Advances in Artificial Intelligence, Big Data and Algorithms
G. Grigoras and P. Lorenz (Eds.)

doi:10.3233/FAIA230818

The Application of Artificial Intelligence in Geotechnical Investigation

Haifeng MEI[a1], Wei ZHANG[b,2], Jing GU[b3]
[a]*Shaanxi Transportation Planning and Design Institute Co., LTD*
[b]*Xidian University, School of Artificial Intelligence*

Abstract. The standard penetration test (SPT) and dynamic probing test (DPT) are commonly used exploration methods in geotechnical investigation. However, errors can occur during data collection, often attributed to factors such as human error. To mitigate this issue, this paper proposes the utilization of an improved YOLOv5 object detection algorithm, a form of artificial intelligence technology, to automatically count the number of hammer strikes during geotechnical investigations. The proposed approach incorporates several enhancements to the YOLOv5 network architecture. Firstly, a focal loss function is introduced to address sample imbalance, ensuring better handling of different classes of hammer strikes. Additionally, online hard example mining technology is employed to improve model accuracy by focusing on challenging samples that are most informative for training. The improved YOLOv5 model is then applied to detect hammer strikes in SPT and DPT tests. To facilitate training and evaluation, a hammer detection dataset is created, tailored to the specific requirements of geotechnical investigation. Experimental results demonstrate the superior performance of the proposed improved YOLOv5 object detection model on the hammer detection dataset.

Keywords. Geotechnical investigation, hammer detection, YOLOv5, online hard example mining.

1. Introduction

Geotechnical investigation encompasses several stages, including the task assignment, project planning, field survey, indoor testing, data processing, review and approval of survey data, and submission of project deliverables, which aims to gather comprehensive information about construction projects, so it plays a crucial role during the construction process. However, ensuring the accuracy of measurement data is a significant concern in geotechnical investigation. In particular, during standard penetration tests and dynamic cone penetration tests, it is crucial for surveyors to accurately count the number of hammer strikes in each trial. Manual data collection introduces inherent uncertainty, which can lead to errors in the statistical data and ultimately impact the results of the geotechnical investigation. With the advancement of artificial intelligence technology, various artificial intelligence (AI) methods have

[1] Haifeng MEI, Shaanxi Transportation Planning and Design Institute Co., LTD;
e-mail: meihaifeng@163.com
[2] Corresponding Author: Wei ZHANG, Xidian University, School of Artificial Intelligence;
e-mail: zhanghuohuo121@163.com
[3] Jing GU, Xidian University, School of Artificial Intelligence; e-mail: jgu6126@163.com

been successfully applied across industries. This paper proposes the utilization of advanced AI techniques to automatically count in the progress of geotechnical investigation, thereby eliminating inaccuracies in survey data caused by human factors. It aims to address issues related to construction quality and efficiency, offering a more reliable and efficient alternative to manual counting.

In the field of AI, object detection has found widespread applications in various domains, including intelligent video surveillance, autonomous driving, and aerial drone imaging. The objective of object detection is to automatically identify specific objects in images or videos. Existing object detection algorithms can be categorized into two main types: the region extraction-based object detection methods (such as RCNN [1], Fast RCNN [2], Faster RCNN [3]) and the end-to-end object detection methods (such as YOLO [4], SSD [5]). The application of object detection algorithms in geotechnical investigation can provide engineers with valuable tools to efficiently and accurately gather essential information, assess construction quality, and identify potential hazards. Therefore, exploring the practical implementation of object detection algorithms within the context of geotechnical investigation holds significant practical value.

Object detection algorithms have experienced remarkable progress in recent years, leading to their widespread adoption in various domains. As different scenarios and objects present unique challenges, a series of detectors have been developed to effectively tackle these challenges. One pioneering object detection technique was R-CNN, which employed the selective search technique to extract candidate regions from input images, followed by classification and bounding box regression for each candidate region. Building upon the foundation of R-CNN, Fast R-CNN and Faster R-CNN algorithms were introduced to further enhance object detection capabilities. Fast R-CNN introduced the Region of Interest (ROI) pooling layer, which improved detection accuracy and speed by effectively handling candidate boxes. Next, Faster R-CNN integrated the Region Proposal Network (RPN) into the detection framework, which automates the process of generating candidate regions, achieving significant improvements in speed and overall performance.

