Efficient Parcel Damage Detection via Faster R-CNN: A Deep Learning Approach for Logistical Parcels' Automated Inspection

Zhi Chen[1,2], Cuifeng Du[5], Quanlong Guan[1,2,4], Yuyu Zhou[2,7], Vichen Hoo[1,7], Xiujie Huang[1,4], Zhefu Li[3](✉), Shuanghuan Lv[3], Xiaofeng Wu[6](✉), and Xiaotian Zhuang[8]

[1] College of Information Science and Technology, Jinan University, Guangzhou, China
[2] Guangdong Institute of Smart Education, Jinan University, Guangzhou, China
[3] Network and Education Technology Center, Jinan University, Guangzhou, China
[4] Guangdong Key Laboratory of Data Security and Privacy Preserving, Guangzhou, China
[5] Cetc Potevio Science&Technology Co., Ltd., Guangzhou, China
[6] Guangzhou Polytechnic of Sports, Guangzhou, China
[7] Guangdong-Macao Advanced Intelligent Computing Joint Laboratory, Guangzhou, China
[8] Beijing JD Zhenshi Information Technology Co., Ltd., Beijing, China

Abstract. Parcel damage detection poses a significant challenge in warehouse automation and conveyor belt transportation, with a particular focus on leveraging the potential of Internet of Things (IoT) technologies. With this purpose in mind, we introduce RPA R-CNN, an improved deep learning-based object detection method that seamlessly integrates IoT capabilities. Our proposed approach enhances the feature extraction network of the Faster R-CNN model and utilizes advanced post-processing techniques to further refine the extracted features. By employing a meticulously curated dataset specifically designed for parcel detection, we train our model and conduct rigorous experimental verification. Encouragingly, the outcomes convincingly demonstrate that the RPA R-CNN method surpasses the performance of the original model in various metrics in domestic or cross-border logistics parcel detection. These promising results not only offer a practical solution to the challenging problem of identifying parcel damage but also underscore the immense potential of integrating IoT principles to optimize automation and enhance operational efficiency within the domestic or cross-border logistics industry.

Keywords: Faster R-CNN · Parcel Damage Detection · IoT

A. Zaslavsky et al. (Eds.): MobiQuitous 2023, LNICST 594, pp. 268–279, 2024.
https://doi.org/10.1007/978-3-031-63992-0_18

1 Introduction

With the rapid growth of e-commerce [4] and the domestic or cross-border logistics industry, there are significant concerns regarding the detection and localization of package damage in domestic or cross-border logistics environments, including scenarios such as automated warehouse transport. The development of efficient methods for accurately identifying and locating damaged parcels is crucial for ensuring the safety of goods and achieving high delivery accuracy. Traditional approaches relying on manual inspection and human judgment suffer from limitations such as low efficiency, lengthy processing times, and potential errors. Meanwhile, there is a growing number of applications applying deep learning techniques to object detection [3], including video object detection [8], UAV location [5], crack detection [27], and so on. Thus, our objective is to propose an efficient object detection method for parcel damage detection through the integration of deep learning and IoT technologies.

While Faster R-CNN [19] has demonstrated remarkable performance in general object detection, it may not be the most suitable choice for parcel damage detection due to feature extraction and localization precision limitations. We propose improvements to the Faster R-CNN model to address these limitations and enhance the efficiency and accuracy of parcel damage detection in logistics scenarios. In our proposed RPA R-CNN model, we incorporate ResNet-101 [11] and PANet [13] to enhance the extraction of image features from parcel images. ResNet-101 excels at capturing detailed information within parcel images, while PANet effectively handles features at different scales. We considered alternative models, including YOLOX-Tiny [6], but ultimately selected ResNet-101 and PANet due to their superior accuracy and stability in parcel damage detection tasks. Although lightweight models like YOLOX-Tiny may offer improved speed in specific scenarios, their performance in complex scenes and with small-sized targets falls short of that of ResNet-101 and PANet. To evaluate the efficacy of our model, we conducted experiments using a parcel detection dataset provided by Roboflow [20] and our self-compiled training dataset. Subsequently, we performed verification tests. The results affirm the superiority of our proposed RPA R-CNN model over the original Faster R-CNN, demonstrating higher precision and accuracy in detecting parcel damage within logistics operations.

