Hindawi
Security and Communication Networks
Volume 2023, Article ID 9839583, 1 page
https://doi.org/10.1155/2023/9839583

Retraction

Retracted: Research on Lightweight Infrared Pedestrian Detection Model Algorithm for Embedded Platform

Security and Communication Networks

Received 13 September 2023; Accepted 13 September 2023; Published 14 September 2023

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope

(2) Discrepancies in the description of the research reported

(3) Discrepancies between the availability of data and the research described

(4) Inappropriate citations

(5) Incoherent, meaningless and/or irrelevant content included in the article

(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] Z. Wu, X. Wang, and C. Chen, "Research on Lightweight Infrared Pedestrian Detection Model Algorithm for Embedded Platform," *Security and Communication Networks*, vol. 2021, Article ID 1549772, 7 pages, 2021.

Hindawi
Security and Communication Networks
Volume 2021, Article ID 1549772, 7 pages
https://doi.org/10.1155/2021/1549772

Research Article

Research on Lightweight Infrared Pedestrian Detection Model Algorithm for Embedded Platform

Zhaoli Wu,[1,2,3,4] **Xin Wang**,[2,3] **and Chao Chen**[2,3]

[1]*China University of Mining and Technology School of Computer Science and Technology, Xuzhou 221116, China*
[2]*Jiangsu Vocational Institute of Architectural Technology School of Information and Electronics Engineering, Xuzhou 221116, China*
[3]*Xuzhou Intelligent Machine Vision Engineering and Technology Center, Xuzhou 221116, China*
[4]*JiangSu Collaborative Innovation Center for Building Energy Saving and Construction Technology, Xuzhou 221116, China*

Correspondence should be addressed to Zhaoli Wu; lb20170009@cumt.edu.cn

Received 11 October 2021; Revised 1 November 2021; Accepted 5 November 2021; Published 30 November 2021

Academic Editor: Jian Su

Due to the limitation of energy consumption and power consumption, the embedded platform cannot meet the real-time requirements of the far-infrared image pedestrian detection algorithm. To solve this problem, this paper proposes a new real-time infrared pedestrian detection algorithm (RepVGG-YOLOv4, Rep-YOLO), which uses RepVGG to reconstruct the YOLOv4 backbone network, reduces the amount of model parameters and calculations, and improves the speed of target detection; using space spatial pyramid pooling (SPP) obtains different receptive field information to improve the accuracy of model detection; using the channel pruning compression method reduces redundant parameters, model size, and computational complexity. The experimental results show that compared with the YOLOv4 target detection algorithm, the Rep-YOLO algorithm reduces the model volume by 90%, the floating-point calculation is reduced by 93.4%, the reasoning speed is increased by 4 times, and the model detection accuracy after compression reaches 93.25%.

1. Introduction

Target detection [1] is an important research direction in the field of computer vision. With the rapid development of deep learning, new target detection algorithms continue to emerge in the visible light environment, but related algorithms rely heavily on sufficient lighting conditions and cannot meet the target detection requirements in under-lighted scenes. Infrared thermal imaging refers to the use of the reflection of infrared light and the thermal radiation signal of the target to convert it into an image that human vision can accept and perceive. It can image the surrounding environment under conditions such as darkness and strong light, can cover most of the lack of light, and can cover most scenes with insufficient light to achieve all-weather and all-time detection.

At this stage, infrared pedestrian detection algorithms can be roughly divided into two types. One is based on the artificially designed pedestrian template ratio. The target contour is extracted by the artificially designed target contour extraction method and compared with the template. The main detection methods are as follows: scale-invariant feature detection method, Haar feature algorithm, gradient histogram algorithm, etc. Traditional infrared pedestrian detection algorithms have high requirements for designers and weak generalization capabilities. The other is based on deep learning and using convolution operations to achieve target detection algorithms that autonomously extract and combine features of targets in the image, and their features are significantly stronger than those designed by humans. The target detection algorithm based on deep learning [2] is mainly divided into one-stage and two-stage. One-state representative networks are SSD [3] series and YOLO [4] series, which use a one-step framework for global regression and classification; two-state representative networks are R-CNN [5] series, which generate suggested regions and then recommend classification and regression of regions.

