

Landslide Detection Method Based on Lightweight Convolution and Attention Mechanisms

Cong Chen¹, Chengyang Zhang², Ran Chen³, Jiyang YU^{4*}

School of Marine Information Engineering, Hainan Tropical Ocean University, Sanya 572022, China^{1,3}
Xiamen City, Fujian Province, 361004, China²

Graduate School of Management of Technology, Pukyong National University, Busan 48547, Korea⁴

Abstract—Landslide monitoring is a crucial component of geological disaster early warning systems. Traditional landslide detection methods often suffer from insufficient accuracy or low efficiency. To address these issues, this study proposes an improved landslide detection algorithm based on YOLOv11n, aiming to enhance both detection accuracy and efficiency by optimizing the model structure. First, the GhostConv module is introduced to reduce redundant computations, thereby improving computational efficiency. Additionally, the C3K2-SCConv optimization module is incorporated, which enhances feature extraction capability and improves the recognition of landslides at different scales by integrating multi-scale information and a weighted convolution strategy. Furthermore, the SimAM attention mechanism is implemented to adaptively adjust feature map weights, strengthening key features in landslide regions and improving detection accuracy. Experimental results demonstrate that the improved model achieves a mean average precision (mAP@0.5) of 83.3%, a precision of 85.5%, and a recall of 78.1%, representing increases of 2.0%, 3.2%, and 2.8%, respectively, compared to the baseline model. The proposed improvements provide a more accurate and efficient landslide detection method, contributing to the precision of geological disaster early warnings and enhancing the reliability of disaster prevention and mitigation efforts.

Keywords—YOLOv11n; landslide detection; GhostConv module; C3K2-SCConv module; SimAM attention mechanism

I. INTRODUCTION

Landslides, as a common and severe geological disaster, typically occur with the sudden collapse or movement of mountain masses, directly threatening the safety of local residents, infrastructure, and the ecological environment [1]. Particularly in mountainous and coastal regions with high rainfall, the frequency of landslide occurrences continues to rise due to triggering factors such as precipitation, earthquakes, and engineering construction. Traditional landslide monitoring methods primarily rely on ground surveys and physical measurement techniques, such as leveling measurements [2], ground-based radar (GB-InSAR) [3], and GPS positioning [4]. These methods can provide high-precision monitoring data and offer significant advantages, especially in small-scale landslide monitoring and early warning applications. However, traditional methods often suffer from limitations in data acquisition, high labor intensity, and difficulties in real-time monitoring. These challenges become particularly prominent in large-scale, complex terrains or post-disaster scenarios, where timely and accurate monitoring results may not be readily

available. In recent years, the integration of deep learning technology has introduced a novel solution for landslide monitoring. Deep neural networks (DNNs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs) have made remarkable progress in fields such as image processing [5-7], pattern recognition, and big data analysis. Specifically, in landslide detection, deep learning models can automatically extract features from large-scale remote sensing imagery, identify potential landslide-prone areas, and detect changes over time. This approach overcomes the reliance of traditional methods on expert experience and manual feature extraction, offering a more efficient and automated solution.

Although deep learning technology has made significant progress in landslide disaster monitoring, several challenges and limitations remain in current research. For example, Han et al. [8] proposed a novel neural network model called Dynahead-YOLO, which integrates scale-aware, spatial-aware, and task-aware attention mechanisms to enhance landslide decoding in complex backgrounds. However, small-sample landslide detection remains a challenge. Yang et al. [9] introduced a lightweight attention-guided YOLO model (LA-YOLO-LLL), which incorporates MobileNetv3 as the backbone network and designs a lightweight pyramid feature reuse and fusion attention mechanism while integrating the level set layer to generate precise landslide boundaries. However, at high resolutions, the model still struggles with detecting small-scale landslides. Liu et al. [10] proposed the SE-YOLOv7 landslide detection algorithm, which integrates an attention mechanism and an improved loss function, significantly improving detection accuracy. Nevertheless, the algorithm is still prone to false detections in areas with complex terrain features. Liu et al. [11] introduced a multi-scale sample-based complex background enhancement method (MSSCBE) to improve sample quality in landslide detection. However, the model exhibited poor generalization ability. Zhang et al. [12] designed a multi-scale feature extraction module that combines efficient channel attention, average pooling, and spatially separable convolution. Despite its effectiveness, challenges persist in processing small-scale landslides in complex backgrounds. Yu et al. [13] proposed an automatic loess landslide detection method based on Google Earth Engine (GEE) and an improved YOLOX algorithm, achieving an average accuracy of 95.43%. However, the model's performance still needs improvement in detecting small-scale landslides, and it is prone to false detections at the edges of remote sensing images. Hou et al. [14] developed a landslide detection method based on an improved YOLOX

object detection model, achieving promising detection performance. However, the accuracy and robustness of the model in detecting landslides across multiple categories, different terrains, and landscapes remain relatively low. Cheng et al. [15] reviewed deep learning technology and its fundamental principles, along with the current status of landslide remote sensing databases. They introduced classic deep learning models for landslide detection and analyzed the strengths and limitations of various models. Chandra et al. [16] proposed a novel generalized efficient layer aggregation network (GELAN) based on an attention mechanism, which significantly improved detection performance. However, the model's complexity remains a major drawback.

Based on the existing research, despite the significant progress made by deep learning techniques in landslide detection tasks, there are still several key challenges that need to be addressed. To this end, this study proposes a landslide detection algorithm based on the improved YOLOv11, with its core contributions mainly reflected in the following aspects:

- 1) By introducing the GhostConv module, the model's parameter count and computational complexity are reduced while maintaining high performance.
- 2) The integration of the C3K2-SCConv module enhances the model's ability to recognize multi-scale features, improving the detection of landslides and other complex features.
- 3) The incorporation of the SimAM attention mechanism strengthens the representation of key features, further improving the overall model performance.

II. RELATED WORK

A. Vision Transformer (ViT)

Vision Transformer (ViT) [17] is a neural network architecture designed for image recognition tasks, inspired by the Transformer model originally developed for natural language processing (NLP). The core idea of ViT is to divide an image into fixed-size patches and process them in a manner similar to how words are handled in NLP tasks. Specifically, an image is first segmented into small patches, each of which is flattened and mapped to a high-dimensional space, akin to word embeddings in NLP. To help the model understand the spatial position of these patches in the original image, positional encoding is added. The patch embeddings, combined with positional encoding, are then fed into a Transformer encoder, which leverages the self-attention mechanism to capture relationships between different patches. This allows the model to grasp the global context of the image rather than just local features. After processing through the encoder, a classification token (similar to the [CLS] token in BERT) is added to the sequence, and its output is used for image classification. Like other deep learning models, ViT is trained in an end-to-end manner using labeled datasets. One of the key advantages of ViT is its strong performance on large-scale datasets, where it often outperforms traditional convolutional neural networks (CNNs, especially when the model size is large. Additionally, due to the self-attention mechanism, ViT can capture the global context of an image, enabling it to model long-range dependencies between different image

patches. However, ViT also has some limitations. First, it requires a large amount of training data to achieve optimal performance, and on small datasets, it often underperforms compared to CNNs. Second, due to the high computational complexity of the Transformer architecture, ViT requires significant computational resources when processing high-resolution images. Despite these challenges, ViT represents a major breakthrough in computer vision and has inspired numerous improved versions, such as DeiT (Data-efficient Image Transformer), which is designed to enhance performance when training data is limited.

B. SE Attention Mechanism

The Squeeze-and-Excitation (SE) attention mechanism module [18] is a method to enhance the performance of neural networks by adaptively recalibrating the inter-channel feature responses. Its core idea is to learn the "importance" of each channel, thereby improving the network's feature representation capability. The working process of the SE module can be divided into three main steps: First, global average pooling (GAP) is applied to compress the spatial information of the input feature map, resulting in a global feature vector of size $1 \times 1 \times C$, where C represents the number of channels. Next, the SE module processes these global features through an "excitation" process, which consists of a two-layer fully connected neural network. The first layer uses the ReLU activation function to introduce non-linearity, while the second layer applies the Sigmoid activation function to output the "weight" or "importance" values for each channel, which lie between 0 and 1. These values represent the contribution of each channel to the final decision. Finally, the SE module performs a "recalibration" by multiplying the obtained channel weights with the original feature map on a per-channel basis, enhancing the response of important channels while suppressing less important ones. The advantages of the SE module lie in its strong adaptability, as it automatically learns the significance of each channel, and its low computational cost, making it an effective method to improve model performance in visual tasks.

