Application of detection and tracking for equipments in open-pit mines based on YOLOv8n+DeepSORT technology

Li Ke[1], Byung-Won Min[2*]
[1]Ph.D. Student, Division of Information and Communication Convergence Engineering, Mokwon University
[2]Professor, Division of Information and Communication Convergence Engineering, Mokwon University

YOLOv8n+DeepSORT 기술 기반 이용한 노천광산 채굴 장비의 검출 및 추적 응용

이크[1], 민병원[2*]
[1]목원대학교 정보통신융합공학부 박사과정, [2]목원대학교 정보통신융합공학부 교수

Abstract To address the inefficiencies of visual interpretation for mining equipment supervision in open-pit mines, this study proposes an enhanced YOLOv8n+DeepSORT framework. By leveraging high-point surveillance for image acquisition and optimizing both YOLOv8n for equipment identification and DeepSORT for real-time tracking, we overcome limitations in accuracy, cost, and real-time monitoring. Field validation at Jingkai Runding Mine, Pingxiang, Jiangxi, demonstrates the technology's efficacy in identifying and tracking mining equipment, featuring rapid algorithm convergence, low computational overhead, and near-precise target detection and tracking. This approach paves the way for algorithmic support to facilitate effective government regulation of open-pit mining operations.

Key Words : Open-pit mine; YOLOv8n ; DeepSORT ; Object Detection; Target tracking.

요 약 중국의 대부분 노천광산은 여전히 시각적 해석을 주로 사용하고 있어, 시각적 해석은 시간이 많이 걸리고 비용이 많이 들며, 정확도는 높지만 효율성은 매우 낮아 노천광산의 채굴 장비를 실시간으로 모니터링할 수 없다. 이 문제를 해결하기 위해 이 논문에서는 향상된 YOLOv8n+DeepSORT 프레임워크를 제안한다. 고지점 모니터링을 통해 이미지 수집을 하고, 장비 인식을 위한 YOLOv8n과 실시간 추적을 위한 DeepSORT를 최적화하여 정밀도, 비용 및 실시간 모니터링 측면에서의 한계를 극복했다. 장시성 핑샹시의 경제개발구 런딩 광산에서의 현장 검증을 통해 이 기술이 채굴 장비 인식 및 추적에서의 기능을 입증하였으며, 알고리즘 수렴이 빠르고 계산 오버헤드가 낮으며, 목표 검출 및 추적의 정확도가 높은 특징을 가졌다. 향후 정부 규제 당국이 노천광산 감독을 잘 수행할 수 있도록 알고리즘 지원을 제공할 수 있다.

주제어 : 노천광산; YOLOv8n; DeepSORT; 목표 검출; 목표 추적

*교신저자 : 민병원(minfam@mokwon.ac.kr)
접수일: 2024년 08월 19일 수정일: 2024년 09월 10일 심사완료일: 2024년 10월 11일

1. Introduction

In China, open-pit mining of solid minerals predominantly utilizes large-scale engineering machinery, including trucks, excavators, bulldozers, and other critical equipment. However, most regulatory authorities in China primarily rely on visual interpretation via video surveillance and periodic audits to monitor opencast mines.

YOLO [1], short for You only look once, is a One-Stage target detection algorithm that combines accuracy as well as speed, making it the framework of choice for industrial applications. DeepSORT [2] is a deep learning based target tracking algorithm that uses Kalman filters for target tracking. Tracking these objects is combined with the use of DeepSORT algorithm for multi-target tracking.

In this paper, we propose the utilization of YOLOv8n+DeepSORT technology to address the challenges in mine supervision. Our method initiates with high-point surveillance to gather data on equipment within the mine. Subsequently, we employ the YOLOv8n algorithm to precisely detect mining equipment within the camera's field of view. This is followed by integration with the enhanced DeepSORT algorithm for real-time tracking of identified mining equipment. Lastly, we conduct experimental validation of the detected targets and analyze the results, ensuring the robustness and accuracy of our proposed approach. The method was validated in a field trial at the Jingkai Runding Mine in Pingxiang City, Jiangxi Province. A 4-minute video clip captured between 16:22:27 and 16:26:31 on January 23, 2023, was selected, and 24 real frames were extracted for testing. The results demonstrate that the proposed YOLOv8n+DeepSORT technology effectively achieves recognition and tracking of mining equipment, exhibiting advantages such as high algorithm convergence speed, low network computing overhead, and near-accurate target detection and tracking accuracy.

2. Related Technology Introduction

2.1 YOLOv8n algorithm

Object detection algorithms are one of the fundamental algorithms in the field of computer vision. Currently, popular object detection algorithms are mainly classified into two categories: two-stage algorithms and single-stage algorithms. Among them, end-to-end algorithms can directly output the category probabilities and coordinates of objects, resulting in faster speeds. Typical end-to-end algorithms include SSD [3], YOLO versions from v1 to v8 [4], DSSD [5], and Retina-Net [6]. In contrast, non-end-to-end algorithms generally require generating candidate regions first, followed by classification and regression. These algorithms offer higher detection accuracy but slower detection speeds. Typical non-end-to-end algorithms include R-CNN [7], Fast R-CNN [8], Faster R-CNN [9], and Mask R-CNN [10]. Although two-stage algorithms have certain advantages in detection accuracy compared to single-stage algorithms, their detection speeds are significantly lower. Therefore, single-stage detection algorithms are often adopted for real-time tasks.

The core idea of the YOLO algorithm is to transform the object detection problem into a regression problem by dividing the image into a fixed-size grid and predicting the bounding box and class probability of an object for each grid cell. The algorithm was first proposed by Joseph Redmon et al. in 2016, and the initial version was YOLOv1, which first introduced the concept of real-time target detection, but performed poorly on small targets and nearby objects detection. YOLOv2 [11] significantly enhanced its capabilities by incorporating the deeper neural network architecture of the Darknet-19 model. This advancement, coupled with the introduction of anchor boxes, led to improved detection performance for small objects, while simultaneously supporting multi-class detection. Furthermore, YOLOv3 [12] bolstered

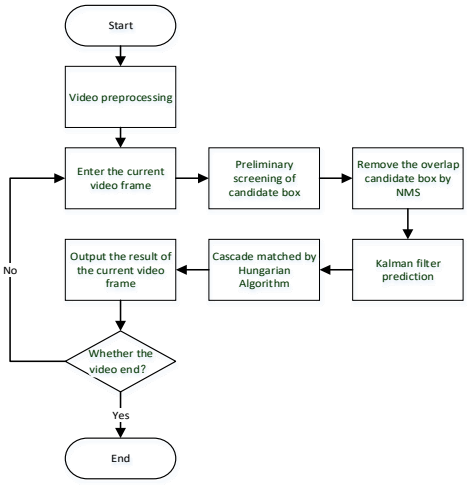
its versatility by integrating the Feature Pyramid Network (FPN) to adeptly handle multi-scale features. Additionally, it refined the bounding box regression and classification branches, resulting in even more accurate and robust detections. YOLOv4 [13] pioneered the integration of techniques such as CIOU, SAM, BottleneckCSP, PANet, and Darknet53, significantly advancing detection performance and speed. Additionally, it introduced hardware acceleration technologies to further elevate real-time capabilities. YOLOv5 [14], with its lightweight model design, enhanced real-time performance and incorporated automated hyperparameter optimization and model architecture search, offering support for diverse hardware and deployment environments. YOLOv6 [15] refined the detection accuracy by incorporating lightweight designs, model compression techniques, and improved training strategies. Lastly, YOLOv7 [16] introduced the scalable and efficient layer aggregation network E-ELAN, model structure reparameterization, and dynamic label assignment strategies, leading to substantial enhancements in the model's detection performance.

