applied sciences

MDPI

Article

Road Rockfall Detection by Integrating Feature Engineering with YOLO and Cascade Decision Fusion

Zhiqing Qin [1,2], Tao Niu [3], Caijin Lu [1,2], Yongsheng Dai [1,2], Xiantao Liu [3], Peng Peng [3,*] and Jiachun Li [4,*]

[1] Fujian Provincial Transportation Planning and Design Institute Co., Ltd., Fuzhou 350000, China; zhiqing.qin@outlook.com (Z.Q.); yantu2004@126.com (C.L.); jgydys@126.com (Y.D.)
[2] Fujian Provincial Engineering Research Center for Cross-Sea Passage Bridges and Tunnels, Fuzhou 350000, China
[3] School of Electrical and Control Engineering, Shaanxi University of Science and Technology, Xi'an 710021, China; 250612056@sust.edu.cn (T.N.); yl011019@163.com (X.L.)
[4] School of Highway, Chang'an University, Xi'an 710064, China
* Correspondence: pengpeng@sust.edu.cn (P.P.); zhs@chd.edu.cn (J.L.)

Abstract

In roadside surveillance imagery, shadows, vegetation, vehicles, exposed pavement, and water stains may exhibit local textures and morphological characteristics similar to those of rockfalls, causing a standalone You Only Look Once (YOLO) real-time object detector to generate frequent false-positive detections. To address this issue, this study proposes a serial cascaded detection method that integrates an improved YOLO detector with machine-learning-based secondary verification. In the YOLO branch, channel-prior convolutional attention (CPCA) and learnable weighted multiscale feature fusion are introduced to enhance target representation under complex background conditions and generate candidate bounding boxes. In the machine-learning branch, handcrafted features describing texture, color, shape, edges, morphology, and frequency-domain characteristics are extracted from the candidate regions. A verifier selected through multi-model comparison and ensemble evaluation is then employed to confirm the YOLO-generated candidates. For parameter optimization, the operating point of the standalone YOLO detector with the highest F_1-score is first selected as the baseline. A two-dimensional grid search is subsequently performed over 95 threshold combinations consisting of five YOLO candidate-confidence thresholds and nineteen machine-learning confidence thresholds. The optimal configuration is determined using a weighted improvement score defined according to the relative changes in precision, recall, and the F_1-score with respect to the baseline. The best overall performance is achieved when the YOLO and machine-learning confidence thresholds are set to 0.25 and 0.75, respectively. Compared with the standalone YOLO detector, the proposed cascaded model improves accuracy from 93.3% to 94.1%, precision from 90.1% to 92.6%, and the F_1-score from 93.4% to 94.0%, while recall decreases slightly from 97.0% to 95.5%. These results demonstrate that interpretable local features can effectively filter out false-positive YOLO candidates, thereby suppressing false alarms and improving overall discrimination performance at the cost of only a limited reduction in recall. The developed system has been deployed on rockfall-prone sections of highways G210 and G108, providing technical support for real-time road rockfall monitoring and early warning.

Keywords: road rockfall detection; YOLO; feature engineering; machine learning; cascaded detection; false-positive suppression

Check for updates

Academic Editor: Christos Bouras

Received: 3 August 2026
Revised: 13 September 2026
Accepted: 14 September 2026
Published: 16 September 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Road rockfalls are characterized by sudden occurrence, uncertain spatial extent, and a limited time window for emergency response. Once fallen rocks intrude onto a roadway, they may cause traffic disruption, vehicle damage, and casualties. Therefore, the timely detection of rockfalls and accurate determination of their locations and affected areas are essential for traffic control and emergency response. In recent years, remote sensing, LiDAR, computer vision, and Internet of Things (IoT) technologies have facilitated the transition of rockfall detection from manual inspection to automated sensing [1–4]. However, LiDAR- and remote-sensing-based methods generally rely on specialized equipment, periodic data acquisition, or complex three-dimensional data-processing procedures. For road sections already equipped with video surveillance systems, developing low-cost, real-time image-based detection methods therefore offers greater practical engineering value [5].

Existing vision-based recognition methods can be broadly divided into conventional feature-engineering approaches and deep-learning-based approaches. Conventional methods use interpretable features, such as texture, color, edges, and geometric morphology, in combination with support vector machines (SVMs), random forests, or ensemble classifiers for target recognition [2,3]. These methods offer transparent computational procedures and are suitable for small-sample tasks. However, their performance depends heavily on feature design and threshold selection, and their generalization capability is limited by variations in target scale, illumination, pavement material, and occlusion conditions [1]. Deep convolutional neural networks can learn hierarchical semantic features through end-to-end training. Among them, the You Only Look Once (YOLO) family achieves a favorable balance between detection accuracy and inference speed and has been applied to geological-hazard identification, anomalous-object detection in road scenes, and moving-rockfall detection [6–9]. Peng et al. demonstrated the potential of an optimized YOLOv8 model for rockfall detection in real-world road environments [10], while Rockfall-YOLO further extended this framework to moving-rockfall detection [11]. Nevertheless, the performance of a standalone YOLO model remains highly dependent on the distribution of its training data. In road images, shadows, vegetation, vehicles, exposed pavement, and water stains may produce responses similar to those of rockfall targets, resulting in numerous false positives. Although architectural advances involving YOLOv8, RT-DETR, YOLOv9, YOLOv10, feature pyramid networks (FPNs), bidirectional feature pyramid networks (BiFPNs), and channel-prior convolutional attention (CPCA) have improved detection performance [12–18], relying solely on the detector remains insufficient for consistently distinguishing actual rockfalls from rockfall-like background objects.

Handcrafted and deep features provide complementary information representations. The former explicitly characterize the texture, color, and morphological properties of candidate regions, whereas the latter are more effective at learning semantic and contextual relationships. These two types of features can be fused at either the feature or decision level. Compared with the direct concatenation of high-dimensional features, late fusion or multi-classifier ensemble strategies can better preserve the independent discriminative capabilities of individual models while alleviating the problems caused by inconsistent scales among heterogeneous features [19]. A feasible approach is therefore to use YOLO to rapidly localize candidate regions and then employ a conventional classifier to perform secondary verification based on features extracted from the candidate bounding boxes. However, existing studies on rockfall detection have rarely investigated the coordinated optimization of the YOLO candidate-confidence threshold and the machine-learning verification threshold. Quantitative analyses of the trade-off between improvements in precision and reductions in recall also remain limited.

Based on the above analysis, this study proposes a cascaded road rockfall detection method that integrates feature engineering with YOLO. In the YOLO branch, CPCA and learnable weighted multiscale feature fusion are introduced to enhance the representation of candidate targets under complex background conditions. Meanwhile, handcrafted features describing texture, color, edges, shape, frequency-domain characteristics, and morphology are extracted from the candidate regions and used by a machine-learning classifier for secondary verification. Through this cascaded strategy, false-positive detections caused by complex road backgrounds can be suppressed while maintaining a relatively high recall. The primary contribution of this study is the integration of interpretable handcrafted features with YOLO-based deep-learning detection results, providing a low-cost and readily deployable cascaded detection solution for road rockfall monitoring.

2. Overall Framework of the Proposed Model

To address the engineering pain point where a single YOLO model frequently generates false alarms in complex road scenarios, forcing maintenance personnel to spend considerable time eliminating each false positive one by one, this paper constructs a serial detection framework consisting of YOLO-based candidate detection, machine-learning-based secondary confirmation, and cascaded decision output. The overall workflow is illustrated in Figure 1. This framework adopts a serial cascading strategy, using YOLO as the candidate object generator and a machine learning classifier as the verifier: only after YOLO locates a suspected hazard region does the corresponding candidate region proceed to subsequent feature extraction and classification steps. This design aims to maintain YOLO's high candidate recall while leveraging interpretable local attributes to filter out background false targets, thereby concentrating limited manual verification efforts on genuine hazard alarms.

