Infrared Small Target Detection Enhancement Using a Lightweight Convolutional Neural Network

Mridul Gupta, *Graduate Student Member, IEEE*, Jonathan Chan, Mitchell Krouss, *Member, IEEE*, Greg Furlich, Paul Martens, *Member, IEEE*, Moses W. Chan, Mary L. Comer, *Senior Member, IEEE*, and Edward J. Delp, *Life Fellow, IEEE*

***Abstract*—Detection of small, point targets is fundamental in applications, such as early warning systems, surveillance, astronomy, and microscopy. The presence of noise and clutter can make it challenging to detect small targets while minimizing false detections. This letter presents a method for infrared small target detection using convolutional neural networks (CNNs). The proposed method augments a conventional space-based detection processing chain with a lightweight neural network to predict the probability that a detection is a target. The proposed network is trained on 7 × 7 pixel windows using both the image sequence and the respective background-subtracted images. Results show that our method improves the probability of detection at low false detection rates.**

***Index Terms*—Background subtraction, convolutional neural networks (CNNs), deep learning, infrared small target detection.**

I. Introduction

DETECTION of small, point targets in infrared images is crucial in several applications, such as remote sensing and surveillance systems. These targets are modeled as point sources, since they are far from the observer. The targets are often buried in background clutter and noise, leading to a low signal-to-clutter ratio (SCR) [1]. Several point target detection approaches [2], [3], [4], [5], [6], [7], [8], [9] have been proposed over the years.

These detection methods produce a target mask where each pixel contains a score for whether the associated image pixel contains a target. We can classify these approaches into single-frame and multiframe approaches. We use the term "frame" to refer to an image captured at a particular time.

Single-frame methods use only spatial information to detect targets. Most of these methods rely on contrast between the target and background or the shape of the target. Relative local contrast measure (RLCM) [2] defines the score for a pixel as the difference between mean pixel values in the eight neighborhood windows of that pixel. Weighted strengthened local contrast measure (WSLCM) [3] improves upon RLCM by including matched filtering, background estimation, and dynamic weights based on target and background characteristics. Kim [10] proposed a single-frame method for target detection called modified-mean subtraction filter (M-MSF), which determines a target mask by subtracting a mean-filtered image from a matched-filtered image. He *et al.* [4] proposed a neural network-based method for target detection known as an infrared-image convolutional neural network (IRI-CNN). Their model is trained on 15 × 15 pixel windows from the image sequence, and the target mask is obtained by running the trained model at multiple locations in the image. These methods work well when the clutter is low but are prone to false detections in highly cluttered images.

Multiframe methods [5], [6], [7] use both spatial and temporal information. The temporal contrast filter (TCF) from [5] subtracts the minimum value at a pixel location in prior frames from the current frame. The target mask is then determined by computing the signal-to-noise ratio at each location. The temporal variance filter (TVF) [6] is another well-known multiframe method. It computes variance at a pixel location using just the prior frames and subtracts it from variance computed using both current and prior frames. These methods do not work well for slow-moving targets, because targets seem similar to the background. Matched filter-optical flow (MF-OF) proposed by Gupta *et al.* [7] rescores a target mask obtained using other methods, such as background subtraction.

It rescores these masks using weights computed on the basis of consistency of appearance and uniformity in the movement of a target. MF-OF does not work well for faster targets due to the limitations of the optical flow method.

Deep learning-based object detection methods have demonstrated good performance [11], [12], but these methods are not expected to work well for point targets [13]. Previously proposed deep learning-based methods for point target detection [8], [9] require a large amount of data for training. The required amount may not be readily available in practice. These models may also require resources not suitable for deployment on systems, such as satellites.

In this letter, we propose an approach using CNNs, which detects point targets with a low number of false detections. The computational requirements are minimal and amenable to execution on a CPU.

II. Proposed Approach

Fig. 1 shows a block diagram of our proposed MF-lightweight CNN (MF-LWCNN). The input is a temporal

Manuscript received 22 June 2022; revised 16 August 2022; accepted 24 August 2022. Date of publication 5 September 2022; date of current version 20 September 2022. This work was supported by Lockheed Martin Corporation, Sunnyvale, CA, USA. *(Corresponding author: Edward J. Delp.)*

Mridul Gupta, Mary L. Comer, and Edward J. Delp are with the Video and Image Processing Laboratory, School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: ace@ecn.purdue.edu).

Jonathan Chan, Mitchell Krouss, Greg Furlich, Paul Martens, and Moses W. Chan are with Lockheed Martin Space, Sunnyvale, CA 94089 USA.