The YOLOv8 algorithm is the latest and most mainstream version in the classic single-stage YOLO algorithm series, surpassing all previous YOLO algorithms in both detection accuracy and speed. Based on the increase in network depth and width, it can be sequentially classified into five models: YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x. Considering the real-time requirements for detecting Open-pit mine equipment, this paper selects the YOLOv8n network model, which offers good accuracy, relatively low model complexity, and excellent computational speed, as the basic network model for experimental research.

2.2 DeepSORT algorithm

The advent of the SORT (Simple Online and Realtime Tracking) algorithm [17] has propelled detection-based tracking into the mainstream of multi-object tracking. Proposed by Bewley et al., the SORT algorithm combines Kalman filtering for trajectory prediction with the Hungarian algorithm for object matching, achieving remarkable performance in multi-object tracking tasks. Building upon the core principles of SORT, Wojke introduced the DeepSORT algorithm, which incorporates a re-identification network to extract distinctive features from objects. This network preserves the texture feature information of individual objects, effectively mitigating the issue of ID switches caused by occlusions. Furthermore, DeepSORT enhances the tracking accuracy by introducing cascaded matching and new track confirmation steps into the overall tracking pipeline, leveraging appearance information to refine the tracking process.

The tracking process of the DeepSORT algorithm for tracking targets is illustrated in Fig.1. The main algorithm can be summarized as follows: at each frame, the algorithm commences by employing a deep learning model to predict all potential objects. Subsequently, data association is performed based on appearance information. Finally, Kalman filtering is applied to update the trajectory information.



[Fig. 1] DeepSORT Algorithm Flow

2.3 Technology for high point panoramic video surveillance networking

Since 2022, China has witnessed the emergence of a high-point panoramic video surveillance networking technology [18]. This technology differs from ordinary image and video surveillance in that it focuses on achieving precise wide-area imagery from elevated positions. It is a comprehensive approach that balances wide-angle observation of large scenes with precise monitoring of specific targets. Ordinary video surveillance methods suffer from drawbacks such as obstructions by environmental objects, limited field of view, and poor visibility in adverse weather conditions. The high-point panoramic video surveillance networking technology effectively addresses these shortcomings.

This technology involves installing remote day-night video surveillance equipment (equipped with real-time computation, object tagging, and other functionalities) at heights of 50 to 100 meters or even higher, where there is a good view, to achieve video surveillance over a radius of tens of kilometers. Through 24/7 remote real-time monitoring, it performs image recognition functions for objects such as people, animals, and various types of vehicles. The video surveillance data used in this paper were captured and generated using this technology.

Currently, according to literature research, commonly used object detection and tracking algorithms are still focused on YOLOv5+SORT or YOLOv5+DeepSORT. The target objects studied include vehicles on highways, Hermetia illucens larvae, and others. Examples include "Traffic Monitoring Video Vehicle Volume Statistics Method Based on Improved YOLOv5s+Deep SORT" proposed by Li Yongshang et al. in 2022 [19], "Research on Vehicle Detection and Tracking AlgorithmBased on Improved YOLOv5 and DeepSort" proposed by You Xiaoyu in 2022 [20], "Research on road vehicle tracking methodsbased on YOLOv5 and DeepSort" proposed by Zhao Yufan in 2023 [21], and "Larvae of Black Soldier Fly Counting Based On Yolov5s Network and Improved SORT Algorithm" proposed by Zhao Xinlong in 2024 [22]. As of now, there is a lack of in-depth research on the latest YOLOv8+DeepSORT technology. Meanwhile, in the complex context of mining operations, there is scant research on object detection and recognition algorithms specifically for mining equipment, despite urgent demands from regulatory authorities in China. Considering these factors, the theoretical research and real-world application presented in this paper possess considerable research value.

3. Enhanced YOLOv8n

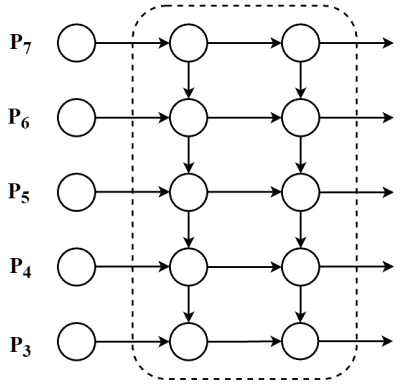
Despite its numerous advantages, the YOLOv8n model often exhibits less-than-ideal detection accuracy in practical, real-world environments. This limitation primarily stems from the inherent complexity of real-world scenarios, characterized by uneven lighting conditions, diverse obstacles, and numerous other factors that adversely affect object detection. To meet the stringent demands of real-time object detection in such environments, it becomes imperative to undertake appropriate modifications and enhancements to the YOLOv8n model.

3.1 Weighted Bidirectional Feature Pyramid Networks

In the realm of traditional object detection algorithms, a prevalent approach involves leveraging feature maps of varying sizes to detect targets across different scales. The authors of BiFPN (Bidirectional Feature Pyramid Network) argue that the methods of Feature Pyramid Network (FPN) and Path Aggregation Network (PANet) often treated all input features equally, disregarding the fact that features with varying resolutions contribute unequally to the fusion process.

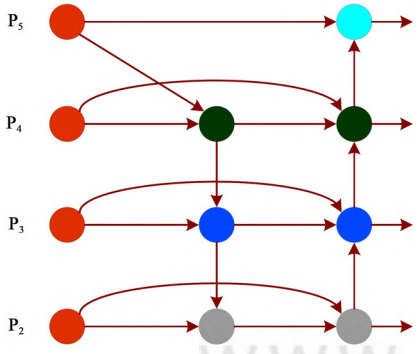
Recognizing this limitation, BiFPN introduces

a bidirectional feature fusion mechanism and a dynamic network architecture design, empowering the fusion process to be more comprehensive and adaptive. The model architecture of BiFPN is illustrated in Fig.2.



[Fig. 2] FPN model structure diagram

However, the underlying architecture of the BiFPN model exhibits a notable limitation in its handling of shallow features, as it tends to diminish their weight during feature fusion. This is particularly problematic given that shallow, high-resolution features are crucial for the detection of small objects. To address this issue and cater to the unique challenges of open-pit mining scenarios, we propose an enhancement to BiFPN by designing a four-level F-BiFPN (Feature-Enhanced BiFPN) structure to the detection of small objects in open-pit mine imagery. The proposed F-BiFPN structure, as depicted in Fig.3.



[Fig. 3] F-BiFPN model structure diagram

The calculation process takes the P4 feature fusion in Fig. 5 as an example, and the specific calculation formula is as follows:

$$P_4^{td} = \mathrm{Conv}\left(\frac{w_1 \cdot P_4^{\in} + w_2 \cdot \mathrm{Resize}(P_5^{\in})}{w_1 + w_2 + \theta}\right)$$
$$P_4^{out} = \mathrm{Conv}\left(\frac{w_1' \cdot P_4^{\in} + w_2' \cdot P_4^{td} + w_3' \cdot \mathrm{Resize}(P_3^{out})}{w_1' + w_2' + w_3' + \theta}\right) \quad (1.1)$$

In Equation (1.1), P_4^{td} is the intermediate feature of layer 4 of the top-down path, P_4^{out} is the output feature of layer 4 of the bottom-up path, w_i is the learnable weight, and θ is a very small parameter to prevent the value from being unstable with a value of 0.0001.

3.2 Enhanced C2f Module

In the complex environment of open-pit mines, a plethora of irrelevant information often arises, posing a significant challenge to model predictions by introducing distractions. To enhance the feature extraction capabilities under such complex backgrounds and effectively ignore irrelevant background information, this paper introduces the SimAM (A Simple, Parameter-Free Attention Module for Convolutional Neural Networks) into both the backbone network and feature fusion modules. By integrating spatial and channel-wise information, SimAM sets a unified 3D attention weight that encompasses both the channel and spatial dimensions, eliminating the need for additional feature generators. Instead, it directly derives the 3D attention weights, thereby strengthening the features of mining equipment while attenuating irrelevant and distracting features.