C. YOLOv11 Algorithm

The network architecture of YOLOv11 [19] is primarily divided into three parts: the backbone, the neck, and the head. In the backbone section, YOLOv11 introduces innovative convolution mechanisms such as C3K2, which allow for more effective extraction of key features in the shallow layers of the network. This creates a more optimized feature extraction structure. Compared to traditional convolution structures, it enables the network to capture essential feature information crucial for object detection with greater precision, especially in complex scenarios. This sets a solid foundation for subsequent detection tasks and addresses the issue of incomplete or inaccurate feature extraction in previous models. In the neck section, the C2PSA mechanism is employed, which integrates a multi-head attention mechanism within C2. This allows the model to better capture spatial context information, improving feature fusion and information transmission efficiency. Compared to conventional designs, this mechanism makes fuller use of contextual information and comprehensively integrates features from different layers. It resolves issues such as reduced detection accuracy and insufficient feature fusion

GhostConv decomposes the input feature map by generating high-quality feature maps using standard convolutions, followed by simple operations (such as linear transformations) to generate low-complexity "Ghost" feature maps. These Ghost feature maps, although requiring fewer computations and

parameters, retain most of the input information and are fused with the feature maps generated by the standard convolution. This design significantly reduces the model's computational and storage requirements, leading to improved computation speed.

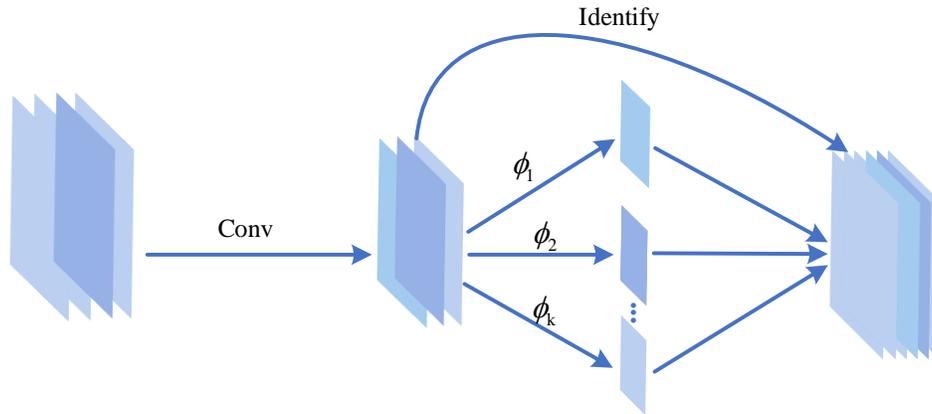


Fig. 2. GhostConv module.

B. C3k2-SCConv Module

The replacement of C3k2 with C3k2-SCConv is aimed at enhancing feature extraction capabilities and improving object detection accuracy while maintaining computational efficiency. C3k2-SCConv inherits the multi-branch structure of C3k2 and introduces SCConv [21] to strengthen feature representation. SCConv potentially enhances the model's focus on key information through spatial and channel attention mechanisms, leading to better performance in complex backgrounds or small object detection tasks. Additionally, SCConv increases the local receptive field, enhancing the robustness of feature representation, allowing the model to maintain high detection capability even in cases of object deformation or occlusion. Compared to traditional convolution, C3k2-SCConv reduces redundant computation through more efficient feature extraction, optimizes cross-layer feature fusion, and reduces information loss, thus achieving a better balance between accuracy and speed.

As shown in Fig. 3, the BottleneckSCConv module is an optimization of the feature extraction unit based on the classic

Bottleneck structure, achieved by replacing one of the standard convolution layers (Conv) with SCConv, thereby enhancing feature representation capabilities. The process begins with initial convolution processing, followed by the SCConv module, which may incorporate local attention or cross-channel interaction mechanisms to strengthen key information extraction. Subsequently, the features from the SCConv branch are fused with the original input through a concatenation (Concat) operation, thereby improving the model's feature representation ability. Building on this, C3k2-SCConv inherits the C3k2 structure and introduces the BottleneckSCConv in its Bottleneck structure to further enhance feature extraction. C3k2-SCConv constructs deep feature extraction capabilities by stacking multiple BottleneckSCConv blocks (n times), and combines multi-branch information through feature concatenation (Concat) operations. Finally, the features are integrated through convolutional layers. This improvement enhances the model's feature extraction ability while maintaining its efficiency, making it more suitable for high-performance detection tasks.

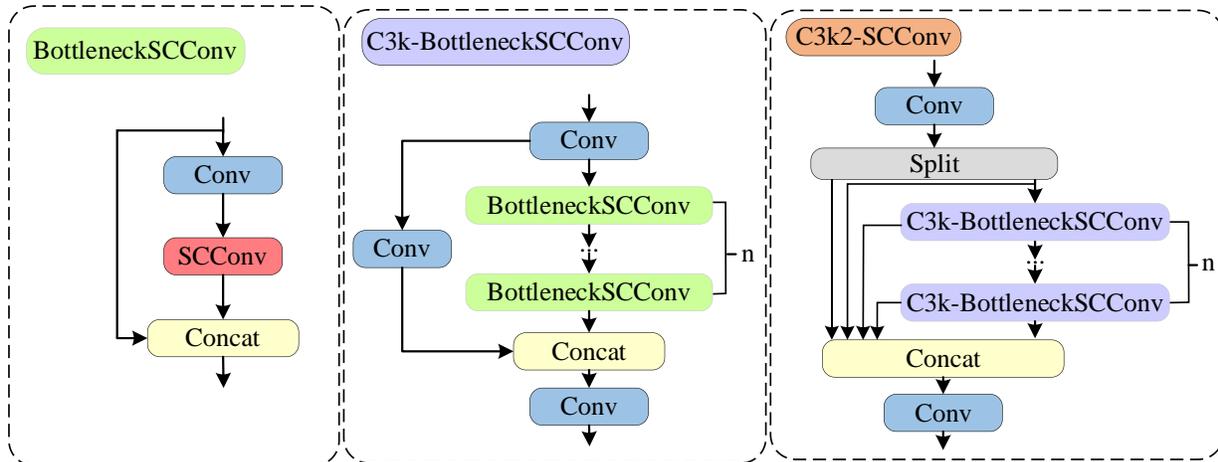


Fig. 3. C3k2-SCConv module.

C. SimAM Attention Mechanism

In this study, we optimized the YOLOv11n object detection algorithm to enhance feature extraction capabilities and detection accuracy while maintaining high computational efficiency. We strengthened the model's attention mechanism to further optimize the expression of target features and the suppression of background information. Specifically, we introduced the SimAM (Simple Attention Module) attention mechanism after the C2PSA module to increase the model's focus on important features, reduce interference from irrelevant information, and thereby improve the accuracy of object detection.

As shown in Fig. 4, the core idea of the SimAM [22] mechanism is to evaluate the differences between neurons through a physical model in order to determine their importance. The energy function $E(x_i)$ calculates the difference between the target neuron x_i and its neighboring neurons $\{x_j | j \in N\}$. A lower energy value indicates a greater difference between the target neuron and its surrounding neurons, implying higher importance. The formula for the energy function is:

$$E(x_i) = \frac{4(\sigma^2 + \lambda)}{(x_i - \mu)^2 + 2\sigma^2 + 2\lambda} \quad (1)$$

In the formula, x_i represents the target neuron, and λ is a hyperparameter

SimAM also requires the calculation of the mean and variance of the neurons within the neighborhood. This helps to more comprehensively assess the relationship between each neuron and its surrounding neurons. The formulas for calculating the mean and variance are as follows:

$$\mu = \frac{1}{N} \sum_{j=1}^N x_j \quad (2)$$

$$\sigma^2 = \frac{1}{N} \sum_{j=1}^N (x_j - \mu)^2 \quad (3)$$

In the formulas, N represents the number of neurons within the neighborhood, and x_j denotes the value of each adjacent neuron.