The series of R-CNN algorithms demonstrated good accuracy in the object detection tasks. However, they suffered from slow speed, cumbersome usage, and limitations in real-time high-speed applications. A series of YOLO algorithms addressed these challenges, significantly improving both real-time speed and accuracy of object detection. The YOLOv1 algorithm [4] proposed in 2016 utilized a fully connected neural network structure and adopted a single-image global object detection approach, which enabled fast detection of multiple objects in the real-time scenarios. However, its relatively shallow network structure limited its accuracy, making it less suitable for the high-precision scenarios. Subsequent versions, such as YOLOv2 [6], integrated Batch Normalization (BN) layers and anchor boxes to handle objects with different aspect ratios and sizes. YOLOv3 [7] improved the speed and stability of model by reconstructing the DarkNet framework. Although a series of YOLO algorithms gradually improved the accuracy of object detection, it has been observed that their speed has gradually decreased. In 2020, YOLOv4 [8] and YOLOv5 [9] pushed the boundaries of both accuracy and speed, making them standout algorithms in the field of object detection. They achieved the highest average precision (AP) on COCO [10] dataset at that time, surpassing other algorithms such as Faster RCNN and Mask RCNN [11].

In general, it is crucial to find a balance between accuracy and speed, depending on the specific requirements of each task. Among the various object detection algorithms available, YOLOv5 has emerged as a strong contender due to its exceptional performance. It offered a combination of fast execution speed, high accuracy, and precise localization capabilities, which made it highly acclaimed and widely adopted in recent years. This paper extends the capabilities of YOLOv5 framework by introducing additional enhancements. Specifically, it incorporates the focal loss function and online hard example mining method, which is applied to the detection of hammer impacts in the standard penetration and dynamic penetration tests conducted as part of geotechnical investigations. By automatically counting the hammer impacts during these tests, the proposed approach streamlines the data collection process. The findings of this study have significant implications for the integration of artificial intelligence technologies in the engineering site investigations. They provide valuable insights and guidance for future research and development in this field.

The contributions of this paper include the following:

- We first success in applying the object detection algorithm in the field of artificial intelligence to the hammer detection during geotechnical investigations and give an automatic counting method to automatically recognize the number of hammer strikes.
- An improved YOLOv5 is presented, where the focal loss function is introduced into the classification loss. It allows the model to pay more attention to hard-to-detect and less abundant positive samples.
- The Online Hard Example Mining (OHEM) is utilized to dynamically adjust and improve the accuracy of model for some challenging objects, thus enhancing the performance of model on the hard-to-detect targets.

The rest of this paper is organized as follows. Section II reviews the evolution of a series of YOLO algorithms, while Section III provides a detailed description of our improved YOLOv5 method. Section IV shows and analyzes the experimental results. Finally, Section V concludes the article and discusses the prospects for the future development.

2. Evolution of YOLO

The YOLOv1 algorithm proposed by Joseph Redmon et al. introduced a single neural network to directly predict bounding boxes and class probabilities for objects in an image. This end-to-end approach could process a large number of data efficiently, making it a fast and scalable solution [12]. However, YOLOv1 has limitations in terms of localization accuracy and potential misclassification of overlapping objects.

YOLOv2 employed several techniques to enhance the accuracy and precision of object detection. A batch normalization layer was employed to improve the stability of the model, and the anchor prior box was utilized to enhance the stability of the model. Moreover, YOLOv2 addressed the challenge of detecting small objects by fusing low-level fine-grained features with convolved high-level semantic features through channel merging. Although YOLOv2 showed improved accuracy compared to its predecessor YOLOv1, the challenge related to localization and classification errors persisted.