Our research endeavor aims to provide the e-commerce and logistics industries with an efficient and accurate method for parcel damage detection, thereby enhancing the safety and delivery precision of goods. Furthermore, our proposed methodology possesses a versatile nature that allows an extension to other detection problems, thereby offering practical value in real-world logistics applications.

2 Related Work

The R-CNN algorithm, introduced by Ross Girshick et al. [7] in 2014., was a significant milestone for object detection. However, processing individual regions separately resulted in slow performance and high computational complexity, making it less practical for real-world applications. To address this challenge, Joseph et al. [16] introduced the YOLO algorithm, which revolutionized object detection by adopting a fully convolutional network structure. Notably, YOLOv3 [18] expanded upon YOLOv2 [17] by incorporating multi-scale prediction, residual connections, and a feature pyramid network, establishing it as a crucial object detection algorithm in the field. In 2020, Ultralytics introduced the YOLOv5-S [23] algorithm, which achieved a commendable balance between lightweight network design and object detection performance. To further enhance the accuracy of object detection, Wang et al. [24] proposed YOLOv7. This version employed trainable bag-of-freebies techniques. Although YOLOv7 attained higher accuracy compared to YOLOv5, it exhibited slower runtime performance. Compared to the traditional YOLO series, Zheng et al. [6] introduced YOLOX-Tiny in 2021, a lightweight version of the YOLOX series. This variant maintains high detection accuracy while reducing computational complexity. Similarly, with a focus on one-stage detection, Yang et al. [29] proposed PDNet, which established a more flexible detection paradigm by adopting pixel-wise prediction. Tian et al. [22] proposed a fully convolutional one-stage approach for object detection, eliminating the need for predefined anchor boxes and improving detection accuracy and efficiency. Haq et al. [9] presents a monocular 3D object detection method for autonomous driving, addressing the challenge of accurate object localization in 3D space and achieving improved detection performance. Wang et al. [25] proposed CrabNet, a fully task-specific feature learning method for one-stage object detection, resulting in improved performance and higher accuracy.

To accurately locate small-sized targets and improve detection performance, Ren et al. [19] proposed Faster R-CNN. Unlike the above-mentioned model, Faster R-CNN utilizes a two-stage detection strategy for accurate localization and detection of small objects while maintaining high speed. It also offers flexibility for future advancements in this field. Several variant algorithms derived from Faster R-CNN, such as Mask R-CNN [10], Cascade R-CNN [2], and so on [1,15,30,32], have also been developed. Tan et al. [21] proposed the EfficientDet object detector, which is built upon the EfficientNet backbone network and introduces the BiFPN and composite scaling methods. These innovations enable EfficientDet to achieve better efficiency in resource-constrained scenarios. ResNet has been widely utilized in many object detection studies with impressive results. Li et al. [12] introduced GFL, which employs ResNet-101 as the backbone network and optimizes continuous labels using the GFL, leading to efficient dense object detection. Another ResNet-based method is SOLO, introduced by Wang et al. [26]. SOLO is a position-based object segmentation algorithm that accurately segments objects by predicting their instance center points and generating dense object segmentation masks. In 2022, Zhang et al. [31] proposed ResNeSt, which utilizes the ResNet structure in the backbone network and enhances feature modeling capabilities through a segmentation attention module. Notably, the Region Proposal Network (RPN) and Region of Interest (RoI) also play critical roles in these algorithms. Xie et al. [28] developed Oriented R-CNN, which focuses on RoI and introduces a direction-sensitive RoI alignment module, resulting in improved directional object detection. Similarly, Ou et al. [14]

proposed AD-RCNN, which enhances the performance and efficiency of small object detection by incorporating a dynamic RPN, visual attention mechanism, and adaptive dynamic training module. The aforementioned research endeavors have made significant contributions to the advancement of technology in the domain of object detection. These studies furnish valuable references for future investigations concerning object detection and its integration with the IoT.

3 RPA R-CNN Framework

Within this section, we predominantly introduce our refined model for Faster R-CNN, referred to as RPA R-CNN. Our RPA R-CNN approach primarily focuses on modifying its feature extraction network while enhancing performance. This adaptation facilitates more precise detection of logistics parcels and enhances the overall detection rate. The structure of our RPA R-CNN model is illustrated in Fig. 1.

