, 608 × 608, and 800 × 800. Table II shows the detection results on our collected dataset for various image resolutions as well as the mean average precision (mAP). Using the 800 × 800 images for target detection yields the best performance, and 416 × 416 images are too small for effective detection; using the larger images improves detection performance. The results also show better performance for missile position detection on the enhanced dataset. For instance, after enhancing and dehazing the remote sensing images, our proposed method achieves better performance than the original dataset without enhancement, improving the mAP from 97.08 to 98.08%. In our experiments, we found that most missile positions and military radars were successfully detected by the proposed method. Most errors occurred with small fighter planes, battleships, and submarines, which are difficult to distinguish from similar ground objects. As fighter planes have different structures, they often lead to false detection because of unclear images or because ground objects can be similar in structure to fighter planes. In addition, submarines and battleships have lower mAPs because most battleships and submarines are docked side by side and are thus regarded as a single target. These are the causes of most detection errors. In Table II, we also break out the results by category to demonstrate the strong performance for small object detection.

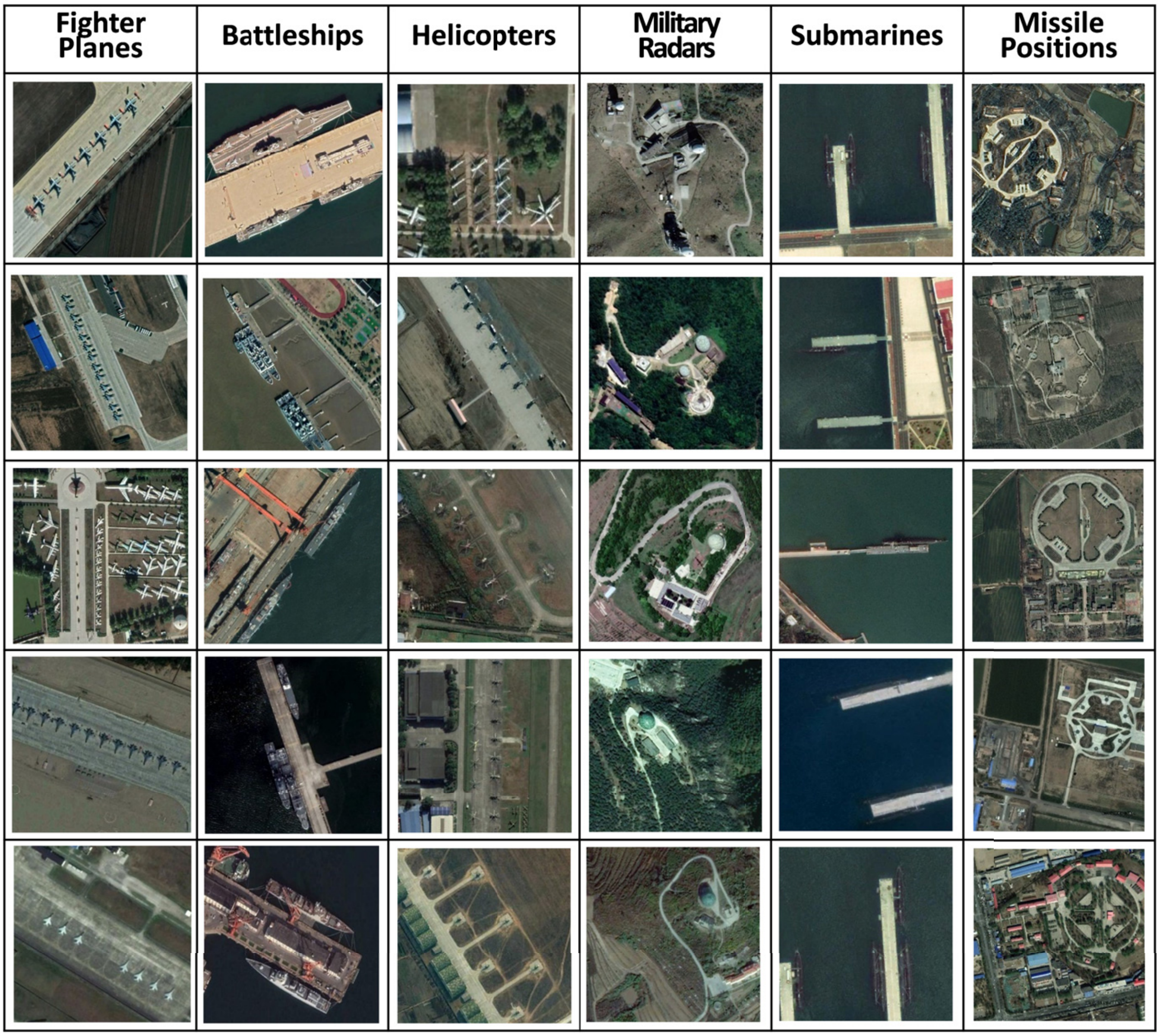

Figure 4. Military target classes for collected dataset.

Table II. Results of proposed method on military target dataset. Target types: FP=fighter planes, H=helicopters, MP=missile positions, MR=military radars, B=battleships, and S=submarines.

					Target types			
Dataset	Resolution	FP	H	MP	MR	B	S	mAP
Original dataset	416*416	92.13	93.55	98.57	94.65	83.41	85.12	91.24
	608*608	98.76	99.23	99.83	98.93	92.66	96.44	97.64
	800*800	97.71	97.46	99.93	98.53	93.92	94.85	97.06
Enhanced dataset	416*416	93.48	95.37	99.69	95.41	88.67	88.61	93.54
	608*608	97.18	97.10	99.84	98.08	92.87	92.87	96.24
	800*800	98.11	97.61	100	99.92	94.93	97.92	98.08

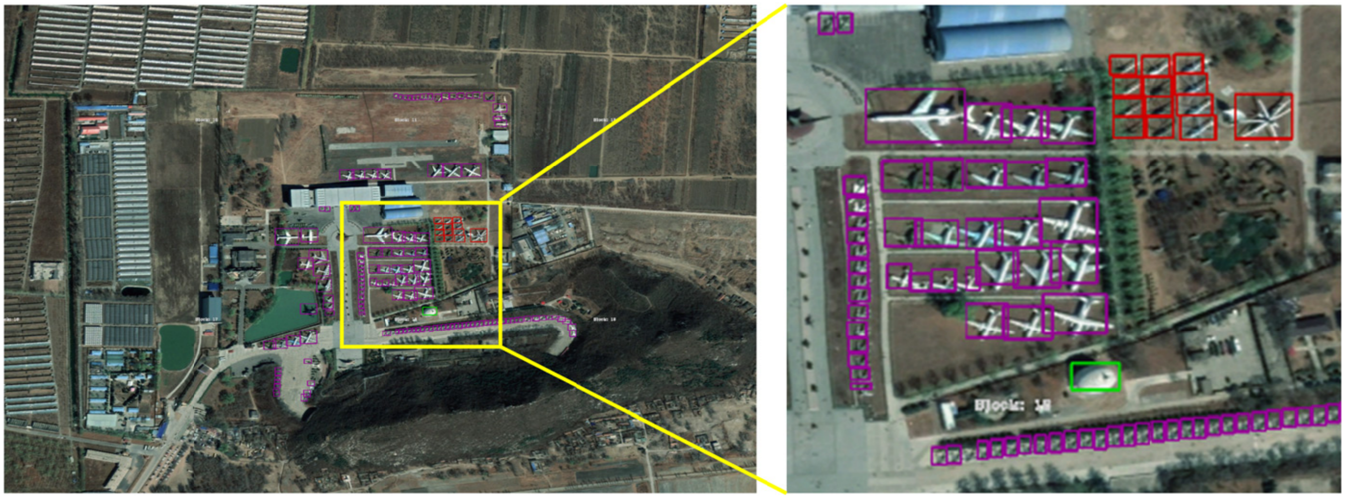

Figure 5. Multiple-target results using YOLOv4-faster.

Table III. Result of three datasets with different methods.

Dataset	YOLOv5	YOLOv4	YOLOv4-tiny	Scaled-YOLOv4	Proposed
TGRS-HRRSD	95.5%	88.7%	65.4%	96.9%	96.3%
NWPU VHR-10	90.3%	90.0%	79.2%	91.3%	91.2%
Our dataset	96.5%	95.6%	71.3%	97.1%	98.1%

Table IV. Results with different methods.