YOLOv3 had further advancements over YOLOv2. It designed a novel backbone network called DarkNet-53, which combined elements from Darknet-19 and ResNet [13] architectures to enable compatibility with inputs of varying scales, enhancing the model's versatility. The K-means clustering was also adopted to generate anchor boxes of different scales, improving the ability to detect small and densely packed objects.

Alexey Bochkovskiy et al. made further optimizations to YOLOv3 and proposed YOLOv4 in 2020. The method implemented the Cross-Stage-Partial (CSP) connection structure, which facilitated effective information transmission between preceding and succeeding layers, leading to improved detection accuracy and speed. Attention mechanisms such as Spatial Attention Module (SAM) and Convolutional Block Attention Module (CBAM) were incorporated to refine feature maps, enhancing detection accuracy and the ability to detect small objects. The introduction of Path Aggregation Network (PANet) [14] further improved feature extraction capabilities, enabling the model to better detect small and multiple objects. Additionally, the utilization of algorithms and modules such as SPP-blocks and panoptic feature pyramid networks contributed to enhance performance significantly.

YOLOv5 was proposed in June 2020, distinguishing itself from previous versions through a lightweight network architecture and the inclusion of state-of-the-art techniques such as swish activation and drop-block, which further enhanced the detector's performance. It also introduced a novel data augmentation strategy specifically designed for the small objects, resulting in improved detection accuracy for such targets.

3. The Improved YOLOv5 Model

3.1. Network Structure of YOLOv5

YOLOv5 offers four versions based on network depth and width: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x [15-17]. For this paper, the YOLOv5l model was selected to align with the specific requirements of the geotechnical investigation task. Compared to YOLOv5s and YOLOv5m, YOLOv5l has a larger model size while still maintaining a fast inference speed relative to YOLOv5x, making it suitable for real-time detection needs.

The key components of YOLOv5 encompass the input end, backbone, neck, and prediction output end (Head), as shown in Figure 1. The input end employs Mosaic data augmentation, which involves randomly cropping and concatenating four images, along with random scale resizing and horizontal flipping. This augmentation technique significantly enriches the dataset and enhances the robustness and generalizability of the model.

For the backbone, YOLOv5 employs CSPDarknet53 as the core architecture, incorporating modules such as Cross Stage Partial (CSP) and Cross-Stage-Partial (CBS). The CBS module consists of convolutional layers, batch normalization layers, and SiLU activation layers, which effectively reduces information loss and computation during feature transmission. The CSP module was originally introduced in the backbone of YOLOv4, which enhanced feature extraction capabilities by establishing connections across different residual blocks and feature branches, resulting in improved detection accuracy and speed. YOLOv5 employs two types of CSP structures: CSP1 and CSP2. The CSP1 module comprises two paths, where one path

contains the CBS module and a residual block, while the other path includes a single CBS module for channel adjustment in the backbone. In contrast, the CSP2 module, used in the neck component, replaces the original residual block with the CBS module. This design ensures the performance while reducing computational complexity, enhancing the model's feature representation capability, and improving the performance of YOLOv5 in object detection tasks.