Digital Object Identifier 10.1109/LGRS.2022.3203931

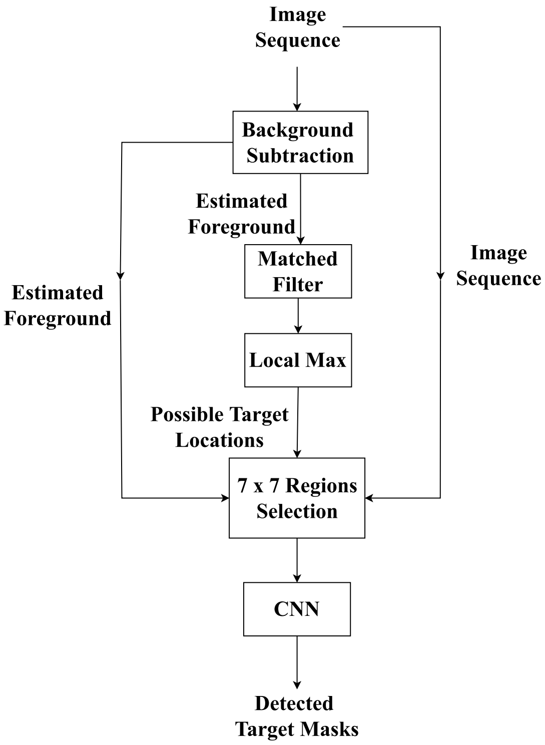


Fig. 1. Block diagram of the proposed method.

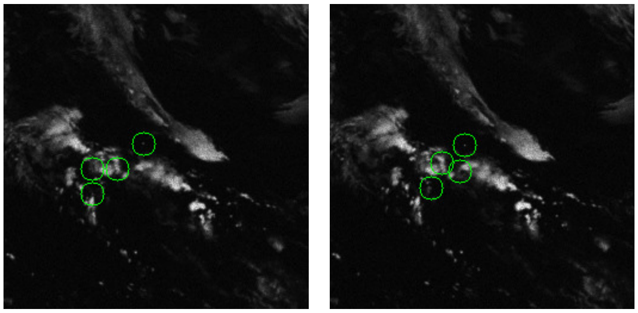


Fig. 2. Example of two images of an image sequence that are 30 frames apart. Targets are marked in circles.

sequence of frames, which we shall call the “image sequence.” These images contain small moving targets with low SCR [1]. Fig. 2 shows two images of an image sequence that are 30 frames apart.

First, we use background subtraction to estimate the foreground image for each frame. An MF is applied to the foreground images with the same shape as the targets to enhance the targets. We assume that no two targets are close to each other spatially. Then, local maxima in the match filtered image are considered possible target locations. A pixel is a local maximum if it is a peak in the 5 × 5 neighborhood around that pixel. We assume that the targets are small and completely contained in a 7 × 7 pixel window. The windows of size 7 × 7 pixel are constructed around the possible target locations in both original and foreground images. We use a CNN to estimate the probability that a target is present in these 7 × 7 pixel windows.

A. Background Estimation

The background estimation method by Wren *et al.* [14] uses the mean of p prior frames. Selection of p is based on the expected dynamics for background and targets. For our experiments, we empirically determined that $p = 10$ provides a good background estimate. To avoid the memory requirements for storing ten prior frames, we use an exponential moving average. The background estimate at time t is defined by

$$B_t = \alpha I_t + (1 - \alpha) B_{t-1} \tag{1}$$

where I_t is the image at time t, B_t is the estimated background at time t, and α is a parameter, which weighs current data against prior data. For our experiments, $\alpha = 0.1$ provides a good estimate of the background. We subtract the estimated background B_t from I_t to form the estimated foreground image.

B. MF Processing

An MF is used to detect possible locations of the targets. The CNN uses these locations in Fig. 1 to estimate the probabilities of a target being present at the locations determined by the MF. Although point targets are much smaller than a pixel, the target energy is spread out over multiple pixels and can be described by a symmetric bivariate Gaussian function. The target energy is assumed to be completely contained within a 7 × 7 pixel neighborhood. Then, possible target locations are expected to be local maxima in the match filtered image. We use 5 × 5 pixel windows for finding local maxima. The result is a mask where the nonzero locations are possible target locations, and the pixel value represents the score of a target being present at that location. In Section II-C, we describe the use of a CNN to enhance this mask to reduce the number of false detections.