Finally, SimAM adjusts the output of each neuron based on its weight, enhancing the contribution of important neurons, thereby improving the model's performance. The formula for the weighted output is as follows:

$$\tilde{X} = \text{sigmoid}\left(\frac{1}{E}\right) \odot X \quad (4)$$

where, E groups are all $E(x_i)$ across the channel and spatial dimensions. "sigmoid" is added to restrict too large value in E . It will not influence the relative importance of each neuron because a sigmoid is a monofonic function.

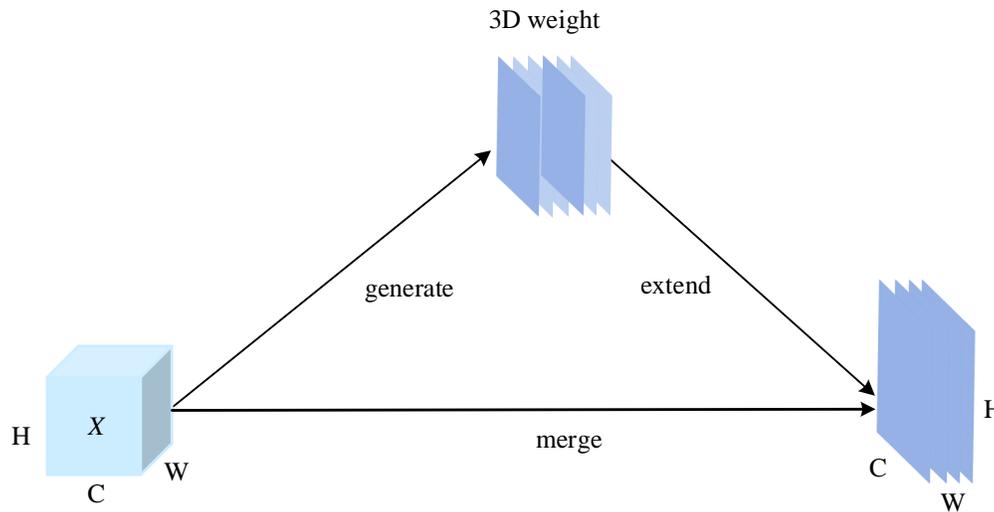


Fig. 4. SimAM attention mechanism.

IV. EXPERIMENTAL SETUP

A. Dataset

The dataset used in this study consists of two main categories: landslides and storms, with all images manually annotated to ensure accuracy and consistency. Landslides refer to the phenomenon of soil, rocks, and other materials sliding or collapsing due to natural factors such as rainfall and earthquakes. Storms include severe meteorological events such as thunderstorms and hurricanes, typically accompanied by

strong winds and heavy rainfall. Existing landslide recognition datasets primarily rely on high-resolution remote sensing images; however, such data fail to accurately reflect the complex environments present during landslides, such as dust clouds during dry conditions (as shown in Fig. 5). Additionally, processing remote sensing images demands significant computational resources, limiting the feasibility of real-time applications. To address these challenges, this study constructed a multi-domain landslide dataset consisting of 4,735 images, with 77.8% of the data sourced from news and

live video frame extraction, and 22.2% from drone remote sensing data, aimed at improving the model's recognition ability in complex scenarios. To enhance the dynamic features of the data, frame extraction was employed. Given the high-speed motion characteristics of landslides, the video frame extraction interval was set to 1, 1.5, or 2 seconds. Since some images were obstructed by watermarks, blurred, or blocked by



buildings, offline data augmentation was applied to part of the news and live video data, resulting in 1,125 images, or 24% of the total dataset. The data augmentation methods combined the few-shot learning strategies proposed by Wang et al. [23], Lu et al. [24], and the random erasure and noise addition techniques proposed by Zhong et al. [25] and Zhang et al. [26], aiming to enhance the system's robustness.



Fig. 5. Comparison of the dust phenomena caused by sandstorms and landslides.

B. Hyperparameter Settings

During training, the image resolution was set to 640×640, with a batch size of 64 to fully utilize the GPU's computational power and accelerate model training. The number of training epochs was set to 200, and the initial learning rate was set to 0.01. To improve data loading efficiency, 4 worker threads were used. Additionally, the momentum parameter was set to 0.937, and the weight decay coefficient was set to 0.0005 to optimize the training process and prevent overfitting.

C. Experimental Platform

The experiments in this study were conducted on the following experimental platform: PyTorch 1.10.0 was used as the deep learning framework, with Python version 3.8 running on the Ubuntu 20.04 operating system, and CUDA version 11.3 to leverage GPU acceleration. The experiments were performed using a single RTX 4090 (24GB) GPU for training, equipped with an AMD EPYC 7T83 64-Core Processor (22 vCPUs) and 90GB of memory. This platform provided powerful computational capabilities for model training, effectively accelerating the training process.

D. Evaluation Metrics

In this study, the evaluation metrics used include F1 score, precision (P), recall (R), average precision (AP), and mean average precision (mAP) [27]. Additionally, the number of parameters (Parameters) was also considered. The formulas for these metrics are as follows:

$$\text{Precision} = \frac{T_p}{T_p + F_p} \quad (4)$$

$$\text{Recall} = \frac{T_p}{T_p + F_N} \quad (5)$$

$$\text{AP} = \int_0^1 P(R) dR \quad (6)$$

$$\text{mAP} = \frac{1}{n} \sum_{i=0}^n AP(i) \quad (7)$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (8)$$

where, T_p represents the number of correctly detected targets; F_p represents the number of falsely detected targets; F_N represents the number of missed targets; n denotes the number of categories; and $AP(i)$ is the average precision of the i -th target class.

V. EXPERIMENTAL ANALYSIS

A. Algorithm Comparison Experiment

To evaluate the advantages of our improved algorithm, we conducted a comparison with other algorithms. In the field of object detection, algorithm performance is typically assessed using several key metrics, such as precision, recall, mAP@0.5, FPS (frames per second), GFLOPS (computational complexity), and the number of parameters (Params). Precision and recall primarily reflect the model's detection accuracy and recall capability, while mAP@0.5 provides a comprehensive measure of the model's performance across different detection thresholds. FPS and GFLOPS assess the model's real-time performance and computational efficiency, respectively, while the number of parameters reflects the model's complexity. By comparing these metrics, we can gain a deeper understanding of the strengths and weaknesses of different algorithms in terms of accuracy, real-time performance, and computational efficiency.

From the experimental comparison results in Table I, different object detection algorithms exhibit varying performances in metrics such as precision, recall, mAP@0.5, FPS, GFLOPS, and the number of parameters (Params). The Ours algorithm achieved the best results in precision (85.5%), recall (78.1%), and mAP (83.3%), while maintaining high real-

time performance (128.5 FPS), low computational complexity (6.3 GFLOPS), and a relatively low number of parameters (2.72M), demonstrating the optimal overall performance. YOLOv1n excelled in real-time performance (147.0 FPS) and computational efficiency (6.3 GFLOPS), but its precision (82.3%) and mAP (81.3%) were slightly lower than those of Ours. YOLOv5s performed well in real-time performance (137.0 FPS), with moderate precision (78.6%) and mAP (76.5%), but its computational complexity (15.8 GFLOPS) and parameter count (7.01M) were relatively high. YOLOv10n showed moderate performance in precision (81.6%), mAP

(77.5%), and real-time performance (122.6 FPS), with lower computational complexity (8.3 GFLOPS) and parameter count (2.73M). YOLOv9-gelan-s demonstrated excellent precision (85.3%) and mAP (82.4%), but its computational complexity (26.2 GFLOPS) and parameter count (7.07M) were higher. In comparison, RT-DETR-18 and YOLOv7-tiny performed poorly in terms of precision, recall, and mAP, with RT-DETR-18 having a higher computational complexity (56.9 GFLOPS) and parameter count (19.87M). Overall, the Ours algorithm generally outperforms traditional algorithms in terms of performance.