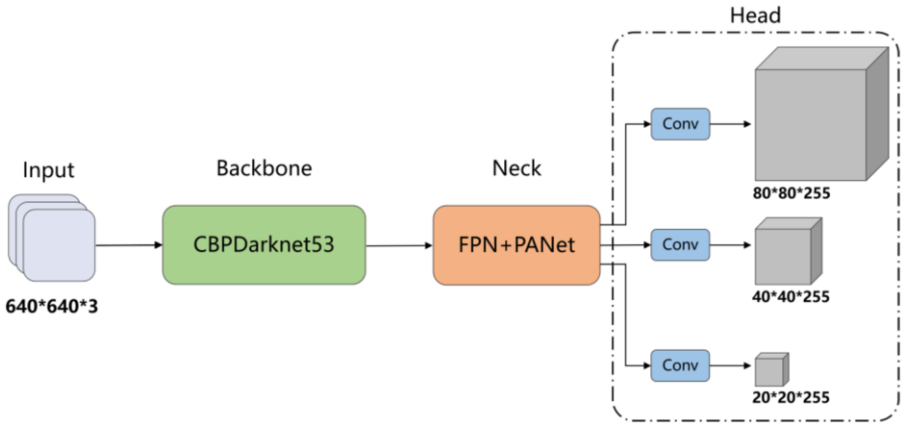

Figure 1. Overall framework of YOLOv5

The neck component of YOLOv5 consists of the SPPF module, CSP module, and a combination of the Feature Pyramid Network (FPN) [18] with the Path Aggregation Network (PAN). The SPP module enhances the ability of network to handle images of various sizes by utilizing max pooling, enabling the network to process images of arbitrary scales. The SPPF module is a lightweight version of the SPP module, offering higher efficiency while maintaining effectiveness. The FPN performs feature fusion through upsampling strategy. On the other hand, the PAN propagates positional information through the feature pyramid in a bottom-up manner. This design could simultaneously utilize the high-resolution information from the low-level features and the high-level semantic information, thus facilitating the recognition of diverse targets. It empowers the network to learn and extract a wider spectrum of feature information.

In the head component, the GIoU loss function is used to calculate the bounding box loss, which considers not only the intersection over union (IoU) between the predicted bounding box and the ground truth box, but also factors such as the distance between their centroids and aspect ratios. By incorporating these additional metrics, the GIoU loss function provides a more accurate calculation of bounding box loss, thereby improving the detection accuracy of model.

3.2. Online Hard Example Mining

The hammer dataset utilized in this study was created specifically for this research. The dataset consists primarily of simple samples, which account for over 90% of the entire dataset. However, the dominance of these simple samples during model training limits the effectiveness of model, which uses the dataset, in detecting more challenging targets. To address this limitation, the technique of Online Hard Example Mining (OHEM) [19] is employed during training, which dynamically adjusts the training process by selecting difficult samples based on the loss and incorporating them into

further training iterations. By actively incorporating challenging instances, the model is guided to learn better and improve its performance in detecting hard-to-discern objects.

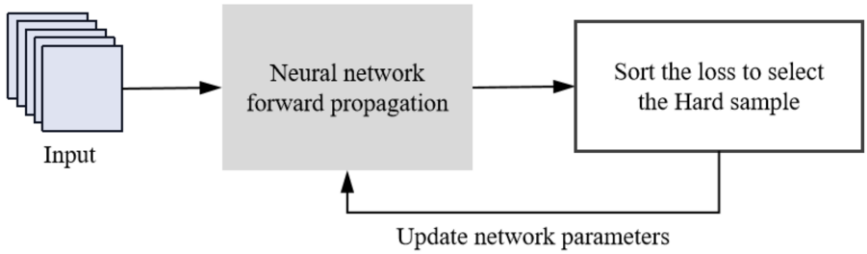

Figure 2. The process of training by OHEM

As shown in Figure 2, the implementation of OHEM based on YOLOv5 consists of the following steps:

1)For each mini-batch of data, the input data was fed through the network, generating the predicted bounding box positions and class probabilities.

2)The loss was computed by quantifying the discrepancies between the predicted bounding boxes and the ground truth boxes in terms of the class probabilities and positions.

3)The losses were sorted, and a subset of boxes with larger losses were selected as the hard samples.

4)The hard samples are utilized to update the network parameters through the back propagation [20].

By incorporating OHEM into the training process, the model can place emphasis on challenging targets, which make the model to better learn the distinctive features of these targets, resulting in improved accuracy and robustness of the object detection model.