First, targeted improvements are made to YOLO for the rockfall detection task. The CPCA (Channel Prior Convolutional Attention) module is introduced into the Backbone of YOLOv8, and learnable weighted multi-scale feature fusion is adopted in the Neck to enhance the feature response of rockfall targets under complex backgrounds. After confidence threshold filtering and non-maximum suppression, YOLO outputs candidate bounding boxes, initial class predictions, and detection confidences, and crops the candidate regions from the original image according to the bounding box coordinates.

Subsequently, each candidate region is passed to the machine-learning verification module. This module extracts multiple categories of handcrafted features, including local binary pattern (LBP), gray-level co-occurrence matrix (GLCM), Gabor, edge, color, shape, frequency-domain, morphological, and texture-contrast features. The fused feature vector is then fed into the optimal classifier selected through comparisons among multiple models and evaluations of voting and stacking ensemble strategies. The classifier outputs the verified rockfall class and the corresponding machine-learning confidence score. The final prediction is determined using a serial gating rule: a candidate bounding box is retained as a final positive detection if and only if it is generated by YOLO, classified as a rockfall target by the machine-learning verifier, and assigned a verification confidence score no lower than the predefined threshold. Otherwise, the candidate is regarded as background interference and discarded. The YOLO and machine-learning confidence thresholds are jointly determined through a two-dimensional grid search to balance false-positive suppression against rockfall-target recall. After cascaded verification, the system outputs information on the presence or absence of rockfall hazards, the number of retained targets, and their spatial locations.

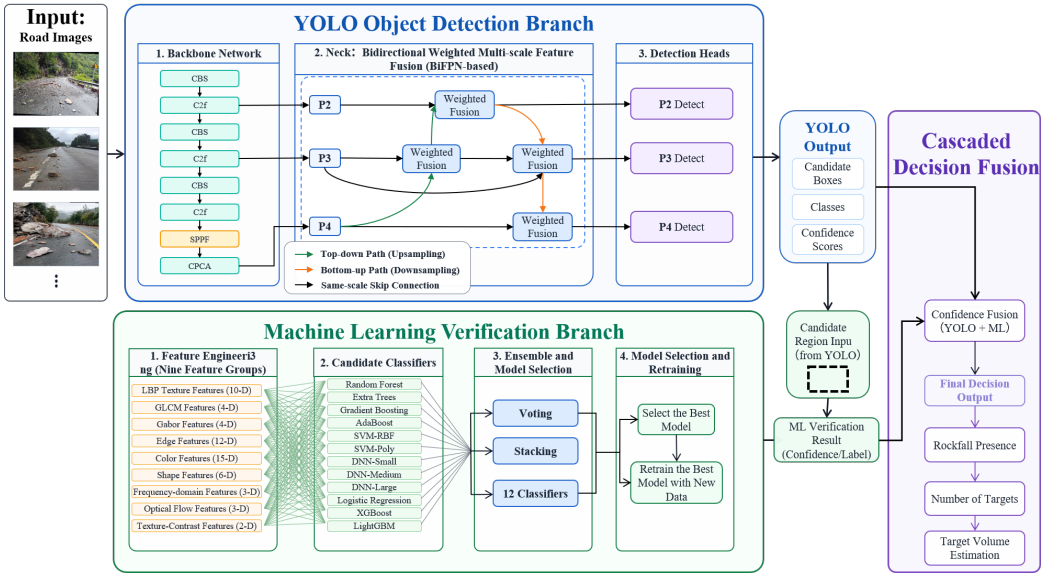


Figure 1. Overall framework of the proposed cascaded road rockfall detection method integrating an improved YOLO detector with machine-learning-based verification.

3. Design and Improvement of a Rockfall Detector Based on the YOLO Architecture

YOLO is responsible for the rapid localization of candidate disaster targets, and its outputs, including bounding boxes, class labels, and confidence scores, are used as inputs for subsequent machine learning-based verification and decision fusion. In this study, an improved detection network is constructed based on YOLOv8, which consists of three main components: the Backbone, Neck, and Detect Head. The Backbone progressively extracts visual features with different receptive fields through modules such as CBS, C2f, and SPPF. After obtaining high-level semantic features, the CPCA (Channel Prior Convolutional Attention) module is introduced to enhance the response to irregularly shaped rockfall targets and suppress interference from complex road backgrounds. In the Neck, a learnable weighted multi-scale feature fusion strategy is incorporated into the original top-down and bottom-up feature propagation paths, enabling more effective interaction between shallow-level spatial details and deep-level semantic information. The Detect Head retains the original three-scale outputs, which are responsible for classification and bounding box regression of disaster targets at different scales. The above improvements mainly focus on attention enhancement in the Backbone and feature fusion optimization in the Neck, providing more reliable candidate regions for subsequent cascade verification.

3.1. Channel-Prior Convolutional Attention Mechanism

Rockfall targets in road images are typically characterized by irregular shapes, substantial variations in surface texture, and indistinct boundaries. Meanwhile, background regions containing vehicles, shadows, vegetation, and pavement stains may produce local responses similar to those of actual rockfall targets, thereby becoming major sources of false-positive YOLO detections. Conventional convolutional operations may propagate background noise together with target-related features. To address this issue, a channel-prior convolutional attention (CPCA) module is incorporated after the high-level semantic feature extraction stage of the backbone, as shown in Figure 2a. Features at this stage possess relatively large receptive fields and strong semantic representation capabilities. Without altering the output structure of the detection head, the CPCA module jointly recalibrates the high-level features along the channel and spatial dimensions, enabling the

network to focus more effectively on informative regions associated with genuine rockfall targets while suppressing responses induced by background interference.

The CPCA module consists of two sequential components: channel attention (CA) and spatial attention (SA), as illustrated in Figure 2b. The CA component first evaluates the importance of individual feature channels using global statistical information, thereby establishing a channel prior. Under the guidance of this prior, the SA component subsequently employs multiscale depthwise convolutions to model both local and long-range spatial dependencies. Compared with attention mechanisms that use a single convolutional kernel or enforce shared spatial weights across all channels, CPCA provides more flexible spatial responses with relatively low additional computational overhead. This property is particularly beneficial for representing rockfall targets with substantial variations in scale and morphology.

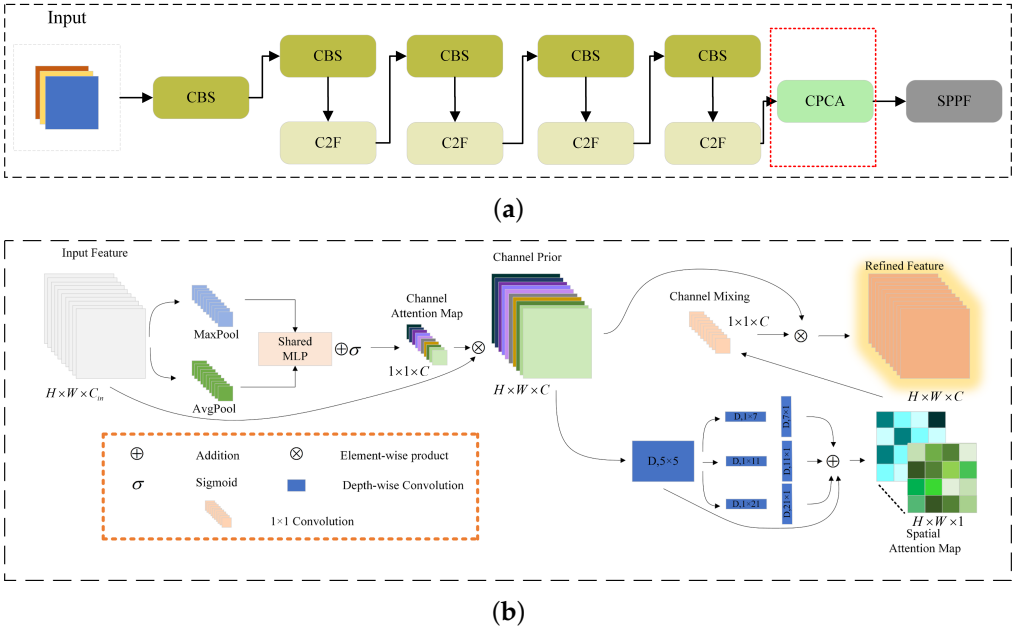


Figure 2. Channel-prior convolutional attention (CPCA) module: (**a**) insertion position of the CPCA module in the YOLOv8 backbone; (**b**) architecture of the CPCA module.