SimAM evaluates each neuron in a neural network by measuring the linear separability between neurons. As shown in Eqs. (1.2)-(1.5), t is the target neuron, x is the neighboring neurons, e_t^* is the hyperparameter, and λ is the energy The lower the energy of a certain neuron,

which indicates that the neuron is more important the more the information difference between that neuron and the surrounding neurons. Based on the importance of the neuron, the neurons are weighted by $\frac{1}{e_t^*}$.

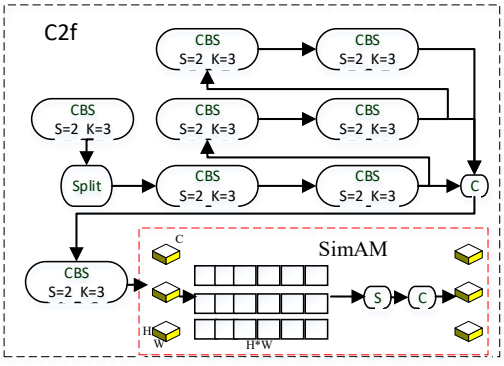
$$e_t^* = \frac{4(\hat{\sigma}^2+\lambda)}{(t-\hat{u})+2\sigma^2+2\lambda} \quad (1.2)$$

$$u_t = -\frac{1}{M}\sum_{i=1}^{M-1} X_i \quad (1.3)$$

$$\sigma^{2_t} = \frac{1}{M-1}\sum_{i=1}^{M-1}(X_i-u_t)^2 \quad (1.4)$$

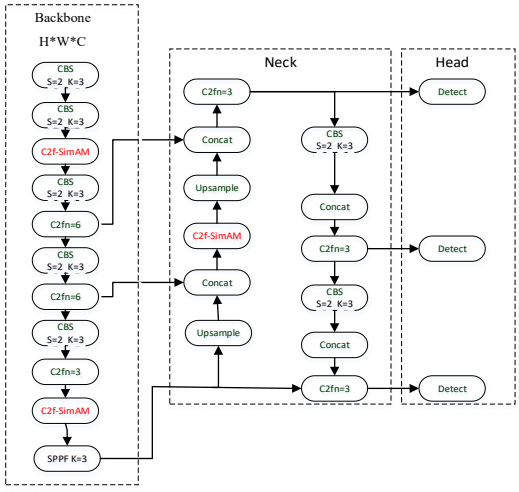
$$\widetilde{X} = sigmoid(\frac{1}{E})\odot X \quad (1.5)$$

The SimAM (Simple Attention Mechanism) is a parameter-free attention module that does not require the incorporation of contextual channel structures or auxiliary modules, thereby eliminating the introduction of additional parameters. In this paper, we propose a novel integration of the C2f module from the YOLOv8n model with the SimAM module, resulting in the C2f-SimAM module. The architecture of this combined module is illustrated in Fig.4, showcasing an elegant blend of computer vision techniques.



[Fig. 4] C2f-SimAM Structure Chart

Incorporating the C2f-SimAM module within the YOLOv8n model's architecture, specifically after the second Conv2d of the backbone network, prior to the SPPF (Spatial Pyramid Pooling Fast) layer, and subsequent to the first Upsample module, results in an enhanced network configuration as depicted in Fig.5 (highlighted in red font).

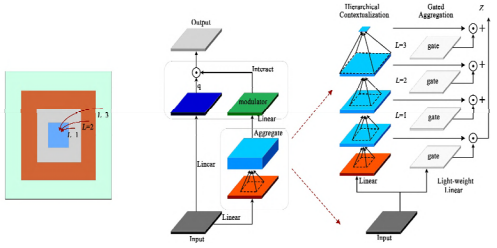


[Fig. 5] YOLOv8n Network Structure After Adding SimAM

3.3 Replacement of the SPPF module

The SPPF module, by utilizing multiple small-sized pooling operations, generates diverse feature maps that are then fused through a cascading process, achieving a harmonious integration of local and global features. However, this interaction paradigm suffers from certain limitations: the spatial information obtained is inadequate, and it fails to accurately discern the varying importance of semantic features in object detection tasks.

To address these shortcomings, we introduce the Focal Modulation (FM) module as a replacement for the SPPF module. Compared to the SPPF module, the aggregation mechanism employed by the FM module yields richer spatial information. Its dynamically adaptive fusion methodology is more efficient, enhancing both the accuracy and robustness of detection, thereby compensating for the precision challenges inherent in YOLOv8n. Fig.6 illustrates the architecture and schematic diagram of the FM module, showcasing its innovative design.



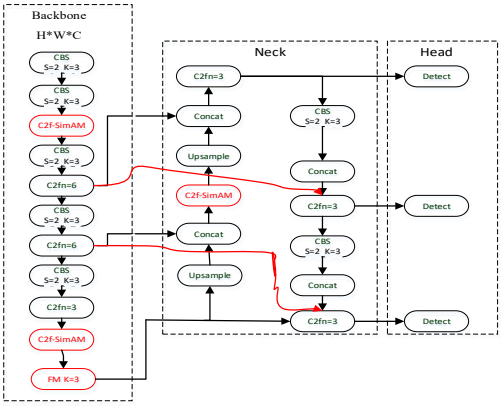
[Fig. 6] Focal Modulation Structure and Schematic

The FM (Focal Modulation) module condenses contextual features from varying granularity levels into a single feature vector through gated aggregation. The overarching focal modulation formula is formulated as follows:

$$y_i = q(x_i) \odot \mathrm{h}\left(\sum_{l=1}^{L+1} g_i^{'}, z_i^{'}\right) \qquad (1.6)$$

In Eq. (1.6), q() is a query mapping function, ⊙ is the per-element multiplication operator, and h() is the modulation function. g^{l_i} and z_i^l are the perceptual weights and corresponding visual features of the lth level at position i, respectively.

In summary, three primary modifications have been made to the YOLOv8n model in this study. Firstly, the feature network has been upgraded to a Weighted Bi-directional Feature Pyramid Network (BIFPN). Secondly, the C2f module has been enhanced by the introduction of the SimAM module. Lastly, the SPPF module has been substituted with the FM module. The overall architecture of the improved YOLOv8n model is depicted in Fig.7, which incorporates the modified four-level F-BiFPN. Consequently, the original SPPF configuration with K=5 (P3-P7) has been revised to FM with K=3 (P3-P5), reflecting the adjustment in feature pyramid levels. Notably, YOLOv8 inherently incorporates the P2 level, therefore, no separate computation is required for the P2 layer.



[Fig. 7] The structure of the improved YOLOv8n model

4. Enhanced DeepSORT algorithm

While DeepSORT exhibits commendable performance in dynamic object tracking, it is not without limitations. In complex mining scenarios, occlusions and intersections between dynamic objects frequently lead to tracking failures. Relying solely on the basic Intersection over Union (IoU) matching proves insufficient to meet the stringent requirements for accurate tracking and matching. Consequently, solely utilizing the IoU of detection and track bounding boxes as the sole criterion for association may be inadequate.

To address this limitation, this section introduces enhancements to the IoU matching mechanism. We incorporate target motion information into the IoU matching process and calculate the similarity between tracks and detections based on their Euclidean distance. By fusing this similarity measure with the improved IoU matching, we aim to elevate the accuracy of the matching process.