Method	mAP (%)	Precision	Recall	F1 score	FPS
YOLOv5	96.5	0.97	0.91	0.93	38
YOLOv4	95.6	0.98	0.97	0.97	18
YOLOv4-tiny	71.3	0.99	0.56	0.71	71
Scaled-YOLOv4	97.1	0.86	0.94	0.89	27
Proposed method	98.1	0.97	0.98	0.97	65

4.3 *Comparison with Other Methods*

In this section, to evaluate the performance of the proposed method for remote sensing target detection in terms of both accuracy and efficiency, we compare it with state-of-the-art models: YOLOv5, YOLOv4, YOLOv4-tiny, and scaled-YOLOv4. YOLOv4 and YOLOv5 use deeper layers of CSPDarknet53 to replace darknet53 of YOLOv3 [17] to obtain more features for the 80 object types. However, YOLOv4-tiny and scaled-YOLOv4 are compatible subset implementations of YOLOv4. All detection methods use the same training and test sets, and the evaluation of detection results was performed using the same standard. Table III shows that our collected dataset yields higher accuracy than the TGRS-HRRSD and NWPU VHR-10 datasets. Additionally, the proposed method outperforms YOLOv5, YOLOv4, and YOLOv4-tiny, and is similar to scaled-YOLOv4. Because YOLOv4-tiny is less sensitive to small objects, it has lower performance. The three datasets show similar results for target detection of optical remote sensing images. Furthermore, the superiority of YOLOv4-faster over YOLOv5, YOLOv4, and YOLOv4-tiny demonstrates that the proposed network architecture yields better performance.

The experimental results for our collected dataset at 800 × 800 resolution are shown in Table IV in comparison with existing deep learning algorithms. YOLOv4-faster yields a precision of 97%, a recall of 98%, and an mAP of 98%. In addition to performance, speed is also important for neural network architectures. YOLOv4-tiny is clearly the fastest with a detection speed significantly higher than that of the other algorithms, but performs poorly. However, the execution speed of our algorithm is 65 FPS, which meets the requirements of real-time detection. Table IV further illustrates the excellent performance of the proposed method for military target detection.

Figure 5 shows the multiple-target detection results in our collected datasets, with three categories of objects: fighter planes (purple border), helicopters (red border), and military radars (green border). The results demonstrate that the proposed framework enables relatively fine-grained, accurate detection of multiple objects in complex, high-resolution scenes, and is strongly robust against interference such as illumination, shadows, and occlusion, and yields reasonable detection performance for small objects such as fighter planes, helicopters, battleships, military radars, and submarines. In summary, YOLOv4-faster achieves higher and more stable detection accuracy than state-of-the-art algorithms. For small training sets (NWPU VHR-10), YOLOv4-faster also exhibits faster convergence speeds. Future studies will continue to extend the dataset, host more challenges, and integrate more algorithms for military target detection.

5. CONCLUSION

We introduce a framework composed of a series of solutions and steps for military target detection in remote sensing imagery. The YOLOv4-faster model is proposed, which uses a modified YOLOv4-tiny module as the feature extraction network and uses three different scales for detection of large, medium, and small targets, respectively. The experimental results indicate that YOLOv4-faster outperforms several state-of-the-art architectures on three remote sensing scene datasets. Moreover, we propose a new dehazing algorithm for remote sensing images that produces images, which are more suitable for further target detection and increases the target detection accuracy. The proposed model achieves the best detection performance among the five methods evaluated: an mAP of 98.08% and 65 FPS on our collected dataset—note that the mAP value exceeds that of the suboptimal method (scaled-YOLOv4) by 0.95%. In addition, the proposed model also yields mAPs of 96.3% and 91.2% on the TGRS-HRRSD and NWPU VHR-10 datasets. In future work, we will improve our method by conducting further research on the detection of blurred, dense small objects and obscured objects.

REFERENCES

[1] R. Girshick, J. T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2014), pp. 580–587.

[2] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: single shot MultiBox detector," in *Computer Vision – ECCV 2016*, edited by B. Leibe, J. Matas, N. Sebe, and M. Welling, Lecture Notes in Computer Science (Springer, Cham, 2016), Vol. 9905.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition CVPR* (IEEE, Piscataway, NJ, 2016), pp. 779–788.

[4] K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, "Deep residual learning for image recognition," *IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2016), pp. 27–30.