C. Convolutional Neural Network

Data are extracted from the original and the foreground images using 7 × 7 pixel windows based on the possible target locations obtained from MF processing. The CNN is used to estimate the probability of a target being present in these 7 × 7 pixel windows.

Fig. 3 illustrates the architecture of our lightweight neural network model. By lightweight, we mean that the model does not require a dedicated graphics processing unit (GPU) for training and has a low computation time [15], [16]. Deep learning networks, such as U-Net [17] and You Only Look Once v4 (YOLOv4) [18], have 30 million and 12 million parameters, respectively. In contrast, the proposed neural network architecture has only 0.9 million parameters. The input to our CNN is a 2 × 7 × 7 tensor where the two channels contain data from the original image and the foreground image. The input is processed with three layers: one max-pooling layer (Maxpool) with window size 2 × 2 and two 2-D convolution layers. We use the notation Conv2D(m, k) in Fig. 3 to indicate that the 2-D convolutional layer has m filters of size $k \times k$. The input is processed with two 2-D convolutional layers with 32 filters of sizes 3 × 3 and 5 × 5, respectively (Fig. 3). The convolution layer

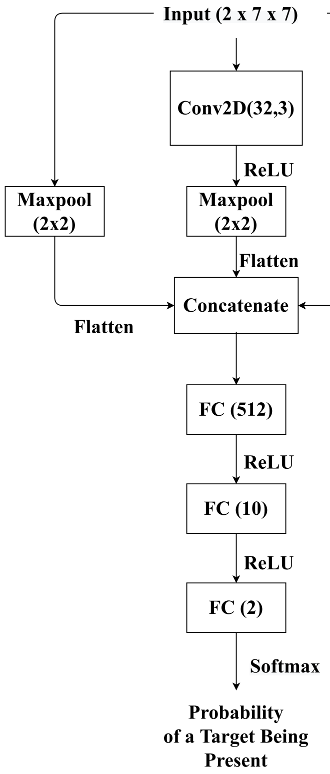


Fig. 3. Architecture of the CNN used in Fig. 1. Conv2D(m, k) is a 2-D convolution layer with m filters of size $k \times k$, and FC(n) is the fully connected layer with n neurons.

TABLE I
IMAGE SEQUENCES USED IN THE EXPERIMENTS

Seq.	Frames	Resolution	Average SCR [1]	Target Count
1	300	300 x 300	0.73	9
2	240	256 x 256	0.43	45
3	240	256 x 256	0.48	4
4	240	256 x 256	0.56	4
5	240	256 x 256	0.52	4
6	300	300 x 300	0.55	7
7	240	256 x 256	0.58	4
8	240	256 x 256	0.47	4
9	240	256 x 256	0.42	4

output is then max-pooled with window size 2 × 2. The outputs from the three max-pooling layers are flattened and concatenated to form a single 1-D vector. This vector is then processed by three fully connected (FC) layers with 512, 10, and 2 neurons, respectively. Rectified linear unit (ReLU) [19] is used as the activation function for all the layers except for the last layer, which uses a softmax activation [19] to estimate probabilities.

Nine image sequences were obtained from the Geostationary Operational Environmental Satellite (GOES)-16 weather satellite, and point targets with varying SCRs [1] and velocities were added to the sequences. SCR is defined in

$$\mathrm{SCR} = \frac{|\mu_t - \mu_b|}{\sigma_b} \tag{2}$$

where μ_t is the average pixel value in the target area, and μ_b and σ_b are the average and standard deviation of the pixel values in a local neighborhood of the target, respectively. We use a region of size 7 × 7 pixel for the target area and a region of size 47 × 47 pixel for the local neighborhood of the target as suggested in [1]. Table I shows information about the image sequences along with the average SCR of targets in all the frames. We use the first five image sequences for training the CNN and the other four sequences for evaluation. We constructed 14 900 7 × 7 pixel windows having targets and 14 900 no-target windows from the GOES training sequences. We used 70% of these windows for training and the rest for validation. The CNN was trained with binary cross-entropy loss, a batch size of 100, and a learning rate of 0.001 for the Adam optimizer [20].

During testing, the CNN estimates the probability that a 7 × 7 pixel window contains a target. A target mask is generated where each possible target location in the target mask is assigned a pixel value equal to the probability estimated by the CNN.