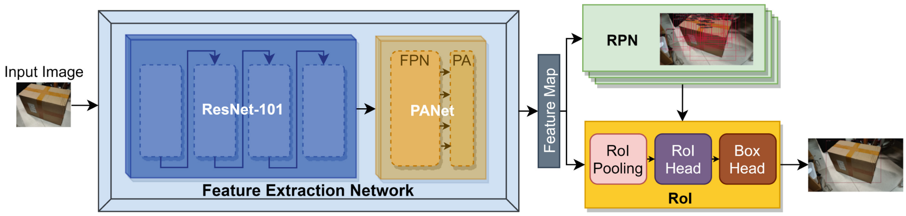


Fig. 1. The proposed RPA R-CNN framework.

3.1 Feature Extraction Network

Our feature extraction process for parcel images primarily relies on ResNet-101 as the backbone network. It consists of 4 stages, each composed of several residual blocks. These blocks extract feature vectors with different resolutions and semantics, allowing the network to capture information at various scales and enhance recognition and classification performance. However, to further optimize the feature propagation and leverage the interaction between features, we introduce the PANet module. The PANet module processes and refines the extracted features, promoting better contextual understanding and preserving relevant details.

In our model, we associate the feature maps obtained from the four stages of the ResNet-101 network with four scales of [256, 512, 1024, 2048]. These feature maps are then fed into the PANet model. This process can be represented as:

$$\begin{cases} F_1 = R_1(I) \\ F_2 = R_2(F_1) \\ F_3 = R_3(F_2) \\ F_4 = R_4(F_3) \\ F_{\text{out}} = P(F_1, F_2, F_3, F_4) \end{cases} \tag{1}$$

where I represents the input image, $R1, R2, R3, R4$ represent the upsampling modules corresponding to the four stages of ResNet-101. The features obtained from the four stages of ResNet-101, denoted as $F1, F2, F3, F4$, have scales of $S1 = 256, S2 = 512, S3 = 1024, S4 = 2048$. These features are inputted into the PANet model, resulting in a high-level abstract representation F_{out} of the input image.

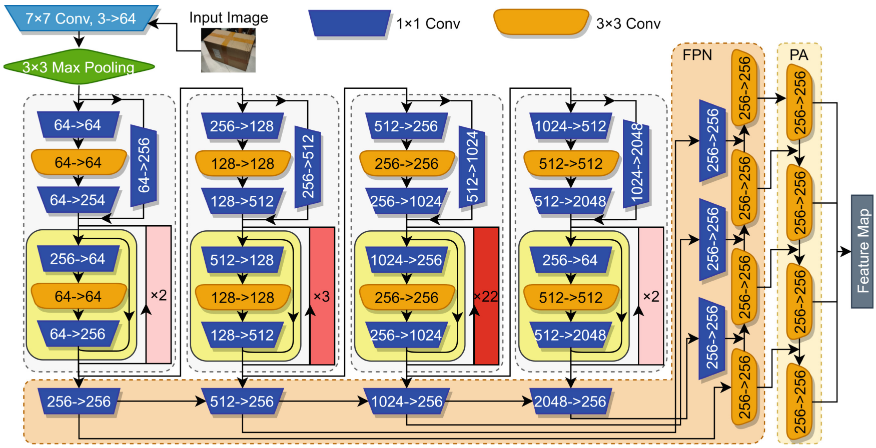


Fig. 2. Feature extraction network framework.

When features are input into PANet, we initially utilize FPN (Feature Pyramid Network) to integrate feature maps from various levels. The feature maps obtained from the four stages of the backbone network are individually passed through a top-down downsampling path, where each feature map undergoes a convolution operation and is downsampled:

$$D_i = Ds(D_{i-1}) + W_i F_i \tag{2}$$

where D_i represents the downsampled feature map at level i, $Ds(\cdot)$ denotes the downsampling operation, and W_i represents the corresponding weight matrix. Although this process may reduce some details in the image or feature maps, it still retains important structures and features. Next, the feature maps go through the bottom-up path for upsampling, and lateral connections are introduced to obtain feature maps with different resolutions and semantic levels:

$$U_i = Us(U_{i+1}) + W_i' F_i \tag{3}$$

where U_i denotes the upsampled feature map at level i, $Us(\cdot)$ represents the upsampling operation, and W_i' signifies the corresponding weight matrix. To further optimize the performance, lateral propagation is used:

$$C_i = U_i + D_i \tag{4}$$

where C_i represents the fused feature map at the i-th level, which effectively captures object information across different scales. Subsequently, the processed features are passed into PA (Path Augmentation) to enhance the feature pyramid's performance by incorporating lateral connections and bottom-up PA. This process enables the extraction of high-level abstract representations from the input image. Figure 2 illustrates our feature extraction network architecture.