The channel attention (CA) module models the interdependencies among the feature channels. Given an input feature map $\mathbf{F}$, global average pooling and global max pooling are first applied along the spatial dimensions to aggregate global contextual information. The resulting two channel descriptors are subsequently passed through a shared multilayer perceptron (MLP). Their outputs are combined through element-wise summation and then processed using a sigmoid activation function $\sigma(\cdot)$ to generate the channel attention map $\mathbf{M}_c(\mathbf{F})$. Finally, the channel attention map is multiplied element-wise by the original feature map $\mathbf{F}$ to obtain the channel-refined feature map $\mathbf{F}_c$. This process is formulated as follows:

$$\mathbf{M}_c(\mathbf{F}) = \sigma[\mathrm{MLP}(\mathrm{AvgPool}(\mathbf{F})) + \mathrm{MLP}(\mathrm{MaxPool}(\mathbf{F}))], \tag{1}$$

$$\mathbf{F}_c = \mathbf{M}_c(\mathbf{F}) \odot \mathbf{F}, \tag{2}$$

where $\mathrm{AvgPool}(\cdot)$ and $\mathrm{MaxPool}(\cdot)$ denote global average pooling and global max pooling, respectively, and $\odot$ denotes element-wise multiplication.

The spatial attention (SA) module is designed to capture spatial dependencies within the feature map. Unlike the spatial attention component of the convolutional block attention module (CBAM), which applies a shared spatial attention distribution across channels, the SA component of CPCA dynamically models attention responses along both the channel

and spatial dimensions. Specifically, multiscale depthwise convolutions are applied to the channel-refined feature map $\mathbf{F}_c$ to capture local and long-range spatial dependencies while preserving the channel attention information. The outputs of the multiscale branches are aggregated through element-wise summation and subsequently processed using a 1×1 convolution to generate the spatial attention map $\mathbf{M}_s(\mathbf{F}_c)$. The spatial attention map is then multiplied element-wise by $\mathbf{F}_c$ to obtain the final refined feature map $\hat{\mathbf{F}}$. The process is expressed as

$$\mathbf{M}_s(\mathbf{F}_c) = \mathrm{Conv}_{1\times1}\left[\sum_{i=0}^{3} \mathrm{Branch}_i(\mathrm{DWConv}_i(\mathbf{F}_c))\right], \tag{3}$$

$$\hat{\mathbf{F}} = \mathbf{M}_s(\mathbf{F}_c) \odot \mathbf{F}_c, \tag{4}$$

where $\mathrm{DWConv}_i(\cdot)$ denotes the depthwise convolution operation in the ith multiscale branch, and $\mathrm{Branch}_i(\cdot)$ represents the corresponding multiscale convolutional branch. Through CPCA-based feature recalibration, the network can allocate greater representational capacity to potential rockfall regions, thereby providing more discriminative inputs for subsequent multiscale feature fusion and detection-head prediction.

3.2. Weighted Multiscale Feature Fusion

Rockfall targets in road surveillance images exhibit substantial variations in scale. Small targets located at long distances are prone to losing edge and positional information during successive downsampling operations, whereas large rockfall regions require stronger high-level semantic representations. Conventional neck structures generally aggregate features from different levels through direct concatenation or equal-weight fusion, making it difficult to dynamically adjust the contribution of each input branch according to the target scale and scene background. To address this limitation, nonnegative learnable weights are introduced into the top-down and bottom-up feature aggregation nodes of the neck.

The resulting fusion structure is illustrated in Figure 3. First, the high-level input feature $\mathbf{P}_4^{\mathrm{in}}$ is resized and adaptively fused with the intermediate-level input feature $\mathbf{P}_3^{\mathrm{in}}$ to generate the top-down intermediate feature $\mathbf{P}_3^{\mathrm{td}}$. Subsequently, $\mathbf{P}_3^{\mathrm{in}}$, $\mathbf{P}_3^{\mathrm{td}}$, and the resized high-resolution output feature $\mathbf{P}_2^{\mathrm{out}}$ are further aggregated through learnable weighted fusion to produce the intermediate-level output feature $\mathbf{P}_3^{\mathrm{out}}$. Similar fusion operations are applied at the remaining feature scales, thereby preserving both shallow spatial details and deep semantic information. Taking the $\mathbf{P}_3$ node as an example, its top-down intermediate feature and final output feature are calculated using Equations (5) and (6), respectively:

$$\mathbf{P}_3^{\mathrm{td}} = \mathrm{Conv}\left(\frac{w_1\mathbf{P}_3^{\mathrm{in}} + w_2\,\mathrm{Resize}\left(\mathbf{P}_4^{\mathrm{in}}\right)}{w_1 + w_2 + \varepsilon}\right), \tag{5}$$

$$\mathbf{P}_3^{\mathrm{out}} = \mathrm{Conv}\left(\frac{w_1'\mathbf{P}_3^{\mathrm{in}} + w_2'\mathbf{P}_3^{\mathrm{td}} + w_3'\,\mathrm{Resize}\left(\mathbf{P}_2^{\mathrm{out}}\right)}{w_1' + w_2' + w_3' + \varepsilon}\right). \tag{6}$$

In Equation (5), w_1 and w_2 denote the learnable weights assigned to $\mathbf{P}_3^{\mathrm{in}}$ and the resized $\mathbf{P}_4^{\mathrm{in}}$, respectively. In Equation (6), w_1', w_2', and w_3' represent the respective contributions of $\mathbf{P}_3^{\mathrm{in}}$, $\mathbf{P}_3^{\mathrm{td}}$, and the resized $\mathbf{P}_2^{\mathrm{out}}$. The fusion weights are first constrained to be nonnegative and are then normalized by the sum of all input weights at the same fusion node together with a stabilization term ε. In this study, ε is set to 0.0001 to prevent the denominator from approaching zero and to maintain training stability. Here, $\mathrm{Resize}(\cdot)$ denotes upsampling or downsampling performed to align the spatial dimensions of the input feature maps, whereas $\mathrm{Conv}(\cdot)$ denotes convolutional feature integration following weighted aggregation. All fusion weights are jointly optimized with the remaining network parameters through

backpropagation, enabling the network to automatically emphasize the feature branches that are more informative for the current detection scale.

Compared with fixed-weight fusion or indiscriminate feature concatenation, the proposed method explicitly establishes a contribution-allocation mechanism for input features at different scales. The high-resolution branch supplements the edge and localization information required for detecting small rockfall targets, whereas the high-level branch provides semantic representations and contextual constraints. Consequently, feature confusion caused by complex pavement textures, shadows, and vegetation backgrounds can be effectively alleviated.