III. EXPERIMENTAL RESULTS

Recall that we obtained nine image sequences from the GOES-16 weather satellite. The CNN was trained on five image sequences, and the other four image sequences were used for evaluation. To compare our method with other existing approaches, we generated receiver operating characteristics (ROCs) for each test sequence. ROC shows the variation of probability of detection (P_d) with respect to change in probability of false detection (P_{fd}). P_d and P_{fd} are defined in

$$\begin{aligned} P_d &= \frac{\mathrm{TP}}{T} \\ P_{\mathrm{fd}} &= \frac{\mathrm{FP}}{N} \end{aligned} \tag{3}$$

where TP is the number of true detections, T is the total number of true targets, FP is the number of false detections, and N is the total number of pixels in an image. By true detections, we mean the number of pixels containing a true target, which are also predicted as containing a target by the CNN. Similarly, false detections is the number of pixels predicted as containing a target by the CNN but do not contain a target.

Fig. 4 shows the target masks for frames from image sequences 6 and 9. The first two images are frames from two test image sequences, and the last two images are their respective target masks generated using the proposed method. Ground truth targets in images are marked with circles, and squares mark the detected targets in target masks. Fig. 5 shows the ROC for our approach and six other existing approaches for every test sequence. P_d and P_{fd} are averaged over all the frames in an image sequence. The x-axis uses a log scale for easier interpretation of the ROCs.

The methods we used for comparison include a basic 7 × 7 size MF on foreground images. This method represents conventional processing. Another comparison method is IRI-CNN [4], which uses a CNN on image sequences with windows of size 15 × 15 pixel. In images with high clutter and noise, spatial information from the image sequence alone does not have enough information to distinguish a target from

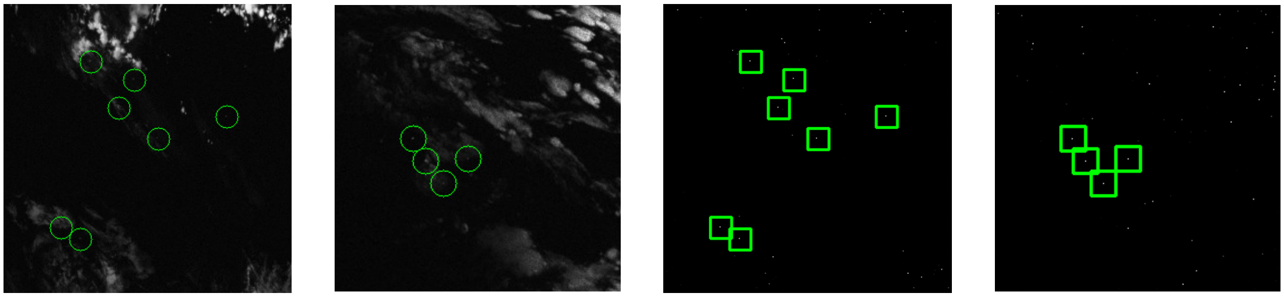
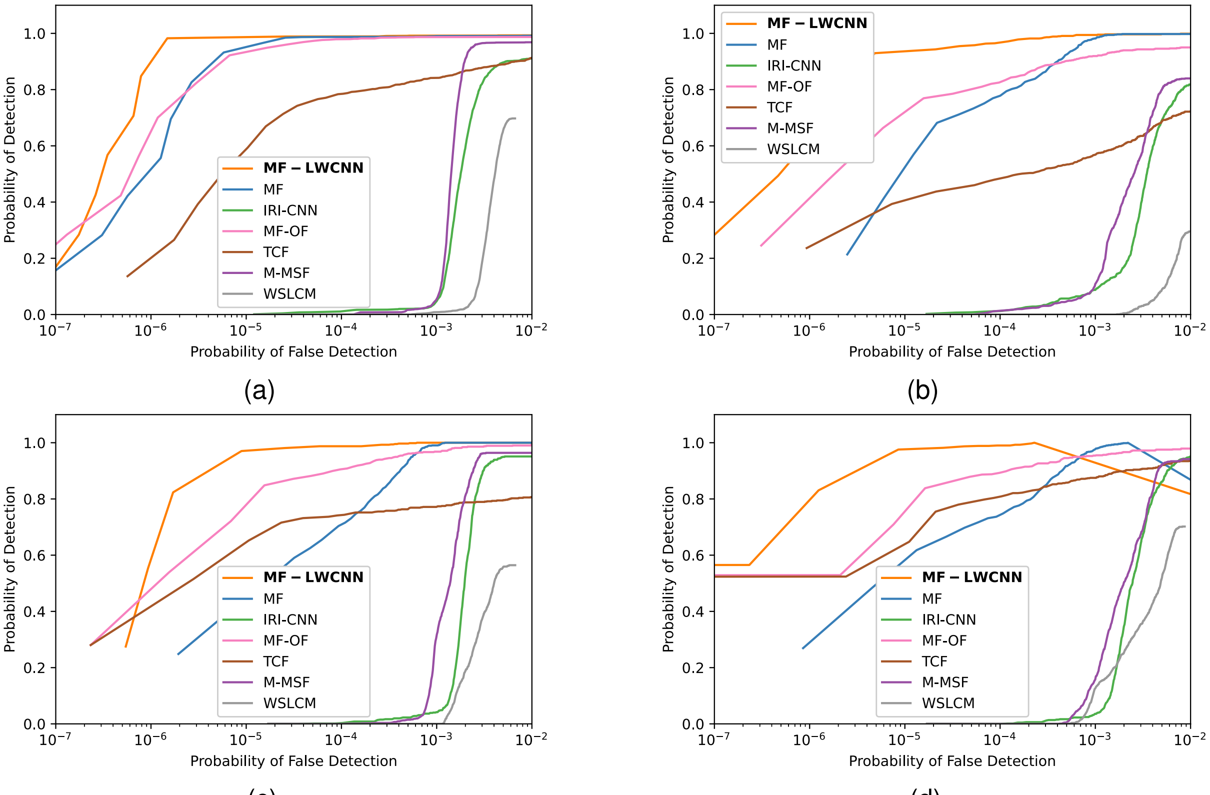