However, the target detection algorithm based on deep learning has a huge amount of computation, and embedded devices cannot meet its computing power requirements. Moreover, because of the low power consumption and low energy requirements of embedded platforms, this paper selects the mainstream target recognition algorithm YOLOv4 for lightweight improvement and integration. Integrate the SPP network to optimize the detection accuracy of the model, and use the RepVGG [6] network combined with the channel pruning limit compression method to compress, so as to obtain a target detection model suitable for deployment on an embedded platform with limited resources.

2. YOLOv4 Target Detection Algorithm

The Rep-YOLO network continues to use the idea of YOLO target detection. The entire image is used as the input of the entire network without the need to generate suggested regions. The regression idea is used in the output layer to obtain the position and category of the bounding box, and then it is suppressed by nonmaximum value. The algorithm removes the redundant bounding box and obtains the final prediction result. The whole process is that the detection network directly performs end-to-end prediction, and the detection speed is relatively high.

The YOLOv4 algorithm optimizes the YOLOv3 model from the perspective of data preprocessing, backbone network, training mode, activation function, etc., so that the detection model achieves a good balance between detection speed and detection accuracy. The YOLOv4 backbone network CSPDarkNet [7] combines the advantages of CSPNet (cross stage partial network). The CSP module is added to the backbone network DarkNet53 of YOLOv3. The shallow feature mapping is divided into two parts and then merged through a cross-layer structure. Quantify the network while maintaining detection accuracy, reducing computing bottlenecks, and reducing memory costs. In addition, YOLOv4 combines the advantages of PANet [8] to spread the semantic information of high-level features to the low-level network and merge it with the high-resolution information of the shallow features to improve the detection effect of small target objects; then, the low-level information is propagated to the high-level network. The feature map can obtain richer semantic information, and finally use the feature map of different layers to predict; YOLOv4 optimizes the loss function, adopts the CIoU-Loss [9] loss function, and considers the intersection ratio, center point distance, and length and width. Comparing the various losses makes the regression speed and accuracy of the prediction box optimal; YOLOv4 optimizes the nonmaximum algorithm, fully considering the intersection ratio and distance information of the coincident bounding boxes, and significantly improves the detection accuracy of overlapping targets.

3. Rep-YOLO Target Detection Algorithm

The target detection algorithm Rep-YOLO proposed in this paper first reconstructs the YOLOv4 backbone network based on the RepVGG network; secondly, it integrates the pyramid pooling model [10] to obtain feature information of different scales, then compresses the target detection model through the channel pruning limit compression method, and finally uses fine-tuning. The method restores the accuracy and obtains a lightweight detection model with high precision, low volume, and fast detection speed.

3.1. Reconstruction of Recognition Network Based on RepVGG-B0 Convolution Module. Ding Xiaohan et al. proposed the RepVGG network in 2021 and applied the characteristics of the ResNet network to the VGG network, that is, adding the identity residual branch and the 1×1 convolution branch to the block module of the VGG network. At the same time, the author adopts the method of structure reparameterization to decouple the training process from the inference process and uses different network structures and model parameters. The combined residual structure is selected in the training phase to improve the detection accuracy, and the OP fusion strategy is used to integrate all networks in the inference phase. The layer is converted into a 3×3 convolutional layer to facilitate model deployment and acceletion. Figure 1 is the structure diagram of RepVGG network.