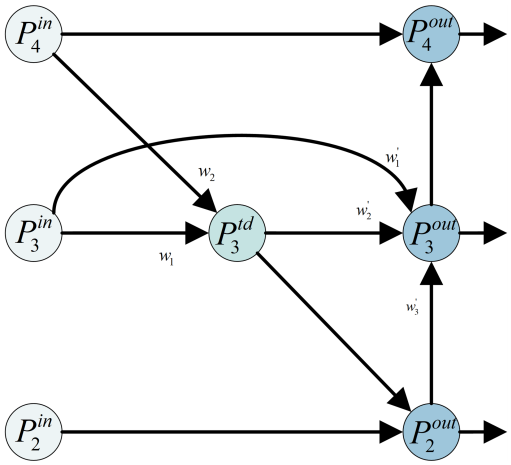


Figure 3. Architecture of the learnable weighted multiscale feature fusion module.

4. Design of a Machine Learning Verification Module Based on Hand-Crafted Features

The YOLO module is responsible for rapidly localizing suspected rockfall regions across the entire road image, while the machine learning module no longer performs independent sliding-window detection on the original image but instead conducts secondary confirmation solely on the candidate regions output by YOLO. Therefore, this module essentially functions as a candidate region classifier, with its input being Regions of Interest (ROIs) cropped from the original image according to the YOLO bounding boxes and resized to a uniform dimension, and its output being the class prediction and associated confidence indicating whether the candidate region belongs to a hazard target. This design preserves YOLO's localization capability while leveraging interpretable hand-crafted features to filter out false targets such as shadows, vehicles, vegetation, and road surface textures.

4.1. Multisource Handcrafted Feature Extraction and Representation

To characterize candidate targets from multiple perspectives, several categories of handcrafted features are extracted, including local binary pattern (LBP), gray-level co-occurrence matrix (GLCM), Gabor, edge, color, shape, frequency-domain, morphological, and texture-contrast features. Specifically, LBP and GLCM describe local texture patterns and spatial relationships between gray levels, whereas Gabor features characterize texture responses at different scales and orientations. Color features represent the overall color distribution within a candidate region, while edge and shape features describe target contours and geometric irregularities. Frequency-domain and morphological features further complement the representation of image structure and regional connectivity.

For the ith candidate region $\mathbf{R}_i$, the extracted feature groups are concatenated along the feature dimension to form the fused feature vector:

$$\mathbf{f}_i = \left[\mathbf{f}_i^{\mathrm{LBP}}, \mathbf{f}_i^{\mathrm{GLCM}}, \mathbf{f}_i^{\mathrm{Gabor}}, \mathbf{f}_i^{\mathrm{Color}}, \mathbf{f}_i^{\mathrm{Edge}}, \mathbf{f}_i^{\mathrm{Shape}}, \mathbf{f}_i^{\mathrm{Freq}}, \mathbf{f}_i^{\mathrm{Morph}}, \mathbf{f}_i^{\mathrm{Con}}\right]^{\mathrm{T}} \in \mathbb{R}^d, \tag{7}$$

where d denotes the total dimensionality of the fused feature vector. Because the magnitudes and numerical ranges of different feature categories may vary substantially, all features are standardized using statistics calculated exclusively from the training set before classifier training. The standardized value of the jth feature of the ith sample is defined as

$$z_{ij} = \frac{f_{ij} - \mu_j}{\sigma_j + \varepsilon}, \tag{8}$$

where μ_j and σ_j denote the mean and standard deviation of the jth feature calculated from the training set, respectively, and ε is a small positive constant introduced to prevent division by zero. The validation and test sets are transformed using the values of μ_j and σ_j obtained from the training set, thereby preventing data leakage. The resulting standardized feature vector $\mathbf{z}_i$ is used as the common input to all candidate classifiers.

4.2. Construction and Ensemble of Candidate Classifiers

To evaluate the suitability of different learning mechanisms for the fused handcrafted features, a candidate model library was constructed, including random forest, extremely randomized trees, gradient boosting trees, AdaBoost, support vector machines with different kernel functions, fully connected neural networks with different architectures, logistic regression, XGBoost, and LightGBM. Let the mth classifier be denoted by $\mathcal{M}_m$. For the ith candidate region, the posterior probability of the rockfall class predicted by $\mathcal{M}_m$ is expressed as

$$p_i^{(m)} = P(y_i = 1 \mid \mathbf{z}_i; \mathcal{M}_m), \qquad m = 1, 2, \ldots, M, \tag{9}$$

where $y_i = 1$ indicates that the candidate region is classified as a rockfall target, $\mathbf{z}_i$ denotes the standardized feature vector of the ith candidate region, and M is the total number of classifiers included in the comparison. All models were trained and validated using the same data partition and evaluation protocol. Their classification performance was comprehensively assessed in terms of accuracy, precision, recall, and the F_1-score.

To exploit the complementary discriminative capabilities of different classifiers, voting and stacking ensemble strategies were further constructed. In the voting ensemble, the posterior probabilities produced by the base classifiers are aggregated through weighted averaging:

$$p_i^{\mathrm{V}} = \frac{\sum_{m=1}^{M} w_m p_i^{(m)}}{\sum_{m=1}^{M} w_m}, \qquad w_m \geq 0, \tag{10}$$

where w_m denotes the voting weight assigned to the mth base classifier. When equal weights are assigned to all base classifiers, Equation (10) reduces to soft voting based on the arithmetic mean of their posterior probabilities.

In the stacking ensemble, the posterior probabilities generated by the base classifiers are combined to form a second-level feature vector, which is subsequently processed by a meta-learner $g(\cdot)$ to obtain the final classification probability:

$$p_i^{\mathrm{S}} = g\left(p_i^{(1)}, p_i^{(2)}, \ldots, p_i^{(M)}\right). \tag{11}$$

To reduce the risk of overfitting caused by directly training the meta-learner on predictions generated from the same training samples, the second-level training features used in the stacking framework were constructed from out-of-fold predictions obtained through cross-validation. After comparing the individual classifiers and ensemble models, the model with the best validation performance was selected as the candidate-region verifier. The selected model was then retrained using the complete training dataset to obtain the final machine-learning classification model.

4.3. Serial Decision Fusion

The final prediction is determined using a serial gating rule rather than by linearly combining the YOLO detection confidence and the machine-learning verification confidence. Let s_i^{Y} denote the detection confidence score of the ith candidate bounding box generated by YOLO, $c_i \in \{0,1\}$ denote the class label predicted by the machine-learning verifier, and s_i^{ML} denote the corresponding confidence score for the rockfall class. Furthermore, let τ_{Y} and τ_{ML} represent the YOLO candidate-confidence threshold and the machine-learning verification threshold, respectively. The final decision for the ith candidate bounding box is defined as

$$\hat{y}_i = \begin{cases} 1, & s_i^{\mathrm{Y}} \geq \tau_{\mathrm{Y}}, \quad c_i = 1, \quad \text{and} \quad s_i^{\mathrm{ML}} \geq \tau_{\mathrm{ML}}, \\ 0, & \text{otherwise}, \end{cases} \tag{12}$$

where $\hat{y}_i = 1$ indicates that the candidate bounding box is retained as a final rockfall detection. A candidate is retained only when it is generated by YOLO, classified as a rockfall target by the machine-learning verifier, and assigned a verification confidence score no lower than the predefined threshold. Otherwise, it is regarded as background interference and discarded.

The two confidence thresholds are jointly determined through a two-dimensional grid search. A relatively low value of τ_{Y} is used to preserve high candidate recall, whereas τ_{ML} controls the strictness of secondary verification. Their joint optimization therefore provides a trade-off between false-positive suppression and rockfall-target recall. The total number of retained rockfall targets is expressed as

$$N = \sum_{i=1}^{n} \hat{y}_i, \tag{13}$$

where n denotes the number of candidate bounding boxes generated by YOLO. The spatial location of each retained target is directly inherited from its corresponding YOLO bounding box. To further support the practical emergency response, the retained targets are further quantified by estimating their approximate volume. In the deployed system, the rockfall volume is estimated from the detected bounding box using a simplified pixel-to-meter conversion:

$$V = \frac{w_p h_p (w_p + h_p)}{2} \cdot s^3 \cdot C, \tag{14}$$

where w_p and h_p denote the pixel width and height of the detected bounding box, respectively, s is the corrected pixel-to-meter scale, and C is an empirical shape correction coefficient. The scale s is estimated from the assumed roadway width and a simple perspective correction, with $W_{\mathrm{road}} = 3.5$ m according to the standard lane width and $C = 0.52$ in the deployed system.