3.2 RPN and RoI Network

After the feature extraction network, high-level abstract representations of the input image are obtained. The RPN is essential for generating proposals. In our approach, 1000 proposals per image are generated. These proposals are then categorized as damaged parcels, undamaged parcels, or regions excluded from training. We employ the Random Sample method to randomly select candidate boxes for damaged and undamaged parcels. These samples constitute the training set used for model training. This strategy effectively addresses the issue of imbalanced positive and negative samples during training, thereby enhancing the efficiency and detection performance of the model.

We process the proposals generated by the RPN by feeding them, along with the feature map, into the RoI layer. Here, we employ RoI pooling to pool the proposals into equally-sized feature map blocks and generate feature vectors using max pooling. Next, we feed the feature vectors, bbox (bounding box) coordinates, and labels into the ROI Head for further processing. In this layer, our primary focus is on regressing the predicted bbox coordinates using the multi-task loss function:

$$\begin{cases} L_{cls}(p_i, p_i^*) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) \\ L_{reg}(t_i, t_i^*) = \frac{1}{N_{reg}} p_i^* L_{reg}(t_i, t_i^*) \end{cases} \tag{5}$$

where L_{cls} and L_{reg} refer to the classification loss and regression loss, respectively. N_{cls} and N_{reg} ensure a balanced influence of both losses in the overall loss function. λ is a balance coefficient that adjusts the weight between the two losses. p_i and t_i are the predicted values for proposal classification and bbox regression, while p_i^* and t_i^* are the ground truth labels. Finally, the output features from the ROI Head undergo further refinement through the Box Head, enhancing the accuracy of object detection and providing us with classification results and predicted bboxes for the input image.

4 Experiment

4.1 Dataset

Our dataset comprises 2400 images, meticulously curated to encompass a wide range of packaging scenarios commonly encountered in real-world logistics operations. These images were initially collected through web scraping using a custom web scraper and were subsequently augmented by incorporating images from Roboflow. The dataset is thoughtfully labeled with four distinct categories: Box, Box_broken, Open_package, and Package, ensuring comprehensive coverage of logistics-related packaging conditions. The training, validation, and test sets consist of 1600, 400, and 400 images, respectively.

Table 1. The parameters used during the training process.

Batch size	Epoch	Learning rate	Optimizer	Loss function
4	120	0.0003	Adam	Multi-Task Loss Function

4.2 Experimental Setup

We used MMDetection, a comprehensive framework for object detection, which supports various algorithms and components. It simplifies experimentation for researchers by providing a configuration-based approach for efficient model construction, training, and customization. The training parameters used are presented in Table 1.

Table 2. Model performance comparison of different models.

Model	mAP	mAP_50	mAP_75	Epoch
YOLOv3	0.45	0.791	0.493	120
YOLOX-Tiny	0.498	0.841	0.576	120
YOLOv5-S	0.501	0.853	0.587	120
Faster R-CNN (ResNet101)	0.632	0.867	0.758	120
Faster R-CNN (ResNeSt)	0.696	0.879	0.785	120
RPA R-CNN	**0.74**	**0.906**	**0.827**	120