3.3. Focal Loss Function

In YOLOv5, the loss function can be expressed as:

$$L = \lambda_{coord} L_{coord} + \lambda_{obj} L_{obj} + \lambda_{cls} L_{cls} + \lambda_{giou} L_{giou} , \tag{1}$$

where L_{coord} represents the localization loss, L_{obj} represents the object confidence loss, L_{cls} represents the classification loss, and L_{giou} represents the GIoU loss. λ_{coord}, λ_{obj}, λ_{cls}, and λ_{giou} are the hyper-parameter used to balance different loss.

Due to the severe class imbalance between the positive and negative samples in the self-built hammer dataset, employing a standard cross-entropy loss function would result in the model prioritizing the numerous negative samples, which could negatively impact the detection performance for positive samples. In order to mitigate this issue, the focal loss function [21] is introduced to address the problem of imbalanced samples, which dynamically adjust the weights of negative samples, thereby encouraging the model to focus on positive samples. The focal loss set an adjustable parameter that controls the weight assigned to negative samples. When this parameter is set to 0, the

focal loss function reduces to a standard binary cross-entropy loss. However, as the parameter value increases, the focal loss function effectively reduces the weight assigned to easily classifiable samples, thereby directing the attention of model towards challenging positive samples that are relatively scarce [22]. The formulation of focal loss function is as follows:

$$FL(p_t) = -\alpha_t (1-p_t)^{\gamma} \log(p_t), \tag{2}$$

where the variable p_t is defined as: $p_t = \begin{cases} p & if\ y=1 \\ 1-p & otherwise \end{cases}$, the predicted probability of the model for positive samples. p is the probability assigned by the model for a sample belonging to the foreground (positive class), while y takes values of 1 and -1, representing the foreground and background respectively. α_t is the weighting factor, which is set to α for the positive samples and $1-\alpha$ for the negative samples. This loss helps address the issue of imbalanced positive and negative samples of dataset.

What's more, a modulation factor γ is employed in the focal loss to emphasize the challenging samples that are difficult to be distinguished. In the improved YOLOv5 model of this paper, the focal loss function is employed to compute the classification loss L_{cls}, which can be represented as follows:

$$L_{cls} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}[y_i = c](1-p_{i,c})^{\gamma} \log(p_{i,c}), \tag{3}$$

where N is the number of training samples, C denotes the localization loss, y_i is the true class of the i -th sample, and $p_{i,c}$ represents the predicted probability of the model for the i -th sample belonging to the class c. The utilization of focal loss function makes the training of object detection model more effective, thereby improving the detection performance on challenging positive samples. Furthermore, it further enhances the generalization ability of model during the training process, since the focal loss function allows for dynamic adjustment of sample weights.

4. Experimental Results and Analysis

4.1. Dataset Construction

A specialized dataset is constructed in this paper, specifically for the detection of hammers, which is utilized in standard penetration testing and dynamic probing experiments in the field of geotechnical investigation to provide the fundamental data for further research. It includes images of two different types of hammers and takes into account various influential factors such as varying lighting conditions and angles to simulate the complexities encountered in the real-world scenarios. Figure 3 shows an example of two types of hammers.

(a) hammer1 (b) hammer2

Figure 3. Examples of hammer categories

To construct this dataset, high-definition cameras on-site is utilized to capture the standard penetration testing and dynamic probing experiments. The recorded videos are then processed by extracting individual frames, which are subsequently screened to obtain multiple images. To ensure accurate labeling, the image annotation tool, called labelimg, is employed to annotate the presence of hammers in each image. The dataset comprises a total of 2400 training images and 800 testing images, with each image containing a single hammer target.

4.2. Experimental Setup

The experimental setup included the following components: (1) Hardware Environment: A NVIDIA GeForce RTX 1060s GPU with 6GB VRAM was utilized. (2) Software Environment: The experiments were conducted on the Windows 11 operating system using PyCharm 2022.1 and CUDA 11.8. (3) Deep Learning Framework: PyTorch 2.0.0 was employed as the deep learning framework.