5. Experimental Results and Analysis

5.1. Experimental Environment

All experiments were conducted on a desktop workstation running Windows 11 (Microsoft, Redmond, WA, USA). The workstation was equipped with an Intel Core i9-13900KF processor (Intel, Santa Clara, CA, USA), an NVIDIA RTX A6000 GPU with 48 GB of video memory (NVIDIA, Santa Clara, CA, USA), and 64 GB of system memory. The proposed model was implemented using PyTorch 1.9.1 (Meta AI, Menlo Park, CA, USA), CUDA 11.7 (NVIDIA, Santa Clara, CA, USA), and cuDNN 7.6.5 (NVIDIA, Santa Clara, CA, USA).

5.2. Training Procedure

The cascaded framework was trained sequentially rather than in an end-to-end manner. First, the improved YOLO detector was trained on the annotated road rockfall dataset. After convergence, the YOLO parameters were fixed, and the trained detector was used to generate candidate regions. The machine-learning verifier was then trained separately using handcrafted features extracted from these candidate regions. Finally, the YOLO candidate-confidence threshold and the machine-learning verification threshold were jointly determined by grid search.

5.3. Data Sources and Annotation

The experimental images were collected from multiple sources, including publicly available online resources, historical archives provided by road administration departments, and on-site image acquisition. The collected images covered diverse illumination conditions, road backgrounds, target scales, and occlusion scenarios. The dataset used to train the YOLO model was annotated using LabelImg. Before data augmentation, the original dataset contained 619 unique road images. After annotation, the original images were randomly divided at the image level into training, validation, and test sets at a ratio of 8:1:1. Specifically, the original training set contained 495 images, while the validation and test sets contained 62 images each. Data augmentation operations, including noise injection, image rotation, and Cutout, were then applied only to the training set.

After augmentation, the training set was expanded to 1738 samples, whereas the validation and test sets remained unchanged. Therefore, the final dataset used for model training and evaluation contained 1738 training samples, 62 validation samples, and 62 test samples. The dataset annotation procedure is illustrated in Figure 4.



Figure 4. Examples of partial annotations from the YOLO dataset.

The dataset used to train the machine-learning models was organized into two folders, namely “rockfall” and “normal”. The “rockfall” folder contained images of genuine rockfall targets in road scenes, as shown in Figure 5a, whereas the “normal” folder contained road-scene images without rockfalls but involving complex background factors such as vehicles and shadows, as shown in Figure 5b.

It should be noted that the regions enclosed by the bounding boxes in the figure were obtained from actual monitoring results collected during the preliminary engineering

deployment stage. Specifically, the improved YOLO model was previously employed to continuously monitor road scenes and automatically save images containing suspected rockfall targets. Because real-world road environments contain numerous interfering objects with visual characteristics similar to those of rockfalls, the saved images included a certain number of false-positive samples. Among them, local regions of vehicles misclassified as rockfall targets represented the most typical type of false-positive detection. To address this issue, the false-positive images were manually screened, and the corresponding misdetected regions were cropped and added to the "normal" dataset as non-rockfall samples. This procedure was intended to reduce the false-positive detection rate and improve the model's ability to distinguish non-rockfall objects in complex road environments. The resulting dataset contained 1939 samples in the "rockfall" class and 1404 samples in the "normal" class.

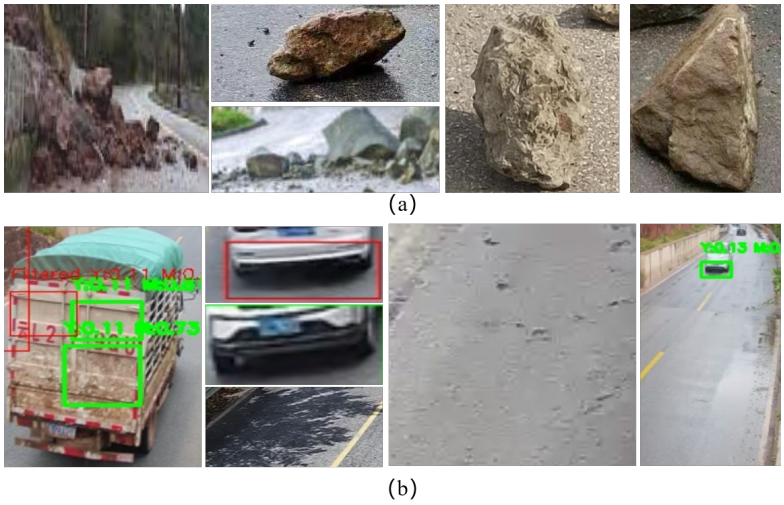


Figure 5. Representative samples in the machine-learning dataset: (**a**) genuine rockfall samples stored in the "rockfall" folder; (**b**) non-rockfall samples stored in the "normal" folder.

To comprehensively evaluate the performance of the proposed cascaded road rockfall detection model, four commonly used classification metrics were adopted: accuracy, precision, recall, and the F_1-score. These metrics assess the model from complementary perspectives, including overall classification performance, the reliability of positive detections, and the ability to avoid missed detections of actual rockfall targets.

Accuracy represents the proportion of correctly classified samples among all samples and is used to evaluate the overall classification capability of the model. It is defined as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{15}$$

where TP, TN, FP, and FN denote the numbers of true-positive, true-negative, false-positive, and false-negative samples, respectively. A higher accuracy indicates a stronger overall ability to distinguish rockfall targets from background regions.

Precision evaluates the reliability of positive detection results and represents the proportion of actual rockfall targets among all samples predicted as rockfalls. It is calculated as

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{16}$$

A higher precision indicates that the model produces fewer false alarms and can more effectively suppress false-positive detections caused by interfering factors such as vehicles, vegetation, and shadows in complex road environments.

Recall measures the ability of the model to detect actual rockfall targets and represents the proportion of correctly identified rockfall samples among all ground-truth rockfall samples. It is defined as

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{17}$$

In road hazard monitoring, a higher recall reduces the likelihood of missing critical rockfall targets and improves the system's ability to identify potential road safety hazards.

Because precision and recall emphasize false-positive and false-negative detections, respectively, a trade-off generally exists between these two metrics. Therefore, the F_1-score was employed to evaluate their overall balance:

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{18}$$

The F_1-score is the harmonic mean of precision and recall and therefore jointly reflects the reliability and completeness of the detection results. A higher F_1-score indicates that the model achieves a more favorable balance between false-positive suppression and missed-detection reduction.

In summary, accuracy, precision, recall, and the F_1-score were used to comprehensively evaluate the performance of the proposed road rockfall detection model and to verify the effectiveness of integrating the machine-learning verification module with the YOLO detection branch.

5.4. Optimization of Cascaded Decision Parameters and Analysis of Experimental Results

The effectiveness of CPCA and the learnable weighted multiscale feature fusion strategy, as well as comparisons with different YOLO versions, has been systematically evaluated in our previous work on road rockfall detection [10,20]. To evaluate the additional contribution of the machine-learning verification stage and the cascaded decision strategy, the standalone improved YOLO detector and the YOLO-ML cascaded detector were compared using the same test samples, ground-truth annotations, and evaluation protocol. The YOLO branch first generated candidate bounding boxes and their corresponding confidence scores, after which the machine-learning classifier performed secondary verification of each candidate region. The primary function of the cascaded framework is to filter out false-positive candidates caused by interfering factors such as vehicles, pavement textures, shadows, and vegetation. Therefore, its performance was evaluated primarily in terms of false-positive suppression and overall discrimination capability.