The BN (batch normalization) layer in the neural network can quickly converge and accelerate the network, effectively solving the gradient disappearance and gradient explosion, but the BN layer will occupy more memory and video memory in the forward reasoning process, increasing the time-consuming model reasoning. The convolutional layer and the BN layer in the residual module are merged by equation (1), and the formula is derived as follows:

Convolutional layer calculation formula:

$$x_1 = \omega \times x + b. \tag{1}$$

BN layer calculation formula:

$$x_2 = \gamma \times \frac{x_1 - u}{\sqrt{\delta^2 + \varepsilon}} + \beta. \tag{2}$$

Here, γ and β are the parameters that need to be learned, u is the sample mean, δ is the sample variance, and ε is a small number to prevent the denominator from being zero.

Incorporating formula (1) into formula (2), the convolutional layer and the BN layer are combined to obtain the following equation:

$$x_2 = \gamma \times \frac{\omega \times x + b - u}{\sqrt{\delta^2 + \varepsilon}} + \beta. \tag{3}$$

Formula (3) can be sorted to get the following equation:

$$x^2 = \frac{\gamma \times \omega}{\sqrt{\delta^2 + \varepsilon}} \times x + \beta + \gamma \times \frac{b - u}{\sqrt{\delta^2 + \varepsilon}}. \tag{4}$$

$\gamma \times \omega / \sqrt{\delta^2 + \varepsilon} = \omega'$, $\beta + \gamma \times b - u / \sqrt{\delta^2 + \varepsilon} = b'$. Available equation

$$x^2 = \omega' \times x + b'. \tag{5}$$

In the RepVGG network structure, there are two branches: 1×1 convolution module and identity module.

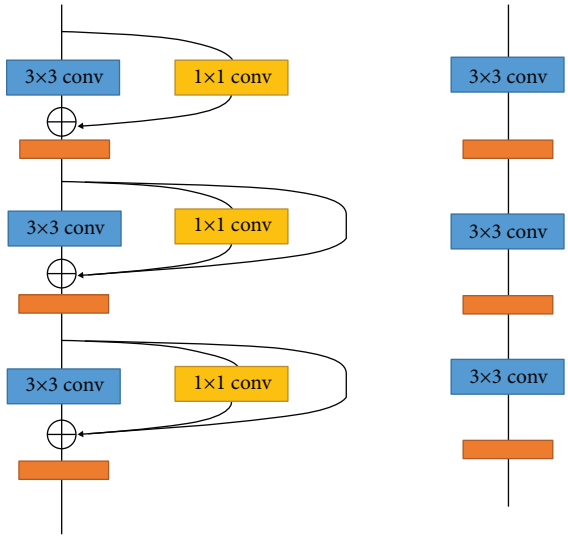

Figure 1: The structure diagram of RepVGG network. (a) Training model. (b) Reasoning model.

For the 1×1 convolution module, it can be equivalent to 1×1 convolution padding as 3×3 convolution, where all positions are 0, except for the convolution kernel loyalty position; for the identity module, it can be equivalent to setting a weight average. It is a $1\ 3 \times 3$ convolution kernel; after multiplying the input feature map, the value before and after the identity remains unchanged. According to the addition characteristics of convolution, the convolution kernel can be added when the shape is the same, so the three convolution branches can be merged. The fusion process is shown in Figure 2.

The main reason why RepVGG uses 3×3 convolution is that modern computing libraries (NVIDIA, cuDNN, etc.) are highly optimized. Table 1 shows the theoretical FLOPs, actual running time, and computational density tested using cuDNN7.5.0 on 1080tiGPU. The results show that the theoretical calculation density of 3×3 convolution is about 4 times that of other models, which means that FLOPs cannot replace the actual speed in different architectures. The difference between FLOPs [11] and speed can be attributed to two important factors: memory access cost and parallelism. Under the same FLOPs, a model with a high degree of parallelism is much faster than a model with a low degree of parallelism, and a simple reasoning structure can avoid multibranch fragmentary calculations. The multibranch topology imposes constraints on the model architecture and limits the application of model pruning. However, the simple architecture allows the convolutional layer to be configured according to actual needs to obtain a better trade-off between model efficiency and performance.