During the model training phase, the Adam optimizer was utilized. The training process employed a batch size of 32, and the model was trained for a total of 100 iterations. The adjustable parameter γ is set to 1.5 in the focal loss function.

4.3. Evaluation Indicators

In this experiment, the mean Average Precision (mAP) was utilized as the evaluation metric to assess the performance and efficiency of the model. Before introducing this metric, the following symbols are defined as follows: TP (True Positive) represents the number of positive samples correctly predicted as positive. It corresponds to the number of predicted bounding boxes that have an intersection-over-union (IoU) with the ground truth boxes greater than or equal to the pre-defined threshold. FP (False Positive) represents the number of negative samples incorrectly predicted as positive. FN (False Negative) represents the number of positive samples incorrectly predicted as negative. TN (True Negative) represents the number of negative samples correctly predicted as negative.

Based on these definitions, the precision (P) and recall (R) for each class can be calculated. Precision is defined as the ratio of TP to the sum of TP and FP, while recall is defined as the ratio of TP to the sum of TP and FN.

$$P_k = \frac{TP}{TP + FP}, \tag{4}$$

$$r_k = \frac{TP}{TP + FN}. \tag{5}$$

Different precision and recall values can be computed by different thresholds. By plotting the Precision-Recall curve with recall on the horizontal axis and precision on the vertical axis, the trade-off between precision and recall for different threshold values is visualized. The area under the Precision-Recall curve represents the Average Precision (AP) for each class. The self-built hammer dataset comprises two types of hammers, so that the calculation formula for mean Average Precision (mAP) is as follows:

$$mAP = \frac{1}{2}\sum_{k=1}^{2} AP_k, \tag{6}$$

where Average Precision (AP) measures the precision of the model in detecting a specific class. And the mean Average Precision (mAP) is the average of AP values across all classes, providing an overall measure of the performance of model in the hammer detection.

4.4. Experimental Result

In this experiment, a modified version of the YOLOv5 algorithm was utilized to train a hammer detection model. The performance of the model was evaluated on the test set using different IoU thresholds, specifically set at [0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95]. The evaluation metric employed for assessing the performance of model is mAP(%), which provides a comprehensive evaluation of the performance across various levels of bounding box overlap, thus indicating its robustness and stability. Figure 4 displays the sample detection results of model on the test set.

Figure 4. Examples of detection results

The performance of model on the test set reaches a stable state after 25 training epochs, as the descripted in Figure 5. Notably, the final model achieves an impressive mAP of 87.53% at an IoU threshold of 0.5. Moreover, the average mAP across different

thresholds ranging from 0.5 to 0.95 is reported to be 83.83%. These results demonstrate the exceptional detection accuracy of model on the test set.

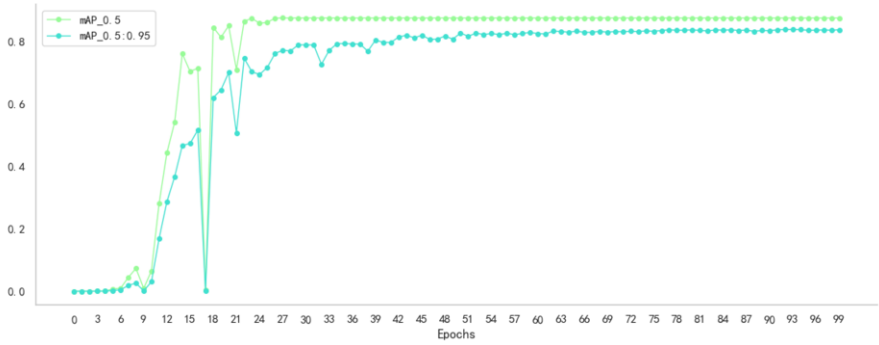

Figure 5. Performance variation of model on the test set

After detecting the position of the falling hammer, the determination of whether is a downward action is based on the variation of center point position of hammer. The specific criterion is as follows. If the downward distance of hammer exceeds 10 units compared to the previous frame for at least four times in eight consecutive frames, it is considered as one downward action. To prevent multiple counting of the same downward action, a 15-frame counting pause is implemented after each detected downward action. This process is considered complete if no downward action is detected within a 10-minute timeframe. Experimental validation has demonstrated the effectiveness of the proposed method that is capable of accurately and automatically counting hammer strikes in standard penetration tests and dynamic probing tests during geotechnical investigations. As a result, it effectively eliminates data errors associated with manual counting.