A two-dimensional grid-search strategy was adopted to jointly optimize the YOLO candidate-confidence threshold and the machine-learning classification-confidence threshold. The YOLO confidence threshold was set to 0.05, 0.10, 0.15, 0.20, and 0.25, whereas the machine-learning confidence threshold ranged from 0.05 to 0.95 with a step size of 0.05, resulting in a total of 95 parameter combinations. A candidate was retained as a final rockfall detection only when it was detected by YOLO, classified as a rockfall target by the machine-learning verifier, and assigned a machine-learning confidence score higher than the specified threshold.

For each parameter combination, the changes in precision, recall, and the F_1-score relative to the improved YOLO baseline were calculated. A composite improvement score was then defined as

$$S = 2\Delta\text{Precision} + 1.5\Delta\text{Recall} + \Delta F_1, \tag{19}$$

where a relatively large weight is assigned to precision to emphasize the ability of the cascaded classifier to suppress false-positive detections, while recall and the F_1-score are incorporated to preserve detection completeness and overall performance. The parameter

combinations were ranked according to the composite score S, and the configuration with the highest score was selected as the optimal setting. The parameter-search results are presented in Figure 6. The best performance was achieved when the YOLO confidence threshold was set to 0.25 and the machine-learning classification-confidence threshold was set to 0.75.

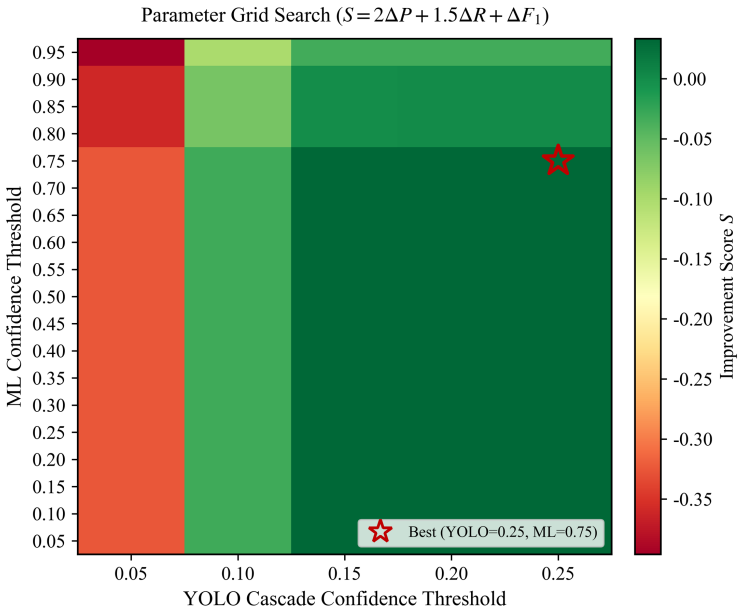


Figure 6. Grid-search results for the YOLO and machine-learning confidence thresholds based on the composite improvement score $S = 2\Delta\text{Precision} + 1.5\Delta\text{Recall} + \Delta F_1$. The red star indicates the optimal threshold combination ($\tau_Y = 0.25$ and $\tau_{ML} = 0.75$).

Figure 7 presents the quantitative results obtained using the optimal threshold configuration. Compared with the improved YOLO detector, the YOLO-ML cascaded detector increases accuracy from 93.3% to 94.1%, corresponding to an improvement of 0.8 percentage points. Precision increases from 90.1% to 92.6%, representing an improvement of 2.5 percentage points, while the F_1-score increases from 93.4% to 94.0%, corresponding to an improvement of 0.6 percentage points. Meanwhile, recall decreases from 97.0% to 95.5%, representing a reduction of 1.5 percentage points. The changes in the four evaluation metrics are summarized in Table 1.

To further analyze the discriminative ability of the cascaded detector, the ROC curve was plotted to evaluate the trade-off between false positives and true positives across different decision thresholds, as shown in Figure 8. The AUC value reaches 0.885, indicating that the proposed cascaded detector maintains robust discriminative capability between genuine rockfall targets and background false-positive candidates.

Table 1. Performance comparison between the improved YOLO detector and the YOLO-ML cascaded detector.

Metric	Improved YOLO (%)	YOLO-ML Cascade (%)	Change (Percentage Points)
Accuracy	93.3	94.1	+0.8
Precision	90.1	92.6	+2.5
Recall	97.0	95.5	−1.5
F_1-score	93.4	94.0	+0.6

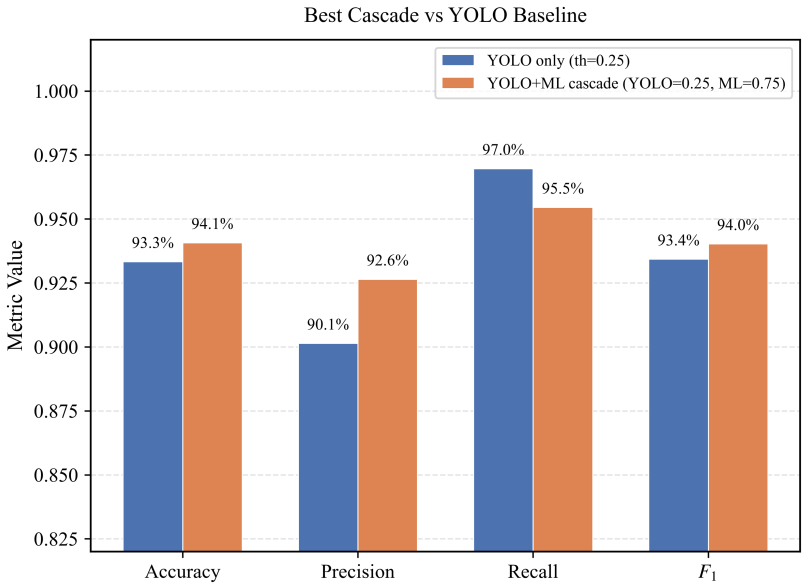


Figure 7. Performance comparison between the optimal YOLO-ML cascaded detector and the improved YOLO baseline.

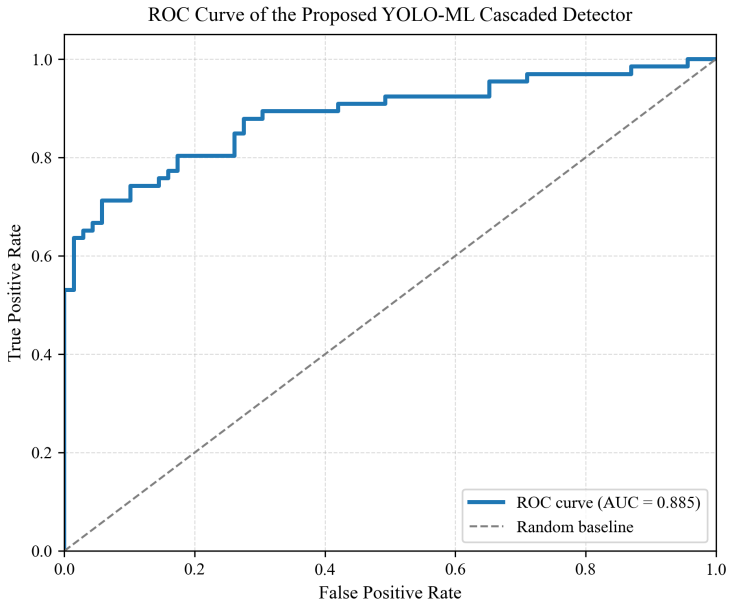


Figure 8. ROC curve of the proposed YOLO-ML cascaded detector. The curve illustrates the trade-off between the true-positive rate and false-positive rate under different decision thresholds, with an AUC value of 0.885.