3.2. Model Channel Sparse Training. Model channel sparse training can distinguish important channels from unimportant channels. In order to facilitate channel pruning, each channel of the first convolutional layer is assigned a scale factor, where the absolute value of the scale factor indicates the importance of the channel. Part of the scale factor gradually approaches 0 after sparse training, and the channel and connection are cut off by setting the threshold to achieve the purpose of reducing the amount of calculation and the model, as shown in Figure 3.

3.3. Target Detection Model Pruning and Fine-Tuning. This paper defines a global variable as the threshold of the entire scale factor to control the pruning rate of the convolutional layer channel. In addition, we also introduce a local safety threshold to prevent excessive pruning of the convolutional layer channel to maintain the integrity of the network. Some special layers (routing layer and shortcut layer) in YOLOv4 need to be handled carefully. Because the maximum pooling layer and the upsampling layer have nothing to do with the number of channels, they are not operated on. After channel pruning, some accuracy may be reduced, so it needs to be fine-tuned to restore accuracy. The model compression process is best to adopt incremental pruning strategy, because excessive pruning will lead to catastrophic degradation of model accuracy, and the original accuracy cannot be restored, as shown in Figure 4.

4. Experiment and Analysis

4.1. Lab Environment. This experiment environment is Ubuntu16.04 operating system, PyTorch deep learning framework; workstation configuration is NVIDIA GTX 1080ti graphics card ×2, Intel Core i7 processor; embedded platform is NVIDIA Jetson TX2 mobile development board.

4.2. Experimental Dataset. This paper uses FLIR's open source infrared dataset [12] and infrared CVC infrared pedestrian dataset to test the performance of the proposed real-time infrared detection network. The data set of this experiment consists of three data sets: FLIR, CVC-09, and CVC-14, and the training, validation, and test sets are redivided in the ratio of 7:2:1. The dataset is shown in Figure 5.

4.3. Model Comparison Experiment. The input size of all experimental models in this paper is 608×608. Tables 1 and 2 compare the floating-point calculations (BFLOPs), model volume, prediction accuracy (mAP), and reasoning time (inference time).

Experiments show that after using the RepVGG network to reconstruct the YOLOv4 backbone network, the amount of calculation is reduced to 42.65% of the traditional YOLOv4 model, the model volume is reduced to 48.97%, the speed is increased by 1.87 times on the GTX 1080ti and 1.72 times on the Jetson TX2, and the accuracy is slightly improved. Compared with the traditional YOLOv4 target detection model, Rep-YOLO has fewer parameters and higher detection accuracy. The model prediction efficiency is also significantly improved, and the model volume and computational complexity are significantly reduced.

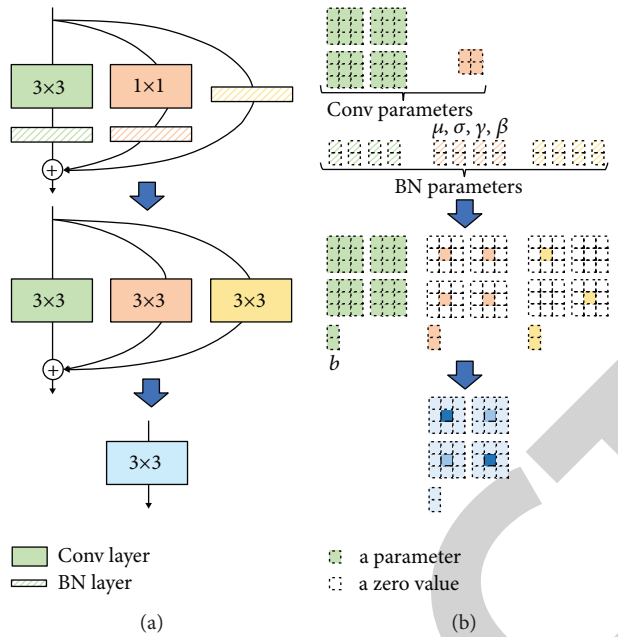

Figure 2: RepVGG branch merge process. (a) Perspective of structure and (b) perspective of parameter.