Fig. 4. Two test images are on the left, and their respective detected target masks are on the right. Ground-truth targets are marked with circles. Detected targets are marked with squares.



(a)

(b)

(c)

(d)

Fig. 5. ROC of the proposed method and six other methods for each test sequence. (a) Sequence 6. (b) Sequence 7. (c) Sequence 8. (d) Sequence 9.

a false detection; hence, the method has a high number of false detections. We use the MF-OF [7] to rescale the target mask from MF using weights based on uniformity of motion and consistency of appearance of a target in prior frames. MF-OF leverages optical flow, which becomes problematic when targets move through heavy clutter. Optical flow also has the limitation of not being able to track a target moving more than one pixel between two consecutive frames. The TCF [5] determines a target mask by subtracting the current frame and the minimum pixel value at that location from the previous frame. Since the targets are buried in clutter and noise, the minimum pixel value from previous frames is very close to the current frame's pixel value, which results in a nearly zero value in the target mask. M-MSF [10] detects a target by subtracting a mean-filtered image from a matched-filtered image. While it detects most of the targets, it has a higher number of false detections, because the MF enhances clutter along with the targets. The last method we use for comparison is the WSLCM [3] that uses an MF and background estimation to suppress the background. The background estimation using spatial information does not work well in highly cluttered images. Due to a bad background estimate, WSLCM cannot detect all the targets in any image sequence.

Table II shows the computation time required by these methods per frame in seconds. Our method can process five frames of size 300 × 300 in a second. While the proposed method (MF-LWCNN), MF, and M-MSF have similar computation times, MF-LWCNN has fewer false detections. For example, at a 95% detection rate, the proposed method raises a significantly lower number of false detections, as shown in

TABLE II

COMPUTATION TIME PER FRAME (SECONDS) FOR EACH METHOD

Method	**Seq. 6**	**Seq. 7, 8, 9**
Frame Size	300 × 300	256 × 256
MF-LWCNN	0.20	0.14
MF	0.18	0.12
IRI-CNN [4]	0.10	0.08
MF-OF [7]	1.88	1.27
TCF [5]	0.63	0.46
M-MSF [10]	0.21	0.16
WSLCM [3]	396	280

TABLE III

AVERAGE NUMBER OF FALSE DETECTIONS PER FRAME AT $P_d = 0.95$ FOR ROC IN FIG. 5

Method	**Seq. 6**	**Seq. 7**	**Seq. 8**	**Seq. 9**
MF-LWCNN	**0.13**	**2.44**	**0.58**	**0.56**
MF	1.36	42.75	37.56	46.52
IRI-CNN [4]	2282.40	-	339.71	646.33
MF-OF [7]	2.49	518.34	26.37	48.77
TCF [5]	-	-	-	-
M-MSF [10]	209.44	-	184.17	-
WSLCM [3]	-	-	-	-

Table III. If a method cannot detect 95% of the targets, its corresponding entry in Table III is “-.” Note that all other methods have higher false detections.

All the experiments were conducted with Python on an Intel Core i9 $-$ 9900X 3.50-GHz CPU with 128-GB RAM.