Let the trajectory set $T=(t_1,...t_i,...t_n)$, the detection set $D=(d_1,...d_j,...d_m)$, where t^i and d_j are state vectors, and each state vector contains position, velocity, and direction information.IoU matching can be considered to introduce the position, velocity, and direction factors to compute their corresponding distances, respectively.The position distance can be computed by the IoU with the following expression.

$$d_c(i,j) = 1 - IoU(C_i, C_j) \quad (1.7)$$

In Equation (1.7), $d_c(i,j)$ represents the positional distance; C_i、C_j represent the positions of the detection frame and the trajectory frame, respectively.

The distance in the velocity direction can be calculated by the relative velocity of the detection frame and the center point of the trajectory frame with the following formula.

$$d_v(i,j) = \frac{| v_i - v_j |}{\sqrt{(w^2 + h^2)}} \quad (1.8)$$

In Equation (1.8), $d_v(i,j)$ is the velocity distance; V_i 、V_j are the relative velocities of the detection and trajectory, respectively; w and h are the width and height of the two bounding boxes.

The distance in the direction of motion is calculated by the angular difference between the detection frame and the center point of the trajectory frame in the direction with the following equation.

$$d_\theta(i,j) = 1 - \cos(\theta_i, \theta_j) \quad (1.9)$$

In Equation (1.9), $d_\theta(i,j)$ is the direction distance; θ_i、θ_j are divided to indicate the direction of motion of the detection and trajectory.

The weighted summation of the above position distance, velocity distance, and direction distance leads to the improved IOU matching with the following equation:

$$IoU(t_i, t_j) = W_c d_c(i,j) + W_v d_v(i,j) + W_\theta d_\theta(i,j) \quad (2.0)$$

In Equation (2.0), W_c、W_v、W_θ are the weights occupied by position, velocity, and direction distances, respectively.

The Euclidean distance can be calculated based on the state vector of detection and trajectory with the following equation:

$$d(t_i, d_j) = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m}(t_i - d_j)^2} \quad (2.1)$$

Normalizing this value yields a similarity score between the detection and the track, as formulated below :

$$s(t_i, d_j) = \frac{1}{1 + d(t_i, d_j)} \quad (2.2)$$

By combining the improved IoU matching with the similarity score, a final association score is derived through weighted summation, as formulated below:

$$s(t_i, d_j) = w_s s(t_i, d_j) + w_{IoU} IoU(t_i, d_j) \quad (2.3)$$

The value $s(t_i, d_j)$ computed from Equation (2.3) is compared against a predefined threshold. If $s(t_i, d_j)$ exceeds this threshold, the detection result and track object are jointly fed into the Hungarian algorithm for optimal assignment. Otherwise, they are not associated and considered as invalid targets.

5. Testing, Validation, and Analysis

5.1 Evaluation indicators

For the target detection task, the evaluation metrics can objectively measure the model's fit to the data and generalization ability. Therefore, in this paper, Precision, Recall, and mAP are chosen as the evaluation metrics for the YOLOv8n improved model.

Precision is mainly used to measure the percentage of correctly predicted targets, and a high precision rate can reduce the probability of false detection, and its calculation formula is shown below:

$$P = \frac{TP}{TP + FP} \quad (2.4)$$

In Equation (2.4), TP (True Positive) represents the number of correctly predicted positive samples by the model, i.e., the model correctly identifies actual positive samples as positive. FP (False Positive), on the other hand, denotes the number of incorrectly predicted positive samples, where the model mistakenly classifies negative samples as positive.

Recall, also known as the true positive rate, measures the proportion of actual positive samples that are correctly predicted as such. It is calculated using the following formula:

$$P = \frac{TP}{TP + FN} \tag{2.5}$$

In Equation (2.5), FN (False Nagative) denotes the number of negative samples that are incorrectly determined as positive categories.

The formula for calculating the average accuracy of n categories is shown below:

$$mAP = \frac{1}{n}\sum_{i=0}^{n}\int_{0}^{1} P(R)dR \tag{2.6}$$

mAP@50 refers to the mean Average Precision of a model when the IoU (Intersection over Union) threshold is set to 0.5. On the other hand, mAP@50-95 signifies the average of the mean Average Precisions computed across a range of IoU thresholds, from 0.5 to 0.95, with a step size of 0.05.

For the target tracking task, there are 2 key evaluation metrics used to measure the performance of the algorithm.

MOTA (Multiple Object Tracking Accuracy) is a comprehensive evaluation metric most commonly used in multi-target tracking tasks to measure the combined metrics of error tracking, missed tracking to ID switching output.

$$\mathrm{MOTA} = 1 - \frac{\sum^{t}\left(\mathrm{FN_t} + \mathrm{FP_t} + \mathrm{IDSW_t}\right)}{\sum \mathrm{GT_t}} \tag{2.7}$$

In Equation (2.7), FN_t is the number of missed tracking targets in frame t, FP_t is the number of incorrectly matched tracking targets in frame t, $IDSW_t$ is the number of tracking target ID switches in frame t, and GT_t is the number of true annotations in frame t.

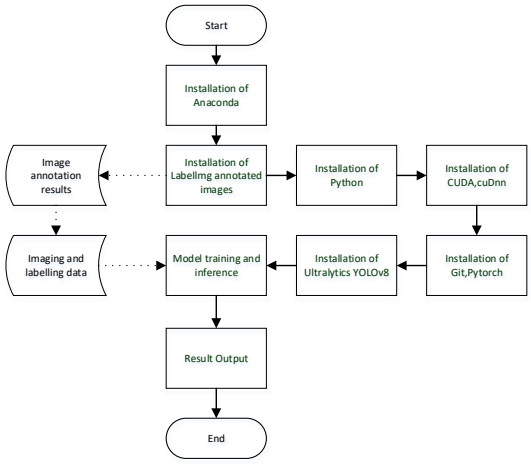
MOTP (Multiple Object Tracking Precision) is the Mean Orientation Error, which is used to measure how well the tracking results match the true value.

$$\mathrm{MOTP} = \frac{1}{\sum_{i=1}^{N} \mathrm{m_i}} \sum_{i=1}^{N}\sum_{j=1}^{\mathrm{m_i}} \mathrm{d_{i,j}} \tag{2.8}$$

In Equation (2.8), M_i is the number of correctly tracked targets in frame i, and $d_{i,j}$ is the distance metric between the jth tracked predicted target in frame i and the actual detected target that was successfully matched.

5.2 Experimental Installation Environment

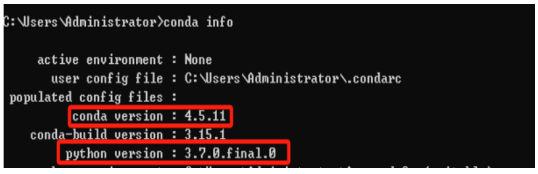
The required environment deployment and test flow for this experiment is shown below:



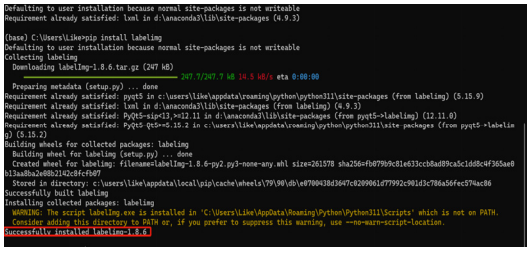
[Fig. 8] Environment Deployment and Testing Process

The model training and testing in this study were run in the same environment on a computer with an NVIDIA GeForce RTX 3070 GPU, a 13th generation

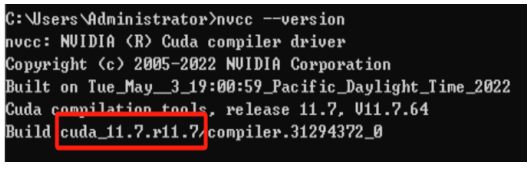
Core i7 RTX4060Ti processor, and Windows 11 as the operating system, using Anaconda version 4.5, Python version 3.7, Labellmg version 1.8, CUDA version 11.7, cuDnn version 8.6, and PyTorch version 2.1. Among them, the installation diagrams for Anaconda, Python, Labellmg, CUDA, cuDnn, and PyTorch are shown in 9, 10, 11, 12, 13, and 14.