Table 1: Model comparison.

Model name	BFLOPs	Model volume	mAP (AP)
YOLOv4 (baseline)	128.46	245	94.25
Rep-YOLO	54.79	120	95.34
YOLOv4-tiny	6.38	25.6	84.59
YOLOv3	94.67	125	89.04

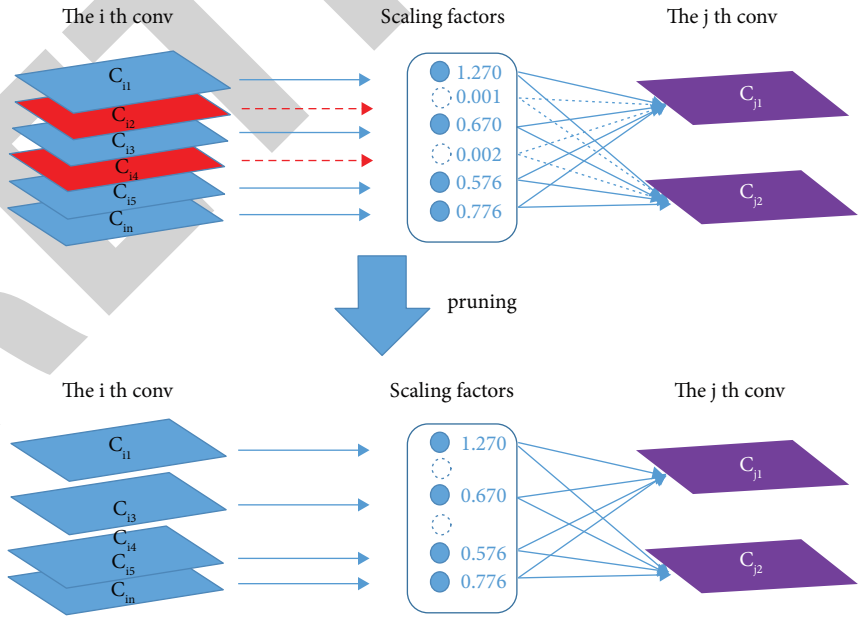

Figure 3: Channel sparse training and pruning.

4.4. Model Thinning Experiment. Different sparse penalty terms α are set in the experiment. After sparse training, the average detection accuracy mAP is shown in Figure 6.

It can be seen from Figure 6 that when $\alpha = 0.0001$, the detection accuracy of the Rep-YOLO target detection model reaches the best. At this time, the scale factors [13] of

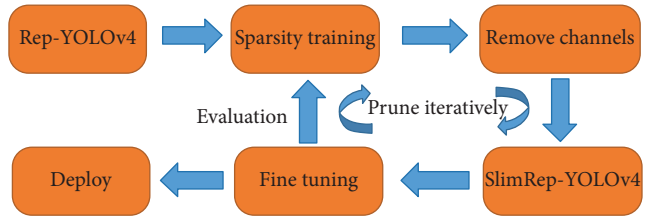

Figure 4: Sparsity training and channel pruning.

Figure 5: Example picture of data.

Table 2: Detection efficiency comparison.

Model name	Inference time	
	GTX 1080ti	Jetson TX2
YOLOv4 (baseline)	89.54	540.34
Rep-YOLO	47.65	313.62
YOLOv4-tiny	15.68	86.47
YOLOv3	85.27	560.39

different channels in the training process of 100 sparsification [14] are all close to 0, as shown in Figure 7.

4.5. Model Cutting Comparison Experiment. In order to further improve the performance of the detection model, this paper cuts the Rep-YOLO model, and the cut rate is set to 0.5, 0.7, and 0.9. The performance of different cut rate models is shown in Table 3.