TABLE I. EXPERIMENTAL COMPARISON RESULTS OF DIFFERENT ALGORITHMS

Algorithm	Precision/%	Recall/%	mAP@0.5/%	FPS (f/s)	GFLOPS	Params (M)
YOLOv5s[28]	78.6	76.5	76.5	137.0	15.8	7.01
YOLOv7-tiny[29]	76.5	66.8	71.3	120.8	13.0	6.01
YOLOv8n[30]	83.2	70.7	78.9	106.3	8.1	3.00
YOLOv9-gelan-s[31]	85.3	74.5	82.4	81.3	26.2	7.07
YOLOv10n[32]	81.6	72.1	77.5	122.6	8.3	2.73
RT-DETR-18[33]	76.6	67.4	70.5	68.4	56.9	19.87
YOLOv11n	82.3	75.3	81.3	147.0	6.3	2.58
Ours	85.5	78.1	83.3	128.5	6.3	2.72

TABLE II. COMPARISON RESULTS OF VARIOUS ALGORITHMS IN AVERAGE PRECISION (AP/%)

Algorithms Classes	YOLOv5s	YOLOv7-tiny	YOLOv8n	YOLOv9-gelan-s	YOLOv10n	RT-DETR-18	YOLOv11n	Ours
landslide	65.8	56.9	66.8	74.3	66.8	62.7	72.0	74.3
storm	87.1	85.7	90.9	90.5	88.1	78.4	90.6	92.3

From the average precision (AP) comparison results in Table II, it is evident that different algorithms exhibit significant performance differences in the landslide and storm tasks. YOLOv9-gelan-s and Ours tied for the best performance in the landslide category, both achieving an average precision of 74.3%. In the storm category, Ours achieved the best performance with an average precision of 92.3%, significantly outperforming other algorithms. YOLOv8n and YOLOv11n also performed well in both tasks, with YOLOv8n achieving an average precision of 90.9% in the storm category. YOLOv10n achieved average precisions of 66.8% and 88.1% for the

landslide and storm categories, respectively, slightly outperforming YOLOv5s, which scored 65.8% and 87.1%. However, neither of these two algorithms reached the optimal performance. In contrast, YOLOv7-tiny and RT-DETR-18 performed poorly, particularly in the storm category, where RT-DETR-18's average precision was only 78.4%, the lowest among all algorithms. Overall, the Ours algorithm excelled in both tasks, especially in complex scenarios (such as storms), where it demonstrated higher detection accuracy, offering the best overall performance. YOLOv10n and YOLOv5s showed moderate performance, with room for further optimization.

TABLE III. COMPARISON RESULTS OF VARIOUS ALGORITHMS IN PRECISION (P/%)

Algorithms Classes	YOLOv5s	YOLOv7-tiny	YOLOv8n	YOLOv9-gelan-s	YOLOv10n	RT-DETR-18	YOLOv11n	Ours
landslide	69.4	66.5	73.9	79.3	73.7	72.8	79.7	77.1
storm	87.8	86.5	92.6	91.3	89.4	80.4	84.9	93.8

From the data in Table III, it can be observed that different algorithms exhibit varying precision in the landslide and storm categories. In the landslide category, YOLOv11n performed the best with a precision of 79.7%, outperforming YOLOv7-tiny (66.5%). In the storm category, Ours demonstrated the best performance, with a precision of 93.8%, significantly surpassing other algorithms, especially RT-DETR-18 (80.4%). YOLO series algorithms (such as YOLOv5s, YOLOv8n, etc.) also performed well in the storm category, but their precision

varied greatly in the landslide category. Overall, Ours performed best in storm detection, while YOLOv11n and Ours were the most outstanding in landslide detection. RT-DETR-18 performed relatively poorly in both tasks.

From the recall (R) data in Table IV, Ours demonstrated the best performance in both the landslide and storm categories, with a landslide recall of 67.2% and a storm recall of 89.0%, significantly outperforming other algorithms. YOLOv11n

followed closely, performing well in the landslide category with a recall of 63.7%, and also showing good performance in the storm category. YOLOv5s achieved a recall of 87.7% in the storm category, which was quite notable, but in the landslide category, its recall was 65.3%, slightly lower than

that of Ours. In contrast, YOLOv7-tiny and RT-DETR-18 generally had lower recall rates, with YOLOv7-tiny's recall in the landslide category being only 51.4%. Overall, Ours has a significant advantage in recall, making it suitable for applications that require high sensitivity to missed detections.

TABLE IV. COMPARISON RESULTS OF VARIOUS ALGORITHMS IN RECALL (R/%)

Algorithms Classes	YOLOv5s	YOLOv7-tiny	YOLOv8n	YOLOv9-gelan-s	YOLOv10n	RT-DETR-18	YOLOv11n	Ours
landslide	65.3	51.4	56.4	63.0	63.0	63.4	63.7	67.2
storm	87.7	82.2	84.9	86.1	81.3	71.4	87.0	89.0

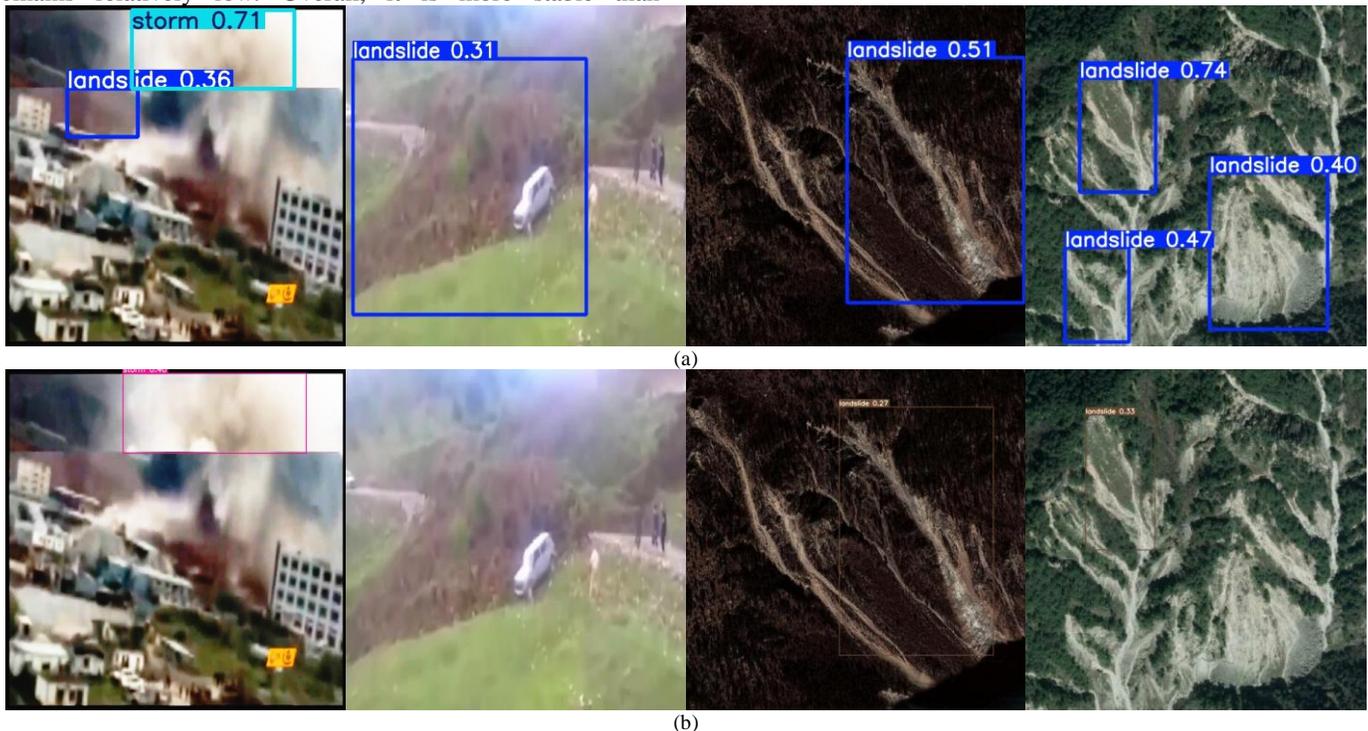
B. Result Visualization

To comprehensively evaluate the superiority of the Ours algorithm in object detection tasks, we conducted a visual comparison with several existing detection algorithms. Although the existing algorithms, such as YOLOv5s, YOLOv7-tiny, YOLOv8n, YOLOv9-gelan-s, YOLOv10n, RT-DETR-18, and YOLOv11n, exhibit certain detection capabilities in different scenarios, they still have limitations in terms of detection accuracy, target confidence, and missed detection phenomena. Therefore, we visually compared the performance of different algorithms in the landslide (landslide) and storm (storm) detection tasks.