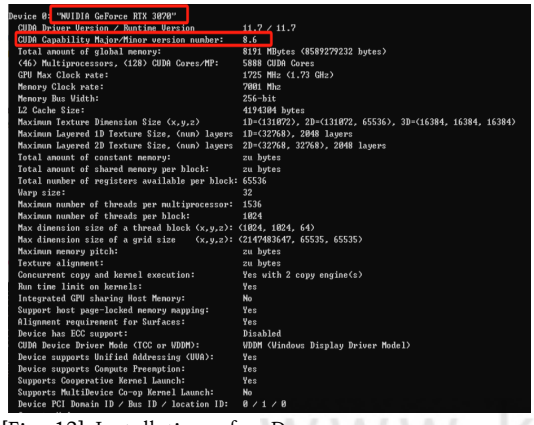
[Fig. 9] Installation of Anaconda and Python



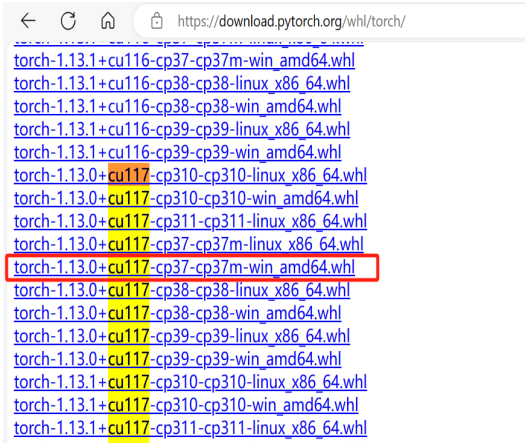
[Fig. 10] Installation of Labellmg



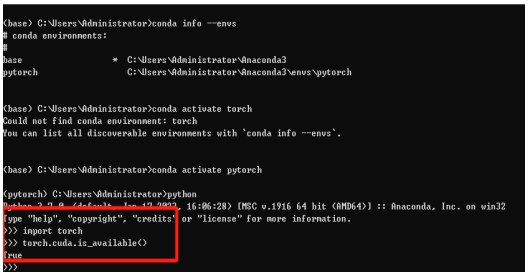
[Fig. 11] Installation of CUDA



[Fig. 12] Installation of cuDnn

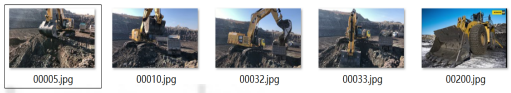


[Fig. 13] Installation of Torch



[Fig. 14] Installation of PyTorch

The dataset utilized for experimental testing in this study originates from Kaggle, one of the most popular and sought-after platforms in the realm of data science. Specifically, a mining excavation dataset was employed, where all videos were segmented into frames and subsequently converted into images. Out of the 12,000 frames, 300 pre-processed and authentic images depicting open-pit mining activities were selected. These images were standardized in the JPEG format and named using a six-digit numbering system (e.g., 000001.jpg). The target and track information were saved in CSV format for ease of access and processing. A selection of images from the experimental dataset is presented in Figure 15.

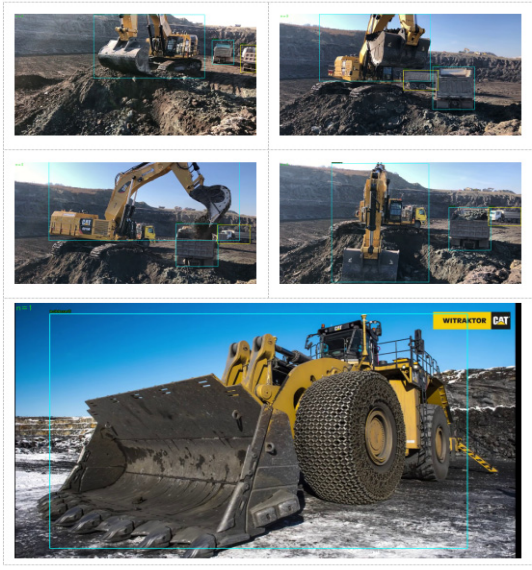


[Fig. 15] Selected images from the dataset experiment

5.3 Experimental process and result analysis

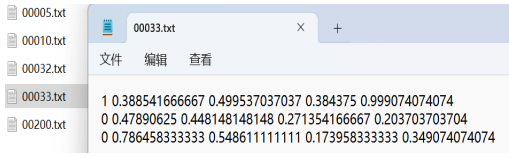
The experiment consists of four steps: image annotation, format conversion, configuration file modification, and image training.

Firstly, we commence by performing image annotation utilizing the LabelImg tool, a graphical image annotation software. This step involves marking the target classes present in each image from our experimental dataset. An illustrative example of annotated images from the dataset is depicted in Fig.16.



[Fig. 16] Image labelling work

Secondly, we convert the annotated results from their native XML format to the YOLO-compatible TXT format. This process involves extracting the bounding box (bbox) information from each XML annotation and saving it in a TXT file, with each image corresponding to a unique TXT file. The format of these TXT files is as shown in Fig.17, where each line represents information about a single target, including class, x_center, y_center, width, and height.



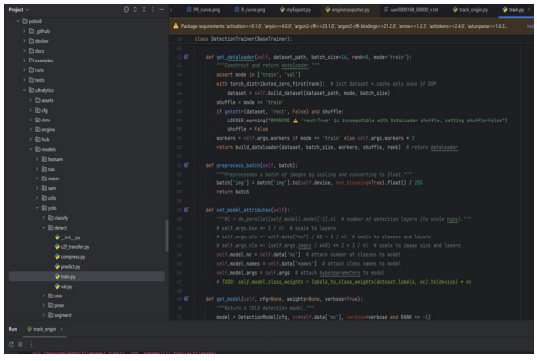
[Fig. 17] Image format conversion

Next, we proceed by creating a new data.yaml file within the mydata folder. This file serves as a repository for the training and validation set split files (train.txt and val.txt). After setting up the environment within the sicisp_2023 folder, we initiate the training process by executing a command that specifies the number of target classes, their specific names, and other relevant parameters. This step is illustrated in Fig.18.

```
Yolo detect train data=Mining Equipment/data/data.yaml
model=sicisp2023/ultralytics/ultralytics/weights/yolov8n.pt
# Here the ultralytics/weights use pre-trained weights
epochs=200
imgsz=640 batch=16 workers=32
```

[Fig. 18] Configuration file changes

Lastly, to commence the training process, we open the Anaconda Prompt and launch the train.py script, as depicted in Fig.19. This script orchestrates the training of our model using the prepared dataset and configuration settings.

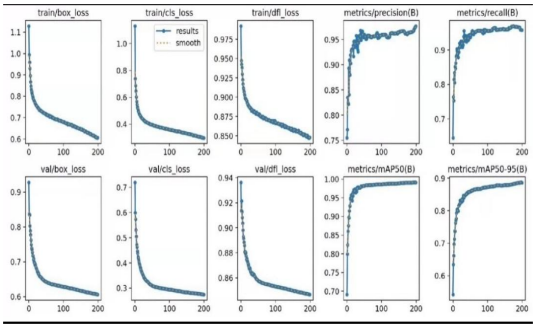


[Fig. 19] Training dataset

Upon completion of the training phase, we obtain the convergence curves, showcasing the training outcomes. As depicted in Fig.20, the first row displays the training loss, precision, and

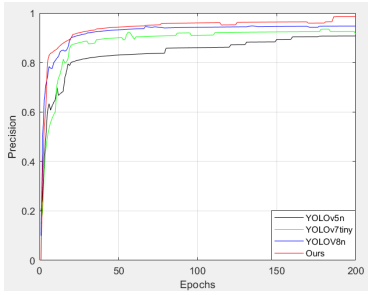
recall metrics, while the second row presents their respective values during validation. Notably, the modified YOLOv8n model exhibits enhanced performance across all evaluated metrics compared to its baseline counterpart. Specifically, Precision improves from 94.2% to 96.2%, Recall rises from 93.8% to 96.6%, mAP@50 increases from 95.6% to 97.8%, and mAP@50-95 advances from 85.6% to 87.4%.