The increase in precision indicates that the machine-learning verifier effectively re-evaluates the candidate bounding boxes generated by YOLO, thereby substantially reducing the proportion of false-positive detections. The 2.5-percentage-point improvement in precision is greater than the changes observed in the other metrics, suggesting that the principal benefit of the cascaded strategy lies in its ability to filter out false-positive candidates. This finding is consistent with the capability of handcrafted features to explicitly characterize local physical properties, such as texture, shape, and gray-level distribution.

The slight decrease in recall is an expected consequence of the serial cascaded architecture. Because each final positive detection must also satisfy the machine-learning verification threshold, a small number of challenging samples correctly detected by YOLO may be rejected during the secondary verification stage. Therefore, the proposed method

does not outperform the improved YOLO detector in every individual metric. Instead, while maintaining a relatively high recall, it exchanges a 1.5-percentage-point reduction in recall for a 2.5-percentage-point improvement in precision and increases accuracy and the F_1-score by 0.8 and 0.6 percentage points, respectively. These results demonstrate that the proposed cascaded strategy achieves a more favorable balance between false-positive suppression and missed-detection control, rather than merely optimizing a single performance metric.

In practical road monitoring, frequent false alarms increase the costs associated with manual verification and on-site response and may also reduce the credibility of the early-warning system. Therefore, improving precision and the F_1-score while maintaining a recall of 95.5% has direct engineering significance. For high-risk road sections where missed detections must be minimized, the machine-learning confidence threshold can be appropriately reduced to improve recall. By contrast, the current optimal threshold configuration is more suitable for scenes with substantial interference from traffic, shadows, or vegetation, where false alarms incur relatively high costs.

To qualitatively evaluate the performance of the cascaded model in real-world road scenes, two representative groups of rockfall images were selected for analysis, as shown in Figure 9. In the figure, Y, M, and F displayed above each bounding box denote the YOLO detection confidence, machine-learning classification confidence, and cascaded decision confidence, respectively. The number of detected rockfall targets and the estimated rockfall volume are also presented at the top of each image.

As shown in Figure 9a, the model accurately localizes the spatial extent of large, irregularly shaped rockfall targets distributed at relatively short distances. Figure 9b contains multiple spatially dispersed rockfall targets with substantial differences in scale and viewing distance, together with interference from rock faces, guardrails, pavement textures, and illumination variations. Despite these challenging background conditions, the model successfully identifies the principal rockfall targets on the road surface and provides separate detection results for targets at different scales. These results demonstrate that the proposed method possesses a certain degree of multiscale rockfall detection capability.



(a) (b)

Figure 9. Qualitative rockfall detection results of the proposed cascaded model: (**a**) detection of multiple rockfall targets at relatively short distances; (**b**) multiscale rockfall detection under complex background conditions.

5.5. Implementation and Field Deployment of the Rockfall Detection System

To facilitate the engineering application of the cascaded model proposed in this paper, we developed rockfall detection system software, the user interface of which is shown in Figure 10. The software integrates six functional modules: cloud-platform upload, camera control, detection control, video/image testing, real-time video monitoring, and system

logging. The cloud-platform upload module is used to configure MQTT parameters, including the server address, port number, topic, and device identifier. The camera control module allows the video-stream address to be specified and the connection status to be managed. The video/image testing module supports offline validation of the model and its parameter settings. The real-time video monitoring module displays on-site video, candidate bounding boxes, and cascaded verification results. The system logging module records device connections, model loading, detection events, and data-upload status, thereby facilitating system operation and maintenance as well as fault tracing.

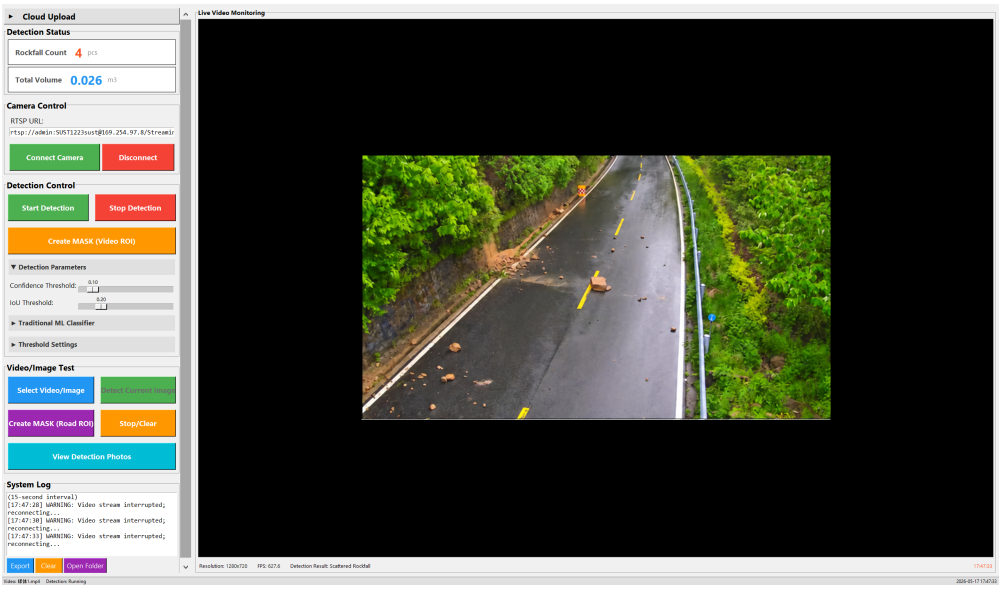


Figure 10. User interface of the developed road rockfall detection system.

In practical deployment, a road-region mask was constructed for each monitoring camera to restrict detection and verification to the effective roadway area. This strategy excludes irrelevant background regions such as mountainsides, vegetation, sky, and areas outside the guardrail, thereby reducing interference caused by different camera viewpoints and roadside backgrounds. As shown in Figure 11, the machine-learning verifier is applied only to candidate regions generated within the roadway mask rather than to the entire image. In general, the verifier does not need to be retrained for every new road section. However, when a new site exhibits substantially different pavement materials, camera parameters, or climatic conditions, a small number of site-specific samples can be collected for incremental updating or periodic retraining.

At present, the developed system has been deployed on several rockfall-prone road sections of China National Highways G108 and G210 in Xi'an, Shaanxi Province, China. The G108 monitoring sites are mainly located in Zhouzhi County, and the G210 monitoring sites are mainly located in Chang'an District. The monitored sections include G108 K1432, G108 K1433, G108 K1465, G210 K1317, and G210 K1339. On 17 May 2026, the monitoring system deployed at the G210 K1339 road section detected multiple rockfall targets intruding onto the roadway and issued a yellow-level rockfall warning. The on-site monitoring results and cloud-platform records are presented in Figure 12.

Figure 12a shows the original road-surveillance image, in which rockfall targets of different scales can be observed scattered across the traffic lane. Figure 12b presents the sequential detection results saved by the system, demonstrating that the model successfully localized and annotated the principal rockfall targets in the monitored scene. Figure 12c shows the temporal variations in the detected rockfall status and target count recorded by

the cloud platform, reflecting the evolution of the detection results over the duration of the event.



Figure 11. Example of road-region mask construction in the deployed system. The polygonal ROI restricts detection and machine-learning verification to the effective roadway area, thereby reducing interference from non-road background regions under different camera viewpoints and road scenes.

This field case demonstrates that the developed system can complete the entire workflow of image acquisition, model inference, target annotation, and warning-message generation using real-world road video streams. In addition, the system provides event records to support subsequent manual verification and on-site emergency response.