5. Conclusions and Future Directions

This paper improves the YOLOv5 algorithm, which introduces the focal loss function and the online hard example mining during the training process. The modified YOLOv5 algorithm was trained and tested on a self-built hammer dataset, thus generating a robust hammer detection model. The experimental results show the exceptional performance of the enhanced YOLOv5 algorithm, achieving an average precision of 83.8% mAP in the hammer detection. As ongoing survey projects continue to generate a growing accumulation of on-site video data, the hammer dataset will be continuously enriched and refined, which will enable further exploration of optimization techniques for the hammer detection model, expanding both the breadth and depth of object detection applications in the geotechnical investigation.

References

[1] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation[J]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, 580-587.

[2] Girshick, R..Fast R-CNN[J]. IEEE International Conference on Computer Vision (ICCV), 2015, 1440-1448.
[3] Ren, S., He, K., Girshick, R., & Sun, J.. Faster R-CNN: Towards real-time object detection with region proposal networks[J]. IEEE International Conference on Computer Vision (ICCV), 2015, 91-99.
[4] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection[J]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, 779-788.
[5] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector[J]. European Conference on Computer Vision (ECCV), 2016, 21-37.
[6] Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger[J]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, 6517-6525.
[7] Redmon, J., &Farhadi, A. (2018). YOLOv3: An incremental improvement[J]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, 6517-6525.
[8] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection[J]. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, 10934-10944.
[9] Wong, B., AbdSalam, R., & Wong, S. H. (2020). YOLOv5: A better, faster, stronger object detector[J]. International Conference on Neural Information Processing (ICONIP), 2020, 1078-1090.
[10] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., … & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context[C]. Zurich, Switzerland: European Conference on Computer Vision (ECCV), 2014: 740-755.
[11] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN[C]. Venice, Italy: IEEE International Conference on Computer Vision (ICCV), 2017: 2980-2988.
[12] Liu Xiaonan, Wang Zhengping, He Yuntao, et al. A Review of Small Object Detection Research Based on Deep Learning[J]. Tactical Missile Technology, 2019(1):100-107.
[13] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition[J]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, 770-778.
[14] Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path aggregation network for instance segmentation[J]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, 8759-8768.
[15] Weilisi, Yu Zehui. Glass Bottle Mouth Defect Detection Based on YOLOv5[J]. Yangtze River Information Communication, 2023, 36(1): 9-11.
[16] Jiang Lei, Cui Yanrong. Small Object Detection Based on YOLOv5[J]. Computer Knowledge and Technology, 2021, 17(26): 131-133.
[17] Song Yuxuan, Zhao Yu, Zhang Jingying, et al. Design of Sitting Posture Monitoring System Based on YOLOv5[J]. Computer Knowledge and Technology, 2023, 19(8): 22-25.
[18] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection[J]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, 936-944.
[19] Shrivastava, A., Gupta, A., & Girshick, R. (2016). Training region-based object detectors with Online Hard Example Mining[J]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, 761-769.
[20] Shi Fei, Qiu Zhen, Han Qin, et al. Improved Faster R-CNN Algorithm Based on Variable Weight Loss Function and Hard Example Mining Module[J]. Computer and Modernization, 2020(8): 56-62.
[21] Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2018). Focal loss for dense object detection[J]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, 2980-2988.
[22] Huang Jian, Zhang Gang. A Review of Object Detection Algorithms Based on Deep Convolutional Neural Networks[J]. Computer Engineering and Applications, 2020, 56(17): 12-23.