Observing the loss curves for both the training and validation sets, we find that the loss values remain consistently low, indicating stable learning. After 200 epochs, the validation set demonstrates a notable reduction in boundary box loss, classification loss, and objectness loss, resulting in superior precision and recall rates and a smoother convergence curve. Overall, the modified YOLOv8n model exhibits excellent convergence characteristics, underscoring its effectiveness for object detection tasks.

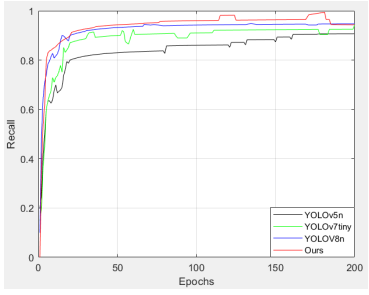


[Fig. 20] Training results of the improved model

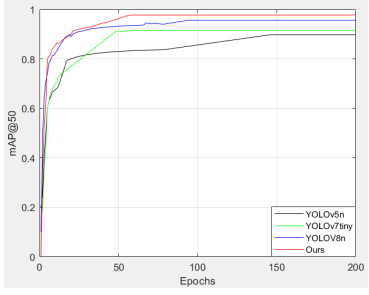
To compare the performance of different object detection algorithms, this paper conducted comparative experiments using representative target detection algorithms, namely YOLOv5n, YOLOv7tiny, and YOLOv8n, on the dataset employed. The final experimental results are presented in Figures 21, 22, 23, and 24.



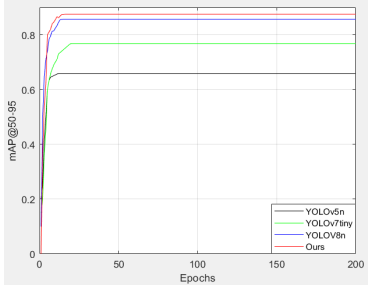
[Fig. 21] Changes in Precision index for comparative experiments



[Fig. 22] Changes in Recall index for comparative experiments



[Fig. 23] Changes in mAP@50 index for comparative experiments



[Fig. 24] Changes in mAP@50-95 index for comparative experiments

As evident from the above figures, on the dataset utilized in this study, the modified YOLOv8n algorithm outperforms both YOLOv5n and YOLOv7tiny from the YOLO family, exhibiting the highest detection accuracy, and surpassing other object detection models across various evaluation metrics. Consequently, the proposed algorithm in this paper effectively fulfills the requirements of object detection in open-pit mining scenarios.

To investigate the individual impacts of different enhanced modules on the YOLOv8n model, an ablation study was conducted. The results of this ablation study are summarized in Table 2. Here, FPS (Frames Per Second) denotes the number of images the model can process per second. A model is considered suitable for real-time detection systems when its prediction speed exceeds 30 FPS. FLOPs (Floating Point Operations) represent the number of floating-point arithmetic operations required by the model, with a lower FLOPs value indicating less computational complexity and faster runtime. Params, on the other hand, reflect the model's complexity and capacity, with more parameters enabling the fitting of more intricate data patterns.

Through experimental comparisons, it is evident that the F-BiFPN structure, modified based on BiFPN, enhances the utilization of shallow features and reduces computational time.

〈Table 1〉 Comparison of ablation experiments

Models	Precision	Recall	mAP@50	mAP@50-90	FPS	FLOPs	Params
YOLOv8n	94.2	93.8	95.6	85.6	31	8.1	3
YOLOv8n+F-BiFPN	94.6	91.8	95.1	78.6	42	5	2.6
YOLOv8n+SimAM	95.6	94.2	93.7	87.3	28	8.2	2.9
YOLOv8n+FM	96	97.1	98.1	88.1	30	8.6	3.1
YOLOv8n+F-BiFPN+SimAM	94.8	93.4	94.2	83.5	34	5	2.7
YOLOv8n+F-BiFPN+FM	95.1	96.7	96.3	85.7	36	5	3
YOLOv8n+SimAM+FM	96	96.3	96.5	87.6	29	8.7	3.1
This paper	96.2	96.6	97.8	87.4	32	5	2.71

The integration of the SimAM module further enables a focus on regions of interest while ignoring irrelevant background information, thus reducing parameters. Additionally, it strengthens the output of neurons related to target locations, while suppressing distractions from neurons unrelated to target positions, effectively improving the network's capability for localizing targets and enhancing small object detection performance. While the FM module improves detection accuracy and robustness, it slightly increases computational time, significantly enhancing the model's feature extraction capabilities and detection precision. However, as shown in Table 2, there are still limitations compared to other models, with the average precision not being the highest at an IoU threshold of 0.5 and a relatively low recall rate. Ultimately, the improved model, incorporating the advantages of all aforementioned modules, meets the precision and speed requirements for object detection in open-pit mining scenarios, outperforming the original YOLOv8n model.

Regarding target tracking tasks, a comparative evaluation of the enhanced algorithm against SORT and DeepSORT tracking algorithms was conducted, with the results presented in Table 3. Our modified DeepSORT algorithm demonstrates superior tracking performance in terms of MOTA (Multiple Object Tracking Accuracy) and MOTP (Multiple Object Tracking Precision). The original DeepSORT algorithm suffered from a lower MOTA due to frequent ID switches, but the improvements yield a 14.5% increase in MOTA. By incorporating target motion state information into the IOU matching module, the modified DeepSORT algorithm enhances the correlation between predicted and true values for moving targets, resulting in an 11.8% increase in MOTP. Overall, the improved DeepSORT algorithm achieves MOTA and MOTP scores of approximately 80%, indicating excellent tracking performance.

〈Table 2〉 Performance comparison of DeepSORT algorithm before and after improvement

Models	MOTA	MOTP
SORT	0.49	0.62
DeepSORT	0.69	0.76
This paper	0.79	0.85

In the experimental setup, the visualization of a subset of the dataset, as depicted in Figure 32, showcases the remarkable tracking performance of the improved algorithm based on YOLOv8n and DeepSORT. This enhanced tracking capability effectively mitigates issues related to ID switches, thereby not only validating the efficacy of the modified algorithm but also providing a more stable and reliable performance for practical applications.

[Fig. 25] Experimental detection of the effect of the model

5.4 Real Case Application and Verification

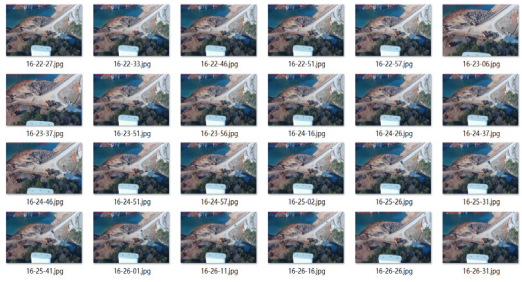
Utilizing the Jingkai Runding Mine in Pingxiang, Jiangxi, as a real-world case study of an open-pit mine, we employed nearby high-elevation surveillance points for monitoring [23], as illustrated in Fig.26. The video data captured is transmitted through a dedicated internet connection to the system's database for temporary storage. The database design, as illustrated in Fig.27, primarily comprises fields for device information, surveillance camera details, and event alert information.