Fig. 6 shows the performance comparison of different models in landslide and storm detection. YOLOv5s exhibits basic detection capabilities, but with lower confidence, resulting in some targets being misdetected or missed. YOLOv7-tiny has even lower confidence, with some landslide areas not detected and a higher number of false detections, indicating that it is not robust enough for small targets or complex backgrounds. YOLOv8n shows improvements in landslide detection, but the confidence in the storm category remains relatively low. Overall, it is more stable than

YOLOv7-tiny, but there are still some missed detections. In contrast, YOLOv9-gelan-s, YOLOv10n, RT-DETR-18, and YOLOv11n show improvements in detection accuracy, but all exhibit overlapping detection boxes, indicating that these models may have redundant predictions when processing landslide areas, generating multiple detection boxes for the same target region, which affects the final detection results. Additionally, YOLOv11n also experiences missed detections, with some landslide areas not being recognized. Among all models, Ours achieves the best detection performance, with the highest confidence, the most stable detection boxes, a wider detection range for landslide targets, and effectively reduces overlapping detection boxes while avoiding missed detections, resulting in the best overall performance.

In object detection tasks, the performance of the model is typically evaluated using precision, recall, and F1 score. The F1-Confidence curve reflects the variation in the F1 score at different confidence thresholds, providing an intuitive representation of the model's overall detection capability at different decision boundaries. To compare the performance of different object detection models, we plotted the F1-Confidence curve and conducted an analysis.





(c)



(d)



(e)



(f)



(g)



Fig. 6. Detection results of different algorithms on the dataset: (a) YOLOv5s; (b) YOLOv7-tiny; (c) YOLOv8n; (d) YOLOv9-gelan-s; (e) YOLOv10n; (f) RT-DETR-18; (g) YOLOv11n; (h) Ours.

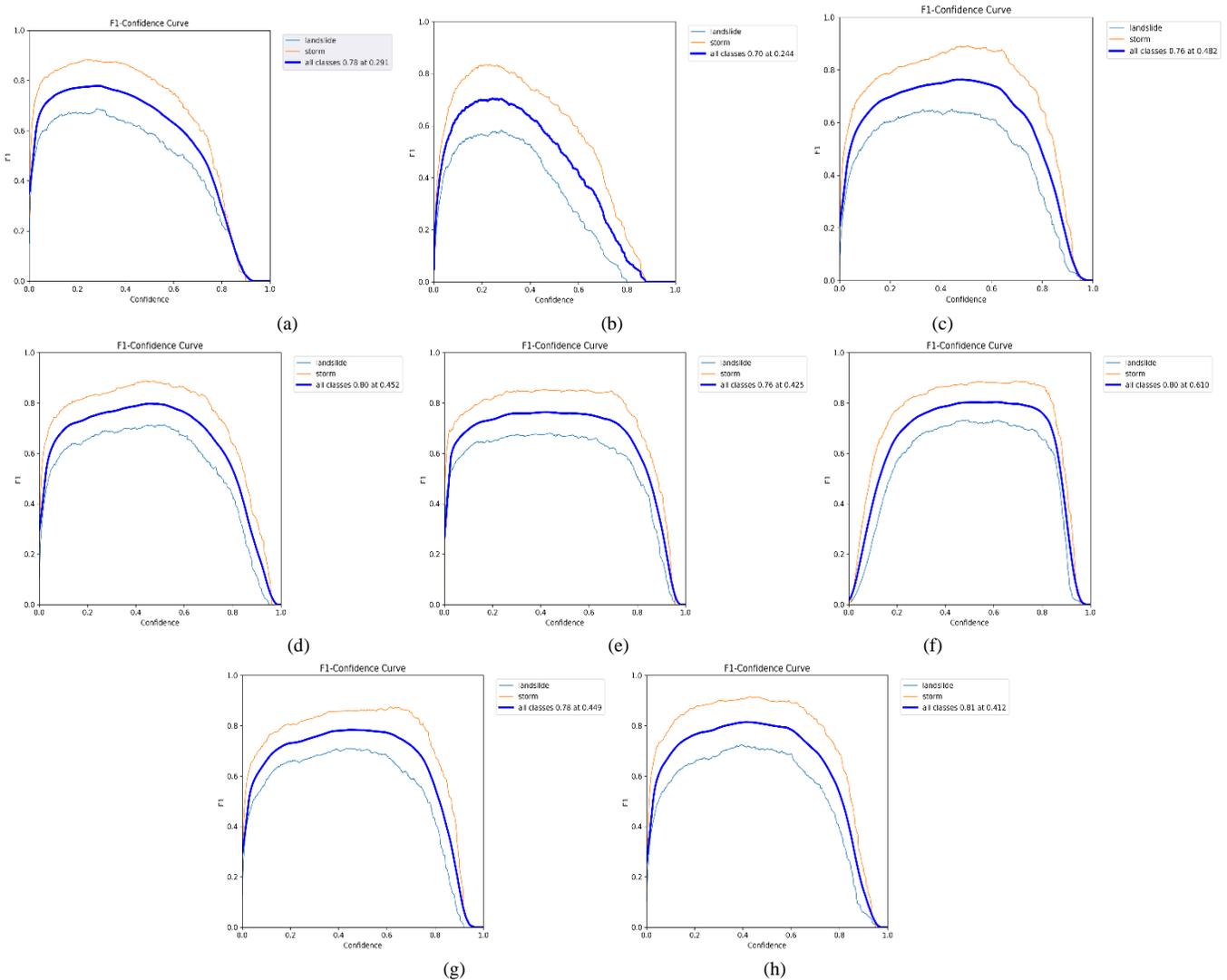


Fig. 7. F1-Confidence Curve: (a) YOLOv5s; (b) YOLOv7-tiny; (c) YOLOv8n; (d) YOLOv9-gelan-s; (e) YOLOv10n; (f) RT-DETR-18; (g) YOLOv11n; (h) Ours.

To comprehensively evaluate the detection capabilities of different object detection models at various confidence thresholds, we plotted the F1-Confidence curve to visually demonstrate the variation in F1 scores across models. Fig. 7(a) to Fig. 7(h) correspond to YOLOv5s, YOLOv7-tiny,

YOLOv8n, YOLOv9-gelan-s, YOLOv10n, RT-DETR-18, YOLOv11n, and our model (Ours), used to compare their performance at different decision boundaries. The results in Fig. 7 show that Ours achieves the highest F1 score of 0.81 at a confidence threshold of 0.412, outperforming all other models

and demonstrating the best performance. YOLOv9-gelan-s, RT-DETR-18, and YOLOv11n also performed well, but still fell short of Ours. YOLOv5s and YOLOv10n achieved F1 scores of 0.78 and 0.76, respectively, showing acceptable overall performance, better than YOLOv7-tiny (0.70) and YOLOv8n (0.76), but still not surpassing Ours. YOLOv7-tiny had the lowest F1 score at only 0.70, indicating weaker detection capabilities. Overall, Ours maintained a high F1 score across different confidence ranges, with a low false detection rate, indicating stronger object detection capabilities.