4.3 Model Performance Comparison

To evaluate the effectiveness of our model, we compared the mAP values of several models, including YOLOv3, YOLOX-Tiny, YOLOv5-S, Faster-RCNN with ResNet101, Faster-RCNN with ResNeSt, and RPA R-CNN, at different confidence threshold levels. As shown in Table 2, it is evident that the RPA R-CNN model achieved the highest mean Average Precision (mAP) score of 0.74 during the same epoch. Furthermore, we examined precision values at mAP thresholds of 0.5 and 0.75. Under the more lenient matching criterion (mAP threshold of 0.5), the RPA R-CNN model reached an mAP_50 value of 0.906, confirming its high level of accuracy. Additionally, under the stricter matching criteria (mAP threshold of 0.75), the model consistently delivered commendable performance, maintaining an mAP_75 value of 0.827. These results demonstrate the consistent superior accuracy of our RPA R-CNN model across different confidence thresholds. These results show that our RPA R-CNN model consistently achieves the highest accuracy among the six models at different confidence thresholds. It is important to note that YOLO series models, such as YOLOX-Tiny, offer faster inference speeds and can be more suitable for real-time applications, but they may compromise accuracy to some extent. In contrast, models from the RCNN series, including our proposed RPA R-CNN, exhibit higher accuracy but may require more computational resources. Our choice of RPA R-CNN is driven by the specific needs of logistics parcel inspection, where accuracy is paramount, and real-time performance can be balanced with available computing resources. Therefore, we emphasize that the efficiency of our model aligns with the precision requirements of the task. RPA R-CNN's superior accuracy, as demonstrated in the comparison results, makes it a suitable choice for damaged logistics parcel inspection tasks where high accuracy and stability are of utmost importance.

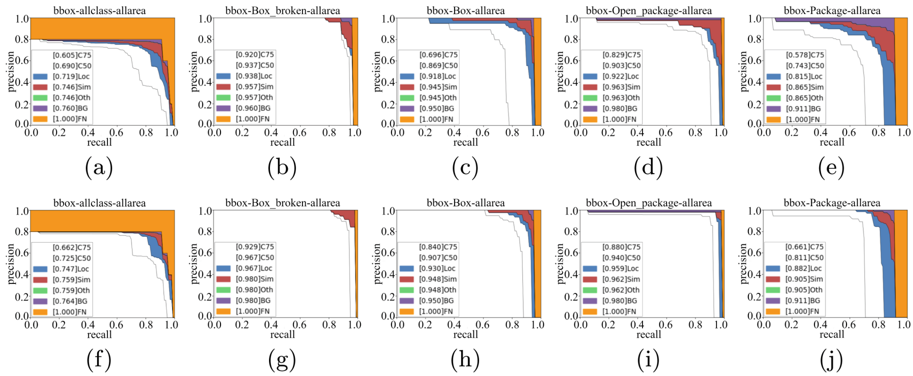


Fig. 3. PR curve of Faster R-CNN (FPN) and RPA R-CNN for parcel damage detection.

4.4 Evaluation of Model Enhancement

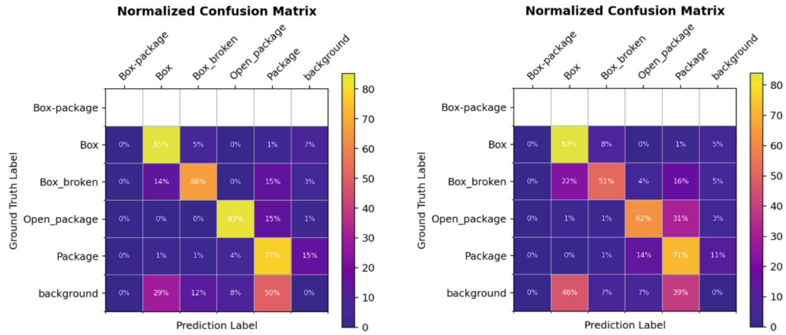
To validate the enhancement of our RPA R-CNN model, we compared the PR curves of the Faster R-CNN (FPN) and RPA R-CNN. As shown in Fig. 3. Subfigures (a), (b), (c), (d), and (e) depict the detection of parcels using Faster R-CNN(FPN) for various categories: allclass, box_broken, box, open_package, and package respectively. Similarly, subfigures (f), (g), (h), (i), and (j) depict the detection outcomes using RPA R-CNN for the corresponding categories.