It can be obtained from Table 3 that with the increase of the cropping rate, the calculation amount of the floating point [15], the model volume, and the prediction time are constantly decreasing. Real-time detection can be achieved

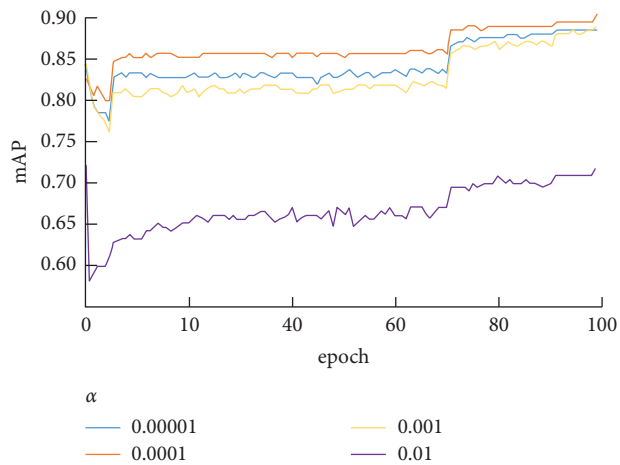

Figure 6: Variation curves of mAP with different penalty terms.

on the GTX 1080ti and the floating point of the Rep-YOLO-0.7 model. The amount of calculation is reduced to 13.17% of the YOLOv4 model, and the volume is reduced to 5.2%. The

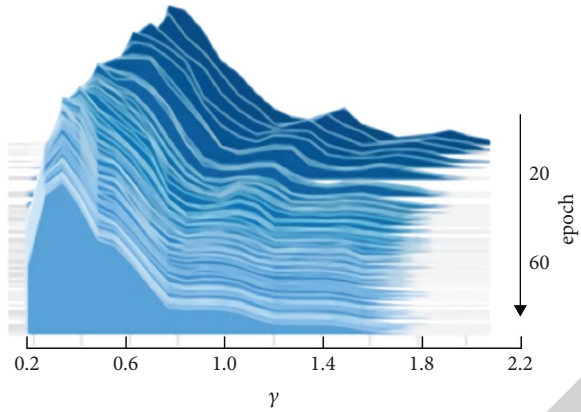

Figure 7: The distribution of scaling factor in sparse process.

Table 3: Performance comparison of different pruned models.

Model name	BFLOPs	Model volume	mAP (AP)	Inference time (ms)	
				GTX 1080ti	TX2
YOLOv4	128.46	245	94.25	89.54	540.34
Rep-YOLO-0.5	36.86	56.93	93.57	34.20	290.51
Rep-YOLO-0.7	16.92	12.83	91.04	30.42	203.56
Rep-YOLO-0.9	8.33	6.72	53.69	24.91	184.92

Figure 8: Object detection results of Rep-YOLO-0.5.

detection speed is increased by 2.94 times and 2.65 times on the GTX 1080ti platform and Jetson TX2 platform, respectively. Experiments have proved that the model channel clipping method can achieve a significant reduction in model volume and floating-point calculations under the premise of ensuring high detection accuracy and significantly improve the detection speed. The test result of the Rep-YOLO-0.7 model is shown in Figure 8.

5. Conclusion

Based on YOLOv4, this paper proposes a real-time infrared pedestrian detection algorithm suitable for embedded platforms, using structural parameter reconstruction ideas to reconstruct the YOLOv4 backbone network, which significantly reduces the amount and volume of model parameters and improves the network while reducing the

amount of model floating-point calculations. This paper proposes a real-time infrared pedestrian detection algorithm based on YOLOv4 for embedded platforms and reconstructs the YOLOv4 backbone network by using the idea of structural parameter reconstruction to significantly reduce the number and size of model parameters and improve the detection accuracy and speed of the network model while reducing the amount of model floating-point computation; using the convolution channel pruning limit compression method, while maintaining the detection accuracy, the model volume and parameter amount are further effectively compressed, the memory usage during the model inference process is reduced, and the operation of the model is greatly improved and efficient. However, this article does not consider the characteristics of different hardware platforms. Later, different models can be designed according to the characteristics of different platforms to improve the generalization ability of the detection model.