[Fig. 26] High Point Surveillance Points

表名	对象 ID	所有者	表空间	行数估计
rt_device_extends	21,938,286	iland_ilanddb_us	pg_default	0
spatial_ref_sys	21,937,017	iland_ilanddb_us	pg_default	5,757
t_alarm	21,938,292	iland_ilanddb_us	pg_default	559
t_alarm_ext	21,938,308	iland_ilanddb_us	pg_default	9
t_alarm_history	21,938,314	iland_ilanddb_us	pg_default	0
t_alarm_level	21,938,321	iland_ilanddb_us	pg_default	45
t_alarm_process	21,938,326	iland_ilanddb_us	pg_default	2,041
t_alarm_repeat	21,938,333	iland_ilanddb_us	pg_default	0
t_alarm_rule	21,938,340	iland_ilanddb_us	pg_default	4,010
t_alarm_statistics	21,938,343	iland_ilanddb_us	pg_default	72,232
t_alarm_strategy	21,938,354	iland_ilanddb_us	pg_default	7
t_alarm_type	21,938,360	iland_ilanddb_us	pg_default	28
t_app_message_log	21,938,367	iland_ilanddb_us	pg_default	0
t_camera	21,938,374	iland_ilanddb_us	pg_default	200
t_camera_rule	21,938,380	iland_ilanddb_us	pg_default	0

[Fig. 27] Database design

This paper selects a 4-minute video clip from 16:22:27 to 16:26:31 on January 23, 2023. The system captures 24 consecutive, authentic images for validation, showcasing multiple targets such as trucks, excavators, bulldozers, and other primary mining equipment commonly found in open-pit mines, as depicted in Figure 28.

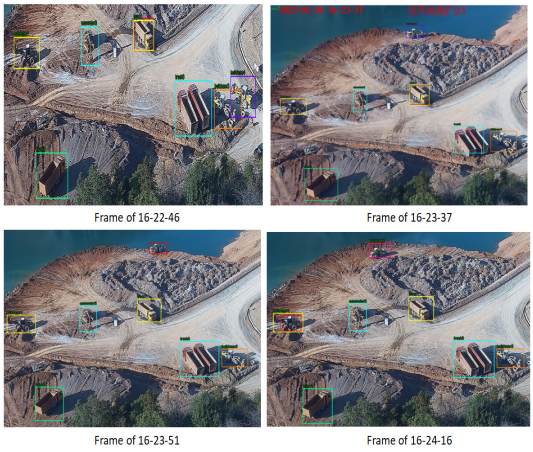


[Fig. 28] Monitoring the real picture of "Dinrun Mining" in the video clip

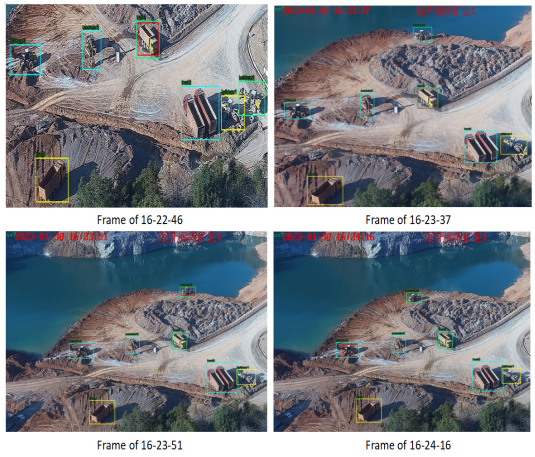
Figures 29 and 30 present the test results of the original YOLOv8n+DeepSORT algorithm and the improved combined algorithm, respectively, on real-world data. For comparison, frames 16-22-4, 16-23-37, 16-23-51, and 16-24-16 are selected. As observed in Figure 35, the bulldozer

(marked with a purple trajectory box) is assigned ID 4 in frames 16-22-4 and 16-23-37, but due to the increased distance, it is lost in frame 16-23-51 and displayed as a red detection box awaiting re-identification, with the ID disappearing. It is re-identified as a bulldozer in frame 16-24-16 with a new ID of 7. Upon reviewing the video, it is evident that this is the same bulldozer and not a new target, highlighting the issue of ID switching caused by environmental factors in the original YOLOv8n+DeepSORT algorithm.

In contrast, Figure 30 demonstrates that the same bulldozer (marked with a green trajectory box) maintains its ID (bulldozer2) consistently throughout the four moments of motion. Additionally, the numbering of the same type of equipment is continuous, starting from 0. As seen in Figure 35, there are a total of three bulldozers identified as bulldozer0, bulldozer1, and bulldozer2. The truck is recognized as truck0, and the excavator is recognized as excavator0. This continuity and consistency in ID assignment demonstrate the improved performance of the combined algorithm in mitigating ID switching issues.



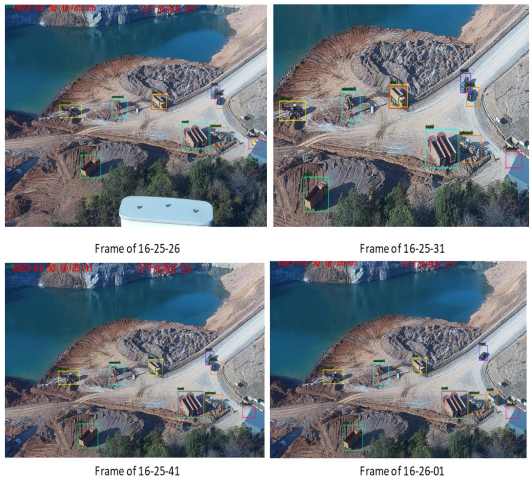
[Fig. 29] YOLOv8n+DeepSORT algorithm detection and tracking results



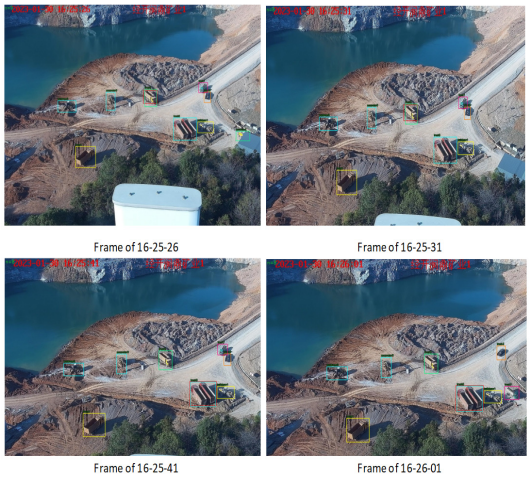
[Fig. 30] Improved algorithm detection and tracking results

Figures 31 and 32 present another set of test results on real-world data for the original YOLOv8n +DeepSORT algorithm and the improved combined algorithm, respectively, comparing frames 16-25-26, 16-25-31, 16-25-41, and 16-26-01. As seen in Figure 37, the bulldozer with ID 7 in frame 16-25-26 retains its trajectory box in subsequent frames 16-25-31, 16-25-41, and 16-26-01 even after it disappears from the scene. This issue results in the inability to recycle ID 7 for a new bulldozer type if it appears, revealing a shortcoming in the original DeepSORT algorithm's cascade update mechanism.

In contrast, Figure 32 shows that when the bulldozer with ID 7 disappears from frame 16-25-26, its trajectory box promptly vanishes in frame 16-25-31. Furthermore, even when truck3 and truck4 overlap in frame 16-25-41, their IDs remain unchanged in frame 16-26-01. This demonstrates that the improved model effectively handles occlusion and overlap scenarios, maintaining a high Multiple Object Tracking Precision (MOTP).