From the PR curves of the two models, RPA R-CNN demonstrates significant advancements in various performance metrics across different parcel categories. Comparing Fig. 3(a)–(e) with (f)–(i) concurrently, we can observe the following: the "allclass" category exhibits a notable improvement with a 5.7% increase in the C75 metric. Similarly, for the "box_broken" category, RPA R-CNN achieves the maximum enhancement with a 3% boost in the C50 metric. Remarkably, the "box" category showcases the highest improvement among all categories, displaying a remarkable 14.4% increase in the C75 metric. Evaluating the "open_package" category, both the C75 and C50 metrics display relatively similar percentage improvements of 5.1% and 4% respectively. On the other hand, in the "package" category, the C75 metric demonstrates the most substantial improvement, with an impressive 8.3% increase. Additionally, despite being relatively modest, the improvements in the Loc, Oth, BG, and FN metrics, which take into account various confounding factors, indicate that these metrics have either remained stable or exhibited slight enhancements. To summarize, our findings indicate that RPA R-CNN outperforms the original model across all evaluated metrics. Furthermore, the enhanced model exhibits a PR curve that closely approximates the standard curve, indicating excellent training performance. In addition, the improved model outperforms the original model in terms of PR curve performance.

Table 3. Parcel damage detection precision comparison.

Ground Truth Label→Prediction Label	Precision	
	RPA R-CNN	Faster R-CNN(FPN)
Box→Box	**85%**	83%
Box Broken→Box Broken	**66%**	51%
Open Package→Open Package	**83%**	62%
Package→Package	**77%**	71%
Background→Box	**29%**	46%

To intuitively observe the improvement of our RPA R-CNN model, we plotted the key data as a confusion matrix, as shown in Fig. 4. By analyzing the key data presented in the confusion matrix (Table 3), we observed that our RPA-RCNN surpasses Faster-RCNN (FPN) in detecting Box, Box Broken, Open Package, and Package tasks. Despite both models showing a tendency to misclassify background labels, our RPA-RCNN exhibits significant improvements, especially in Box Broken and Open Package tasks, with a boost of 15% and 21% respectively. These results show that RPA R-CNN achieves higher precision across various parcel inspection tasks. Notably, the significant enhancement in detecting Box Broken and Open Package tasks specifically suggests that our model is particularly suitable for inspecting damaged packages. However, it's worth noting that the RPA R-CNN model's size increased to approximately 747 MB compared to around 470 MB for the Faster R-CNN (FPN) model. This increase in model size was a trade-off for improved detection accuracy.



(a) RPA R-CNN (b) Faster R-CNN(FPN)

Fig. 4. Confusion matrix of parcel damage detection on different models.

5 Conclusion

In this study, we propose RPA R-CNN, a deep learning-based object detection method for detecting parcel damage in IoT environments such as warehouse automation and conveyor belt transportation. Our approach improves upon Faster R-CNN by modifying the feature extraction network to ResNet-101 with PANet for accurate parcel image feature extraction, then leveraging RPN and ROI techniques for processing the extracted features. To evaluate our model's effectiveness, we employ parcel detection datasets from Roboflow and internet-crawled images for training. We conduct comparative experiments with commonly used object detection models, and our results indicate that RPA R-CNN excels in mean average precision. Through model enhancement evaluation, we demonstrate that RPA R-CNN achieves superior detection rates for domestic or cross-border logistics parcels compared to the original model.

We observe the relevance of parcel damage localization detection in IoT scenarios like warehouse automation and conveyor belt systems. In practical applications, our model can provide real-time monitoring, data transmission, and analysis for detecting parcel damage. This can significantly improve efficiency and service quality in the logistics industry. Future research directions include expanding our findings to other detection problems and developing practical methods and techniques for the IoT field. By leveraging the advantages of IoT, we can further enhance real-time monitoring and analysis capabilities, contributing to the improvement of logistics operations.

Acknowledgements. This work was supported by the Science and Technology Planning Project of Guangdong (2021B0101420003, 2020ZDZX3013, 2023ZZ03), NSFC (62077028, 62377028), Guangdong Key Laboratory of Data Security and Privacy Preserving (2023B1212060036), the Science and Technology Planning Project of Guangzhou (202206030007), the Opening Project of Key Laboratory of Safety of Intelligent Robots for State Market Regulation (GQI-KFKT202205), Guangdong-Macao Advanced Intelligent Computing Joint Laboratory (2020B1212030003) and the High

Performance Public Computing Service Platform of Jinan University. Zhefu Li and Xiaofeng Wu are the corresponding authors.