This study improved the YOLOv11n object detection algorithm and evaluated its training and validation processes. To analyze the model's convergence and detection performance, we plotted the loss curve and the trends of key evaluation metrics. The loss curve illustrates the optimization of the model during the training process, while metrics such as precision, recall, and mAP reflect the model's detection capabilities. Through these curves, we validated the stability and performance improvement of the modified model.

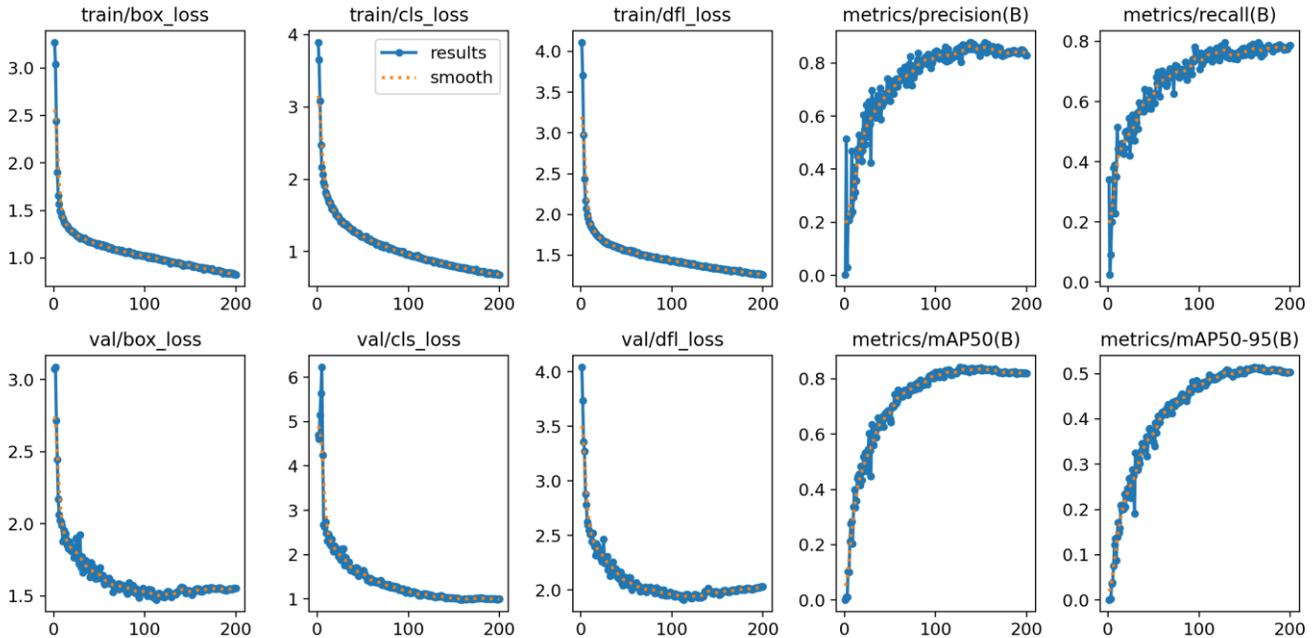


Fig. 8. Loss reduction trends and performance metric variations of the improved YOLOv11n algorithm during training and validation.

Fig. 8 illustrates the loss reduction trend and changes in key evaluation metrics during the training and validation process of the improved YOLOv11n algorithm, used to assess the model's convergence and detection performance. From the loss curves, both training losses (box_loss, cls_loss, dfl_loss) and validation losses (val/box_loss, val/cls_loss, val/dfl_loss) show a steady downward trend, indicating that the model continuously optimizes its object detection capabilities during training without significant overfitting or underfitting. In terms of model evaluation metrics, precision and recall steadily increase and stabilize, suggesting that the improved model gradually enhances its target classification and detection abilities. The mAP50 is close to 0.833, and the mAP50-95 is approximately 0.516, demonstrating the model's good detection performance and strong generalization ability at different IoU thresholds. Overall, the improved YOLOv11n algorithm outperforms the original version, but its small target detection capabilities and mAP50-95 metric can still be further improved by adjusting the learning rate, optimizing data augmentation strategies, or enhancing non-maximum suppression (NMS) processing.

To more clearly evaluate the model's classification ability, the confusion matrix is used to visually quantify the model's

prediction accuracy and misclassification rates for different categories. To assess the detection performance of the YOLOv11n algorithm and its improved version in the landslide and storm categories, we plotted the normalized confusion matrix. This allows for a comparative analysis of the classification accuracy and misclassification rates of the two models, in order to verify the performance improvement of the modified model.

Fig. 9 shows the confusion matrices of the YOLOv11n algorithm and the improved YOLOv11n algorithm, used to evaluate the model's classification performance on landslide and storm targets. From the results, the improved YOLOv11n algorithm shows improvements in both landslide and storm detection. The accuracy of landslide detection increased from 0.74 to 0.78, and the accuracy of storm detection increased from 0.90 to 0.91, while the false detection rate decreased. In particular, the proportion of storms misclassified as background reduced from 0.37 to 0.21, indicating that the improved model is more stable in storm detection. Additionally, the background misclassification rate also decreased, with the probability of landslides being misidentified as background dropping from 0.25 to 0.21, suggesting that the improved algorithm has optimized false detection suppression.

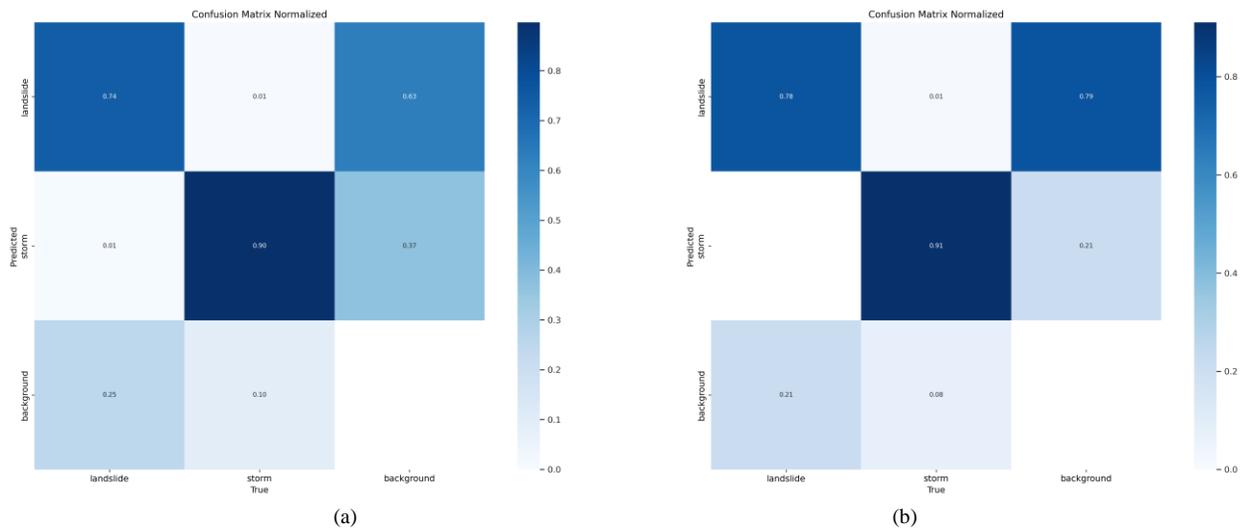


Fig. 9. Normalized confusion matrix: (a) YOLOv11n algorithm; (b) Improved YOLOv11n algorithm.