IV. CONCLUSION

In this letter, we presented a method (MF-LWCNN), which improves the detection of small, point targets by augmenting a conventional processing chain with a lightweight CNN. Our method performs well compared with the existing approaches and works on a CPU. MF-LWCNN can also be used with foreground images obtained using other background subtraction methods. Our proposed method is trained using spatial information from the original image sequence and foreground images. Future work will explore training the CNN on temporal information while keeping the computation requirements low. We have not tested our proposed method on targets that are spatially close to each other. Future work will also consider closely spaced targets.

ACKNOWLEDGMENT

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied of Lockheed Martin Corporation, Sunnyvale, CA, USA. The authors would like to thank the National Oceanic and Atmospheric Administration (NOAA), Washington, DC, USA, for making the Geostationary Operational Environmental Satellite (GOES) imagery used in this study available to the public. Address all correspondence to Edward J. Delp, ace@ecn.purdue.edu.

REFERENCES

[1] L. Zhang and Z. Peng, “Infrared small target detection based on partial sum of the tensor nuclear norm,” *Remote Sens.*, vol. 11, no. 4, p. 382, Jan. 2019.

[2] J. Han, K. Liang, B. Zhou, X. Zhu, J. Zhao, and L. Zhao, “Infrared small target detection utilizing the multiscale relative local contrast measure,” *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 4, pp. 612–616, Apr. 2018.

[3] J. Han *et al.*, “Infrared small target detection based on the weighted strengthened local contrast measure,” *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 9, pp. 1670–1674, Sep. 2021.

[4] J.-K. He, D.-Z. Yang, C.-B. An, and J. Li, “Infrared dim target detection technology based on IRI-CNN,” in *Proc. SPIE*, Hong Kong, vol. 12166, Feb. 2022, Art. no. 121665E.

[5] S. Kim, S.-G. Sun, and K.-T. Kim, “Highly efficient supersonic small infrared target detection using temporal contrast filter,” *Electron. Lett.*, vol. 50, no. 2, pp. 81–83, 2014.

[6] X. Sun, T. Zhang, and M. Li, “Moving point target detection using temporal variance filter in IR imagery,” in *Proc. SPIE*, Wuhan, China, vol. 6786, Nov. 2007, Art. no. 67861Z.

[7] M. Gupta *et al.*, “Small target detection using optical flow,” in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, USA, Mar. 2021, pp. 1–9.

[8] Q. Hou, Z. Wang, F. Tan, Y. Zhao, H. Zheng, and W. Zhang, “RISTDNet: Robust infrared small target detection network,” *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[9] Q. Hou, L. Zhang, F. Tan, Y. Xi, H. Zheng, and N. Li, “ISTDU-Net: Infrared small-target detection U-net,” *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[10] S. Kim, “Double layered-background removal filter for detecting small infrared targets in heterogenous backgrounds,” *J. Infr., Millim., Terahertz Waves*, vol. 32, no. 1, pp. 79–101, Jan. 2011.

[11] G. Cheng, X. Yuan, X. Yao, K. Yan, Q. Zeng, and J. Han, “Towards large-scale small object detection: Survey and benchmarks,” 2022, *arXiv:2207.14096*.

[12] G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, “When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative CNNs,” *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 5, pp. 2811–2821, May 2018.

[13] J. Du, H. Lu, M. Hu, L. Zhang, and X. Shen, “CNN-based infrared dim small target detection algorithm using target-oriented shallow-deep features and effective small anchor,” *IET Image Process.*, vol. 15, no. 1, pp. 1–15, Jan. 2021.

[14] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, “Pfinder: Real-time tracking of the human body,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 780–785, Jul. 1997.

[15] A. K. Sharma and H. Foroosh, “Slim-CNN: A light-weight CNN for face attribute prediction,” in *Proc. 15th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, Buenos Aires, Argentina, Nov. 2020, pp. 329–335.

[16] Y. Zhou, S. Chen, Y. Wang, and W. Huan, “Review of research on lightweight convolutional neural networks,” in *Proc. IEEE 5th Inf. Technol. Mechatronics Eng. Conf. (ITOEC)*, Chongqing, China, Jun. 2020, pp. 1713–1720.

[17] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, Oct. 2015, pp. 234–241.

[18] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, “YOLOv4: Optimal speed and accuracy of object detection,” 2020, *arXiv:2004.10934*.

[19] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation functions: Comparison of trends in practice and research for deep learning,” 2018, *arXiv:1811.03378*.

[20] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, *arXiv:1412.6980*.