References

1. Ali, R., Kang, D., Suh, G., Cha, Y.J.: Real-time multiple damage mapping using autonomous UAV and deep faster region-based neural networks for GPS-denied structures. Autom. Constr. **130**, 103831 (2021)
2. Cai, Z., Vasconcelos, N.: Cascade R-CNN: delving into high quality object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6154–6162 (2018)
3. Chen, Z., et al.: Few-shot learning for quality detection of logistical parcels. In: 2023 11th International Conference on Information Systems and Computing Technology (ISCTech). IEEE (2023)
4. Cheng, X., Li, N., Rysbayeva, G., Yang, Q., Zhang, J.: Influence-aware successive point-of-interest recommendation. World Wide Web **26**(2), 615–629 (2023)
5. Dilshad, N., Ullah, A., Kim, J., Seo, J.: LocateUAV: unmanned aerial vehicle location estimation via contextual analysis in an IoT environment. IEEE Internet Things J. **10**(5), 4021–4033 (2022)
6. Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: YOLOX: exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430 (2021)
7. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
8. Guo, S., Zhao, C., Wang, G., Yang, J., Yang, S.: EC^2Detect: real-time online video object detection in edge-cloud collaborative IoT. IEEE Internet Things J. **9**(20), 20382–20392 (2022)
9. Haq, M.A., Ruan, S.J., Shao, M.E., Haq, Q.M.U., Liang, P.J., Gao, D.Q.: One stage monocular 3D object detection utilizing discrete depth and orientation representation. IEEE Trans. Intell. Transp. Syst. **23**(11), 21630–21640 (2022)
10. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
12. Li, X., et al.: Generalized focal loss: learning qualified and distributed bounding boxes for dense object detection. In: Advances in Neural Information Processing Systems, vol. 33, pp. 21002–21012 (2020)
13. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8759–8768 (2018)
14. Ou, Z., et al.: AD-RCNN: adaptive dynamic neural network for small object detection. IEEE Internet Things J. **10**(5), 4226–4238 (2023). https://doi.org/10.1109/JIOT.2022.3215469
15. Pang, J., Chen, K., Shi, J., Feng, H., Ouyang, W., Lin, D.: Libra R-CNN: towards balanced learning for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 821–830 (2019)
16. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)

17. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7263–7271 (2017)
18. Redmon, J., Farhadi, A.: YOLOv3: an incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
19. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, vol. 28 (2015)
20. Roboflow: Parcel detection dataset (2023). https://universe.roboflow.com/detectionbox
21. Tan, M., Pang, R., Le, Q.V.: EfficientDet: scalable and efficient object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10781–10790 (2020)
22. Tian, Z., Shen, C., Chen, H., He, T.: FCOS: fully convolutional one-stage object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9627–9636 (2019)
23. Ultralytics: YOLOv5: An open source real-time object detection system (2020). https://github.com/ultralytics/yolov5
24. Wang, C.Y., Bochkovskiy, A., Liao, H.Y.M.: YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7464–7475 (2023)
25. Wang, H., Wang, Q., Zhang, H., Hu, Q., Zuo, W.: CrabNet: fully task-specific feature learning for one-stage object detection. IEEE Trans. Image Process. **31**, 2962–2974 (2022)
26. Wang, X., Kong, T., Shen, C., Jiang, Y., Li, L.: SOLO: segmenting objects by locations. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12363, pp. 649–665. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58523-5_38
27. Weng, X., Huang, Y., Li, Y., Yang, H., Yu, S.: Unsupervised domain adaptation for crack detection. Autom. Constr. **153**, 104939 (2023)
28. Xie, X., Cheng, G., Wang, J., Yao, X., Han, J.: Oriented R-CNN for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3520–3529 (2021)
29. Yang, L., et al.: PDNet: toward better one-stage object detection with prediction decoupling. IEEE Trans. Image Process. **31**, 5121–5133 (2022)
30. Zhai, Y., Yang, K., Zhao, Z., Wang, Q., Bai, K.: Geometric characteristic learning R-CNN for shockproof hammer defect detection. Eng. Appl. Artif. Intell. **116**, 105429 (2022)
31. Zhang, H., et al.: ResNeSt: split-attention networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2736–2746 (2022)
32. Zou, W., Zhang, Z., Peng, Y., Xiang, C., Tian, S., Zhang, L.: SC-RPN: a strong correlation learning framework for region proposal. IEEE Trans. Image Process. **30**, 4084–4098 (2021)