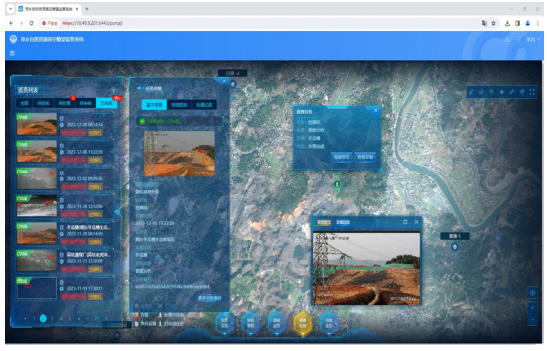
[Fig. 31] YOLOv8n+DeepSORT algorithm detection and tracking results



[Fig. 32] Improved algorithm detection and tracking results

As evidenced by the aforementioned real-world application cases, experimental validation has shown that the improved YOLOv8n+DeepSORT algorithm exhibits superior tracking performance and enhanced robustness. It effectively mitigates the issue of ID switching caused by factors such as occlusion and varying target distances. Furthermore, the algorithm successfully tracks multiple targets simultaneously without losing any, thereby resolving the challenge of dynamic target ID transformation during tracking.

We have also coded and implemented the improved YOLOv8n+DeepSORT algorithm in an early warning system for over-boundary mining in open-pit mine, with the application system interface shown in Fi.33. Regulatory authorities have leveraged this early warning system to effectively enhance their capability for early detection of over-boundary mining in opencast mines, significantly reducing the human costs associated with monitoring such activities.



[Fig. 33] Implementation of Algorithm YOLOv8n+ DeepSORT for Early Warning Applications in Open-pit mine

6. Conclusion

In this research, we present novel enhancements to the YOLOv8n model, commencing with the refinement of the BiFPN structure. Specifically, we introduce a custom four-level F-BiFPN architecture tailored to the unique challenges of open-pit mine environments, significantly enhancing the exploitation of shallow features. Secondly, we integrate the SimAM module, which elevates the network's emphasis on the region of interest by selectively amplifying or suppressing pertinent features based on their spatial relevance to the target location. This refinement bolsters the network's positioning prowess, yielding improved performance in detecting small targets. Furthermore, we substitute the SPPF module with the FM module, adopting a dynamically adaptive

fusion approach that promotes more efficient detection processes. This substitution not only amplifies the model's accuracy and robustness but also significantly augments its feature extraction capabilities and detection precision.

Additionally, we enhance the DeepSORT algorithm by incorporating target motion information into the IoU matching framework. By computing the Euclidean distance between tracks and detections and fusing this information with the modified IoU matching, we achieve a notable improvement in tracking accuracy. Experimental evaluations and field deployments at the Jingkai Runding Mine in Pingxiang, Jiangxi Province, conclusively demonstrate the efficacy of our YOLOv8n+DeepSORT technology in accurately recognizing and tracking mining equipment. The technology boasts rapid algorithm convergence, minimal network computational requirements, and precise target detection and tracking capabilities.

REFERENCES

[1] Redmon J, Divvala S and Girshick R, "You only look once: Unified, real-time object detection", in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp.779-788. Dissertations.

[2] Wojke N,Bewley A and Paulus D, "Simple online and realtim e tracking with a deep association metric", in Proceedings of the International Conference on Image Processing(ICIP), 2017, pp.3645-3649. Dissertations.

[3] Qiong W and Sheng-bin L, "Single Shot MultiBox Detector for Vehicles and Pedestrians Detection and Classification", in Proceedings of 2017 2nd International Seminar on Applied Physics, Optoelectronics and Photonics (APOP 2017), 2017, pp.36-42. Dissertations.

[4] Shi Y ,Li S , and Liu Z, "MTP-YOLO: You Only Look Once Based Maritime Tiny Person Detector for Emergency Rescue", Journal of Marine Science and Engineering, Vol.12, No.4, pp.669-784, 2024.

[5] Fu C Y, Liu W and Ranga A, "Deconvolutional single shot detector", arXiv preprint arXiv:1701.06659, 2017.

[6] Lin T Y, Goyal P and Girshick R, "Focal loss for dense object detection", in Proceedings of the IEEE international conference on computer vision, 2017, pp.2980-2988. Dissertations.

[7] Girshick R,Donahue J and Darrell T, "Rich feature hierarchies for accurate object detection and semantic segmentation", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp.580-587. Dissertations.

[8] Girshick R, "Fast r-cnn", in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp.1440-1448. Dissertations.

[9] Ren S, He K and Girshick R, "Faster r-cnn: Towards real-time object detection with region proposal networks", in Proceedings of the Advances in neural information processing systems, 2015, pp.28. Dissertations.

[10] He K, Gkioxari G and Dollár P, "Maskr-cnn", in Proceedings of the IEEE international conference on computer vision, 2017, pp.2961-2969. Dissertations.

[11] Redmon J and Farhadi A, "YOLO9000: better, faster, stronger", in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp.7263-7271. Dissertations.

[12] Redmon J and Farhadi A, "Yolov3: An incremental improvement", arXiv preprint, 2018.

[13] Bochkovskiy A, Wang C Y and Liao H Y M, "Yolov4: Optimal speed and accuracy of object detection", arXiv preprint, 2020 .

[14] Zhu X, Lyu S and Wang X, "TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios", in Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp.2778-2788. Dissertations.

[15] Li C, Li L and Jiang H, "YOLOv6: A single-stage object detection framework for industrial applications", arXiv preprint, 2022.

[16] Wang C Y, Bochkovskiy A and 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, 2023, pp.7464-7475. Dissertations.

[17] BEWLEY A, GE Z Y and OTT L, "Simple online and realtime tracking",in 2016 IEEE International Conference on Image Processing (ICIP), 2016, pp.3464-3468. Dissertations.

[18] China Security and Protection Industry Association, Technical requirements for high point panoramic video surveillance networking, China, 2024.

[19] Yongshang L, "Traffic Monitoring Video Vehicle Volume Statistics Method Based on Improved YOLOv5s+Deep SORT", Computer Engineering and Applications, Vol.58, No.5, pp.271-279, 2022.

[20] Xiaoyu Y, "Research on Vehicle Detection and Tracking AlgorithmBased on Improved YOLOv5 and DeepSort", Ph.D. Dissertation, Chang'an university, China, 2022.

[21] Yufan Z, "Research on road vehicle tracking methodsbased on YOLOv5 and DeepSort", Ph.D. Dissertation, Northwest Normal University, China, 2023.

[22] Xinlong Z, "Larvae of Black Soldier Fly Counting Based On Yolov5s Network and Improved SORT Algorithm", Transactions of the Chinese Society for Agricultural Machinery, Vol.1, No.1, pp.1-12, 2024.

[23] Li Ke and Byung Won Min, "Design and Implementation of Early Warning Monitoring System for Cross-border Mining in Open-pit Mines" Internet of Things Society, Vol.10, No.2, pp.25-41, 2024

이 크(Li Ke) [정회원]

- June 2013, East China Jiaotong University School of Software (Bachelor of Engineering)
- June 2016, East China Jiaotong University School of Software (Master of Engineering)
- January 2022 - Present, has been pursuing Ph.D. in Intelligent Fusion in IT at Mokwon Universit (Student)

〈관심분야〉

Information theory and their applications,Image recognition and analysis

민 병 원(Byung-Won Min) [정회원]

- He received M.S. degree in computer software from Chungang University, Seoul, Korea in 2005.
- He received Ph.D. degree in the dept. of Information and Communication Engineering, Mokwon University, Daejeon, Korea, in 2010.
- He is currently a professor of Mokwon University since 2010.

〈관심분야〉

digital communication systems, Big Data