Data Availability

The simulation experiment data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Jiangsu Collaborative Innovation Center for Building Energy Saving and Construction Technology (Grant no. SJXTY1603) and National Educational Technology Research Project of Central Audio Visual Education Center (Grant no. 186130061); Xuzhou city will promote the special Key Research and Development Plan for Scientific and Technological Innovation (Industrial Key Technology Research and Development) Project "R&D and application of water resources cloud control platform at river basin level "(Grant no. KC21108), Xuzhou city will promote the special policy guidance plan for scientific and technological innovation (industry-university-research cooperation)" Big data-based multiobjective coordinated and balanced allocation of large-region water resources "(Grant no. KC21335), and the special project "Research on the training mode of artificial intelligence teachers (Higher Vocational Colleges) of smart campus of modern educational technology research in Jiangsu Province" (Grant no. 2020-R-84360).

References

[1] R. Lu, X. Yang, W. Li, J. Fan, D. Li, and X. Jing, "Robust infrared small target detection via multidirectional derivative-based weighted contrast measure," *IEEE Geoscience and Remote Sensing Letters*, 2020.

[2] C. Gao, D. Meng, Y. Yang, Y. Wang, X. Zhou, and A. G. Hauptmann, "Infrared patch-image model for small target detection in a single image," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 4996–5009, 2013.

[3] Z. Gao, J. Dai, and C. Xie, "Dim and small target detection based on feature mapping neural networks," *Journal of Visual Communication and Image Representation*, vol. 62, pp. 206–216, 2019.

[4] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, Las Vegas, NV, USA, June 2016.

[5] Z. Zhu, M. Xu, S. Bai, T. Huang, and X. Bai, "Asymmetric non-local neural networks for semantic segmentation," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 593–602, Seoul, Korea, November 2019.

[6] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "Deepflow: large displacement optical flow with deep matching," in *Proceedings of the 2013 IEEE International Conference on Computer Vision*, pp. 1385–1392, Sydney, Australia, December 2013.

[7] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "Hmdb: a large video database for human motion recognition," in *Proceedings of the 2011 International Conference on Computer Vision*, pp. 2556–2563, Barcelona, Spain, November 2011.

[8] S. Saxena and J. Verbeek, "Heterogeneous face recognition with CNNs," in *Proceedings of the Computer Vision – ECCV 2016 Workshops*, pp. 483–491, Springer International ′Publishing, Amsterdam, Netherlands, October 2016.

[9] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobilenetV2: inverted residuals and linear bottlenecks," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, Salt Lake City, UT, USA, June 2018.

[10] Y. Gao, J. Ma, and A. L. Yuille, "Semi-supervised sparse representation based classification for face recognition with insufficient labeled samples," *IEEE Transactions on Image Processing*, vol. 26, no. 5, pp. 2545–2560, 2017.

[11] G. Zhao, X. Huang, M. Taini, S. Z. Li, and M. Pietikäinen, "Facial expression recognition from near-infrared videos," *Image and Vision Computing*, vol. 29, no. 9, pp. 607–619, 2011.

[12] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets, neural information processing systems," in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pp. 2672–2680, Montreal Canada, December 2014.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[14] D.-P. Fan, Z. Lin, Z. Zhang, M. Zhu, and M.-M. Cheng, "Rethinking RGB-D salient object detection: models, data sets, and large-scale benchmarks," *IEEE Transaction Neural Network Learning Systsem*, vol. 32, no. 5, pp. 2075–2089, 2021.

[15] F. Perazzi, P. Krahenbuhl, Y. Pritch, and A. Hornung, "Saliency filters: contrast based filtering for salient region detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 733–740, Providence, RI, USA, June 2012.