[5] A. Bochkovskiy, C. Y. Wang, and H. Y. Mark Liao, "YOLOv4: optimal speed and accuracy of object detection," *Proc. Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2020), pp. 14–19.

[6] A. Bochkovskiy, *Darknet: Open Source Neural Networks in Python* (2020), Available online: https://github.com/AlexeyAB/darknet.

[7] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," Neural Inf. Process. Syst. **25**, 1097–1105 (2012).

[8] P. Viola and M. Jones, "Robust real-time object detection," Int'l. J. Comput. Vis. **4**, 34–47 (2001).

[9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2005), pp. 886–893.

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

[11] M. Yousefi and J. H. L. Hansen, "Block-based high-performance CNN architectures for frame-level overlapping speech detection," IEEE/ACM Trans. Audio Speech Lang. Process. **29**, 28–40 (2021).

[12] H. Park, S. Park, and Y. Joo, "Detection of abandoned and stolen objects based on dual background model and mask R-CNN," IEEE Access **8**, 80010–80019 (2020).

[13] Y. Song, B. He, and P. Liu, "Real-time object detection for AUVs using self-cascaded convolutional neural networks," IEEE J. Ocean. Eng. **46**, 56–67 (2021).

[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," IEEE Trans. Pattern Anal. Mach. Intell. **39**, 1137–1149 (2016).

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *Proc. IEEE Int'l. Conf. on Computer Vision* (IEEE, Piscataway, NJ, 2017), pp. 22–29.

[16] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," *IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2017), pp. 21–26.

[17] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2018), pp. 2311–2314.

[18] J. Nelson and J. Solawetz, Roboflow team, "YOLOv5 is Here: State-of-the-Art Object Detection at 140 FPS," 2020. Available online: https://github.com/ultralytics/yolov5.

[19] C.-Y. Wang, A. Bochkovskiy, and H.-Y. Mark Liao, "Scaled-YOLOv4: Scaling cross stage partial network," *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Piscataway, NJ, 2021), pp. 13029–13038.

[20] V. E. Adam, "You only look twice: Rapid multi-scale object detection in 521 satellite imagery," arXiv:1805.09512 (2018).

[21] Z. Xu, X. Liu, and N. Ji, "Fog removal from color images using contrast limited adaptive histogram equalization," *Image and Signal Processing* (IEEE, Piscataway, NJ, 2009), pp. 1–5.

[22] S. G. Narasimhan and S. K. Nayar, "Contrast restoration of weather degraded images," IEEE Trans. Pattern Anal. Mach. Intell. **25**, 713–724 (2003).

[23] D. Miyazaki, D. Akiyama, M. Baba, R. Furukawa, S. Hiura, and N. Asada, "Polarization-based dehazing using two reference objects," *Proc. IEEE Int'l. Conf. on Computer Vision Workshops* (IEEE, Piscataway, NJ, 2013), pp. 852–859.

[24] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," IEEE Trans. Pattern Anal. Mach. Intell. **33**, 2341–2353 (2011).

[25] Q. Zhu, J. Mai, and L. Shao, "A fast single image haze removal algorithm using color attenuation prior," IEEE Trans. Image Process **24**, 3522–3533 (2015).

[26] D. Berman, T. Treibitz, and S. Avidan, "Non-local image dehazing," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2016), pp. 1674–1682.

[27] Z. Gu, Z. Zhan, Q. Yuan, and L. Yan, "Single remote sensing image dehazing using a prior-based dense attentive network," Remote Sens. **11**, 3008 (2019).

[28] http://weegee.vision.ucmerced.edu/datasets/landuse.html.

[29] https://gdo152.llnl.gov/cowc/.

[30] https://github.com/CAPTAIN-WHU/DOTA_devkit.

[31] K. Li, G. Wan, G. Cheng, L. Meng, and J. Han, "Object detection inoptical remote sensing images: a survey and a new benchmark," ISPRS J. Photogramm. Remote Sens. **159**, 296–307 (2020).

[32] MINT & Analysis, http://geimint.blogspot.com/2008/12/chinese-military-airfields.html.

[33] https://github.com/chaozhong2010/VHR-10_dataset_coco.

[34] https://github.com/CrazyStoneonRoad/TGRS-HRRSD-Dataset.