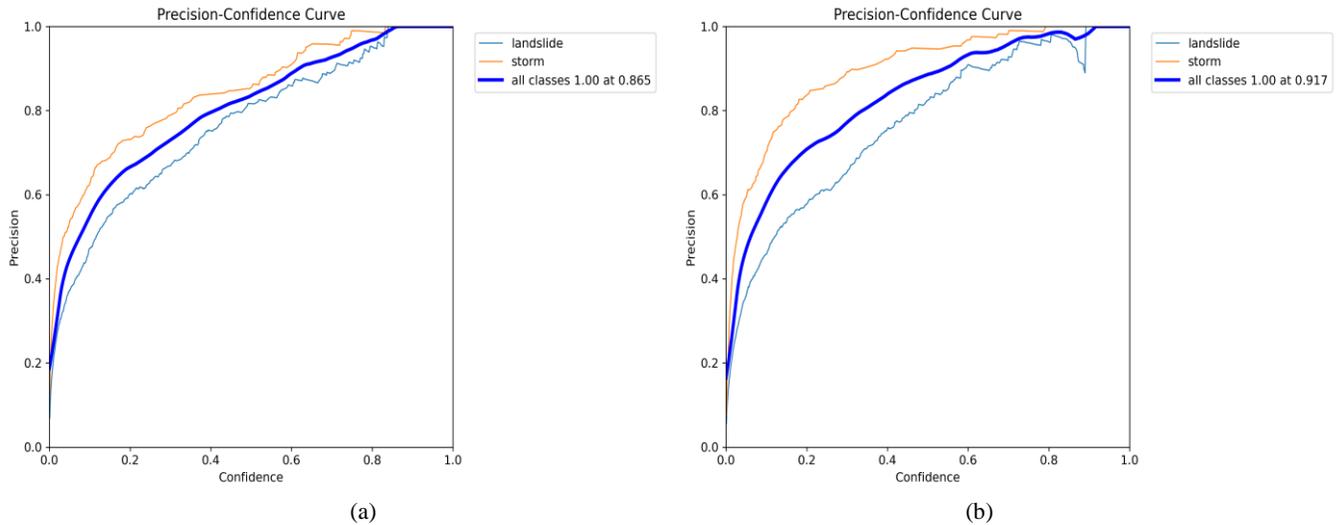


Fig. 10. Precision-confidence curves: (a) YOLOv11n algorithm; (b) Improved YOLOv11n algorithm.

Fig. 10 presents the Precision-Confidence curves for the YOLOv11n algorithm [Fig. 10(a)] and the improved YOLOv11n algorithm [Fig. 10(b)], used to evaluate the variation in precision at different confidence thresholds. The left panel [Fig. 10(a)] shows the original YOLOv11n, while the right panel [Fig. 10(b)] shows the improved YOLOv11n. The curves correspond to landslide, storm, and all classes. Overall, as the confidence threshold increases, precision gradually improves, indicating a reduction in false positives at higher confidence levels and more accurate predictions. Compared to the original model, the improved YOLOv11n shows an increase in precision across all confidence ranges, with the confidence threshold corresponding to the highest precision increasing from 0.865 to 0.917. This suggests that the improved model maintains a higher level of precision at higher confidence thresholds, thereby reducing false positives caused by low-confidence predictions.

Fig. 11 presents the Precision-Recall (P-R) curves for the YOLOv11n algorithm [Fig. 11(a)] and the improved YOLOv11n algorithm [Fig. 11(b)], used to assess the variation in precision at different recall rates. The left panel [Fig. 11(a)] shows the original YOLOv11n, while the right panel [Fig. 11(b)] shows the improved YOLOv11n. The P-R curves for landslide, storm, and all classes are plotted, with the performance at mAP@0.5 indicated for each class. Overall, the mAP@0.5 of the improved model increased from 0.813 to 0.833 across all classes, with the mAP of the storm class rising from 0.906 to 0.923 and the mAP of the landslide class increasing from 0.720 to 0.743, indicating an improvement in detection performance. The P-R curves show that the storm class consistently outperforms the landslide class in detection, while the improved YOLOv11n maintains a slower decline in precision at higher recall rates, suggesting enhanced detection ability at higher recall levels and a reduction in false negatives. Overall, the improved model achieves better performance in balancing precision and recall.

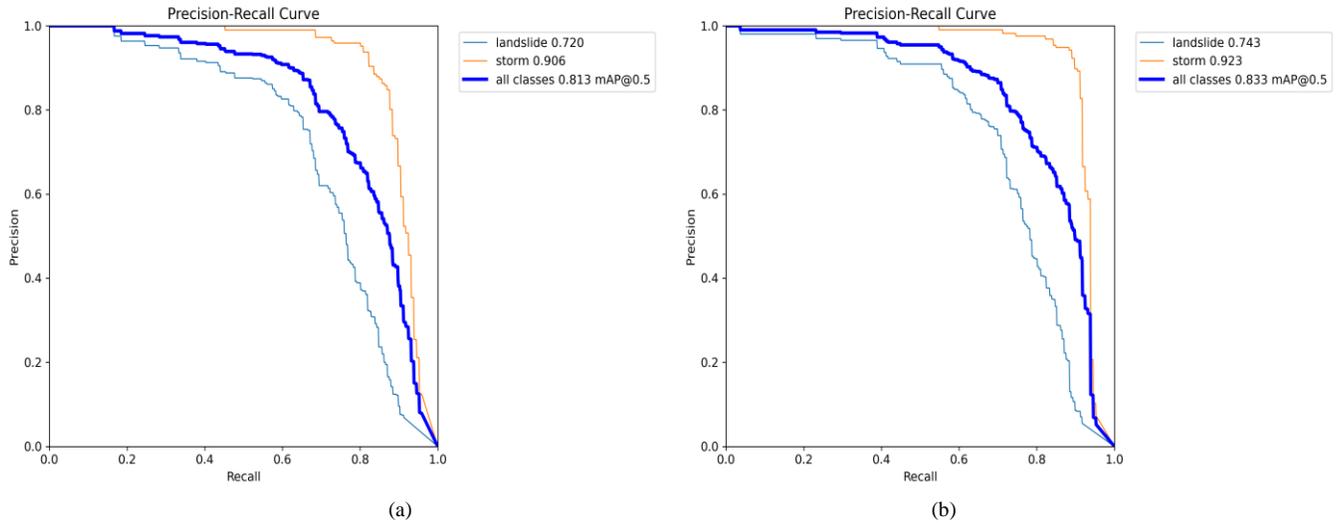


Fig. 11. Precision-recall curves: (a) YOLOv11n algorithm; (b) Improved YOLOv11n algorithm.

C. Ablation Study

To evaluate the impact of different modules on the YOLOv11n object detection performance, we conducted ablation experiments and performed a comparative analysis of detection accuracy, recall rate, mAP@0.5, inference speed (FPS), computational complexity (GFLOPS), and parameter count (Params) for each improvement. By progressively

introducing modules such as C3k2-SCConv, GhostConv, and SimAM, we assess their contributions to model performance and whether the improvements can maintain high detection accuracy while also considering inference speed and computational efficiency. The experimental results, as shown in Table V, present the performance of each configuration across various metrics.

TABLE V. ABLATION STUDY RESULTS

number	Experiment	Precision/%	Recall/%	mAP@0.5/%	FPS (f/s)	GFLOPS	Params (M)
1	YOLOv11n	82.3	75.3	81.3	147.0	6.3	2.58
2	YOLOv11n+C3k2-SCConv	85.6	74.6	82.2	102.8	6.4	2.73
3	YOLOv11n+C3k2-SCConv+GhostConv	85.9	75.3	82.8	130.3	6.3	2.72
4	YOLOv11n+C3k2-SCConv+GhostConv+SimAM	85.5	78.1	83.3	128.5	6.3	2.72

Table V presents the ablation experiment results for YOLOv11n and its various improvements, evaluating the impact of each configuration on precision, recall, mAP@0.5, frame rate (FPS), computational complexity (GFLOPS), and parameter count (Params). The baseline model, YOLOv11n, achieves the fastest inference speed with an mAP@0.5 of 81.3% and FPS of 147.0 f/s, but there is still room for improvement in detection accuracy. After adding C3k2-SCConv, precision increases to 85.6%, and mAP@0.5 slightly improves, but recall decreases, and FPS drops to 102.8 f/s, indicating that the increased computational load impacts inference efficiency. The further introduction of GhostConv increases precision to 85.9%, recall recovers by 5.3%, and mAP@0.5 rises to 82.8%, while FPS reaches 130.3 f/s, suggesting that GhostConv effectively reduces computational redundancy and enhances inference speed. Finally, with the addition of SimAM, recall further improves to 78.1%, and mAP@0.5 increases to 83.3%. Overall, the gradual inclusion of C3k2-SCConv, GhostConv, and SimAM significantly enhances detection accuracy while maintaining high inference speed, with minimal impact on computational complexity and parameter count.

VI. DISCUSSION

There are several limitations in the current study. First, the dataset lacks images captured under extreme weather conditions such as heavy rain or snow, which may weaken the model's generalization ability in complex real-world environments. Second, the sample distribution of landslide and storm categories may be imbalanced, potentially leading to suboptimal detection performance for underrepresented classes. Moreover, the current research primarily focuses on the detection of landslides and storms, while the model's ability to identify other types of geological disasters, such as debris flows or earthquakes, has not been sufficiently validated. Additionally, the model is relatively complex, which not only increases the time cost for training and inference but also poses challenges for deployment and maintenance. Future research can address these issues in two main directions. One is the expansion and augmentation of the dataset, by incorporating a larger number of images from various scenarios and weather conditions to enhance the model's robustness and generalization. The other is the optimization of the model architecture and its real-time performance, by exploring more efficient network designs and optimization algorithms to reduce computational complexity and parameter size.

VII. CONCLUSION

This study optimizes the YOLOv11n object detection algorithm by introducing the GhostConv module to optimize the convolutional computations, reducing redundant calculations while significantly improving the model's operational efficiency without compromising feature extraction capability. The introduction of the C3k2-SCConv module further optimizes the feature extraction process, effectively enhancing the representation of key features in landslide images, while maintaining optimal computational performance. The incorporation of the SimAM attention mechanism adapts the model's focus on different region features, allowing the network to more effectively perceive critical information in landslide images, thereby further improving detection accuracy. Experimental results show that the improved model outperforms other algorithms in terms of mAP@0.5, precision, and recall, demonstrating superior detection performance and computational efficiency.

REFERENCES

- [1] Casagli N, Intrieri E, Tofani V, et al. Landslide detection, monitoring and prediction with remote-sensing techniques[J]. *Nature Reviews Earth & Environment*, 2023, 4(1): 51-64.
- [2] Savvaadis P D. Existing landslide monitoring systems and techniques[J]. *From stars to earth and culture*, 2003: 242-258.
- [3] Ferrigno F, Gigli G, Fanti R, et al. GB-InSAR monitoring and observational method for landslide emergency management: the Montaguto earthflow (AV, Italy)[J]. *Natural Hazards and Earth System Sciences*, 2017, 17(6): 845-860.
- [4] Qiu D, Wang L, Luo D, et al. Landslide monitoring analysis of single-frequency BDS/GPS combined positioning with constraints on deformation characteristics[J]. *Survey Review*, 2019, 51(367): 364-372.
- [5] Zhang, Shengdong, et al. "Generative adversarial and self-supervised dehazing network." *IEEE Transactions on Industrial Informatics* 20.3 (2023): 4187-4197.
- [6] Zhang, Shengdong, et al. "Semantic-aware dehazing network with adaptive feature fusion." *IEEE Transactions on Cybernetics* 53.1 (2021): 454-467.
- [7] Liu, Yun, et al. "Single nighttime image dehazing based on unified variational decomposition model and multi-scale contrast enhancement." *Engineering Applications of Artificial Intelligence* 116 (2022): 105373.
- [8] Han, Zheng, et al. "A novel Dynahead-Yolo neural network for the detection of landslides with variable proportions using remote sensing images." *Frontiers in Earth Science* 10 (2023): 1077153.
- [9] Yang, Yueheng, et al. "Lightweight attention-guided YOLO with level set layer for landslide detection from optical satellite images." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 17 (2024): 3543-3559.
- [10] Liu, Qing, et al. "Se-yolov7 landslide detection algorithm based on attention mechanism and improved loss function." *Land* 12.8 (2023): 1522.
- [11] Liu, Xiaohui, Ling Xu, and Jinyu Zhang. "Landslide detection with Mask R-CNN using complex background enhancement based on multi-scale samples." *Geomatics, Natural Hazards and Risk* 15.1 (2024): 2300823.
- [12] Zhang, Wenjie, et al. "LS-YOLO: A novel model for detecting multiscale landslides with remote sensing images." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 17 (2024): 4952-4965.
- [13] Yu, Zhengbo, Ruichun Chang, and Zhe Chen. "Automatic detection method for loess landslides based on GEE and an improved YOLOX algorithm." *Remote Sensing* 14.18 (2022): 4599.
- [14] Hou, Heyi, et al. "A universal landslide detection method in optical remote sensing images based on improved YOLOX." *Remote Sensing* 14.19 (2022): 4939.
- [15] Cheng, Gong, et al. "Advances in deep learning recognition of landslides based on remote sensing images." *Remote Sensing* 16.10 (2024): 1787.
- [16] Chandra, Naveen, et al. "A Novel Attention-Based Generalized Efficient Layer Aggregation Network for Landslide Detection from Satellite Data in the Higher Himalayas, Nepal." *Remote Sensing* 16.14 (2024): 2598.
- [17] Arnab, Anurag, et al. "Vivit: A video vision transformer." *Proceedings of the IEEE/CVF international conference on computer vision*. 2021.
- [18] Hu, Jie, Li Shen, and Gang Sun. "Squeeze-and-excitation networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
- [19] Wang, Dingran, et al. "SDS-YOLO: An improved vibratory position detection algorithm based on YOLOv11." *Measurement* 244 (2025): 116518.
- [20] Wang, Tao, and Shiqing Zhang. "DSC-Ghost-Conv: A compact convolution module for building efficient neural network architectures." *Multimedia Tools and Applications* 83.12 (2024): 36767-36795.
- [21] Li, Jiafeng, Ying Wen, and Lianghua He. "Sconv: Spatial and channel reconstruction convolution for feature redundancy." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023.
- [22] Yang, Lingxiao, et al. "Simam: A simple, parameter-free attention module for convolutional neural networks." *International conference on machine learning*. PMLR, 2021.
- [23] Wang, Yaqing, et al. "Generalizing from a few examples: A survey on few-shot learning." *ACM computing surveys (csur)* 53.3 (2020): 1-34.
- [24] Lu, Jiang, et al. "A survey on machine learning from few samples." *Pattern Recognition* 139 (2023): 109480.
- [25] Zhong, Zhun, et al. "Random erasing data augmentation." *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. No. 07. 2020.
- [26] Zhang, Hongyi, et al. "mixup: Beyond empirical risk minimization." *arXiv preprint arXiv:1710.09412* (2017).
- [27] Kou R, Wang C, Peng Z, et al. Infrared small target segmentation networks: A survey[J]. *Pattern Recognition*, 2023, 143: 109788.
- [28] Chen, Chen, et al. "Improved YOLOv5s model for key components detection of power transmission lines." *arxiv preprint arxiv:2502.06127* (2025).
- [29] Liu, Le, et al. "Improved YOLOv7 Based on Reparameterization and Attention Mechanism for Continuous Casting Slab Detection." *IEEE Transactions on Instrumentation and Measurement* (2025).
- [30] Zhong, Jiaqi, et al. "Improved real-time object detection method based on YOLOv8: a refined approach." *Journal of Real-Time Image Processing* 22.1 (2025): 1-13.
- [31] Anandakrishnan, Jayakrishnan, et al. "Precise Spatial Prediction of Rice Seedlings From Large Scale Airborne Remote Sensing Data Using Optimized Li-YOLOv9." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* (2024).
- [32] Meng, Zhichao, et al. "YOLOv10-pose and YOLOv9-pose: Real-time strawberry stalk pose detection models." *Computers in Industry* 165 (2025): 104231.
- [33] Zhou, Chunheng, et al. "MSRT-DETR: A novel RT-DETR model with multi-scale feature sequence for cell detection." *Biomedical Signal Processing and Control* 103 (2025): 107378.