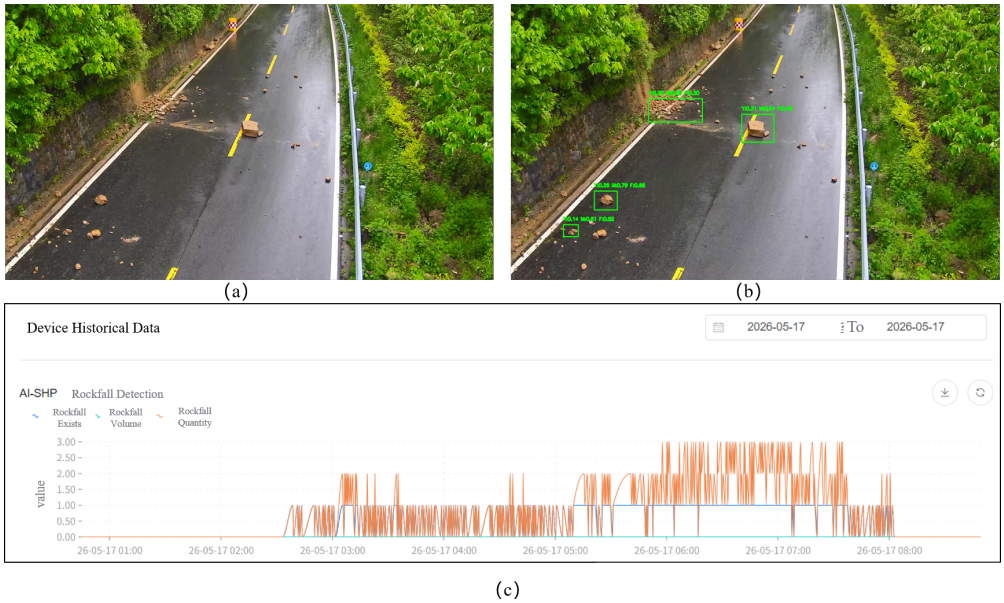


Figure 12. Field monitoring and warning case at the G210 K1339 road section: (**a**) original road-surveillance image; (**b**) sequential rockfall detection results; (**c**) historical rockfall monitoring records from the cloud platform.

6. Conclusions

Visual detection of road surface rockfalls in complex road environments is challenged by strong interferences such as shadows, vegetation, vehicles, and pavement textures.

A single YOLO model tends to generate a large number of false candidate bounding boxes, resulting in a high false alarm rate. To address this issue, this paper proposes a road surface rockfall detection method that integrates an improved YOLO model with feature engineering-based cascading verification.

The proposed method first introduces the CPCA channel prior attention mechanism and a learnable weighted multi-scale feature fusion strategy into the YOLO branch to enhance the representation capability of candidate targets. Subsequently, multiple handcrafted features, including texture, color, shape, edge, and frequency-domain features, are extracted from the candidate regions. A machine learning-based verification model is then constructed to perform secondary confirmation, forming a serial decision-making mechanism of "candidate detection–attribute verification."

Experimental results demonstrate that, compared with the baseline YOLOv8 model, the proposed cascaded model improves the accuracy from 93.3% to 94.1%, precision from 90.1% to 92.6%, and F_1-score from 93.4% to 94.0%, while the recall decreases slightly from 97.0% to 95.5%. By maintaining a high recall level, the proposed method improves precision by 2.5 percentage points and effectively suppresses false alarms caused by complex road environments.

Furthermore, this study independently developed a rockfall detection software system integrating image acquisition, real-time detection, parameter configuration, log recording, and warning upload functions. Field experiments were conducted on typical disaster-prone national highways, including G210 and G108, to validate the practical application value of the proposed system. To address the perception bottlenecks in complex environments, future research will introduce radar-vision fusion technology to compensate for visual deficiencies under strong-light, low-light, and adverse weather conditions, thereby further improving the accuracy and robustness of rockfall monitoring.

Author Contributions: Conceptualization, P.P. and J.L.; methodology, T.N., X.L. and P.P.; software, T.N. and X.L.; validation, T.N., X.L. and C.L.; formal analysis, X.L. and P.P.; investigation, Z.Q., C.L. and Y.D.; resources, Z.Q., C.L. and J.L.; data curation, T.N., X.L. and Y.D.; writing—original draft preparation, T.N., X.L. and P.P.; writing—review and editing, Z.Q., P.P. and J.L.; visualization, T.N. and X.L.; supervision, P.P. and J.L.; project administration, Z.Q., P.P. and J.L.; funding acquisition, Z.Q., P.P. and J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Fujian Provincial Science and Technology Major Special Project, grant number 2025HZ041001, and by the 2025 Transportation Research Program of the Department of Transport of Shaanxi Province, grant numbers 25-10K and 25-06K.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Research data can be obtained by contacting us at pengpeng@sust.edu.cn.

Acknowledgments: During the preparation of this manuscript, the authors used GPT-5.6 (OpenAI, San Francisco, CA, USA) for English-language polishing and stylistic refinement. The tool was used solely to improve the clarity, grammar, and readability of the manuscript and was not used for study design, data collection, data analysis, or interpretation of the results. The authors reviewed and edited all AI-assisted output and take full responsibility for the content of this publication.

Conflicts of Interest: Authors Zhiqing Qin, Caijin Lu and Yongsheng Dai were employed by the company Fujian Provincial Transportation Planning and Design Institute Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

YOLO	You Only Look Once
ML	Machine Learning
CPCA	Channel-Prior Convolutional Attention
CA	Channel Attention
SA	Spatial Attention
CBS	Convolution–Batch Normalization–SiLU
SPPF	Spatial Pyramid Pooling—Fast
NMS	Non-Maximum Suppression
ROI	Region of Interest
LBP	Local Binary Pattern
GLCM	Gray-Level Co-Occurrence Matrix
MLP	Multilayer Perceptron
SVM	Support Vector Machine
MQTT	Message Queuing Telemetry Transport

References

1. Yang, Z.-Q.; Qi, W.-W.; Xu, C.; Shao, X.-Y. Exploring deep learning for landslide mapping: A comprehensive review. *China Geol.* **2024**, *7*, 330–350. [CrossRef]
2. Farmakis, I.; DiFrancesco, P.M.; Hutchinson, D.J.; Vlachopoulos, N. Rockfall detection using LiDAR and deep learning. *Eng. Geol.* **2022**, *309*, 106836. [CrossRef]
3. Zoumpekas, T.; Puig, A.; Salamó, M.; García-Sellés, D.; Blanco Núñez, L.; Guim, M. An intelligent framework for end-to-end rockfall detection. *Int. J. Intell. Syst.* **2021**, *36*, 6471–6502. [CrossRef]
4. Abaker, M.; Dafaalla, H.; Eisa, T.A.E.; Abdelgader, H.; Mohammed, A.; Burhanur, M.; Hasabelrsoul, A.; Alfakey, M.I.; Morsi, M.A. Deep learning-and IoT-based framework for rock-fall early warning. *Appl. Sci.* **2023**, *13*, 9978. [CrossRef]
5. Novellino, A.; Pennington, C.; Leeming, K.; Taylor, S.; Alvarez, I.G.; McAllister, E.; Arnhardt, C.; Winson, A. Mapping landslides from space: A review. *Landslides* **2024**, *21*, 1041–1052. [CrossRef]
6. Li, B.; Li, J. Methods for landslide detection based on lightweight YOLOv4 convolutional neural network. *Earth Sci. Inform.* **2022**, *15*, 765–775. [CrossRef]
7. Hou, H.; Chen, M.; Tie, Y.; Li, W. A universal landslide detection method in optical remote sensing images based on improved YOLOX. *Remote Sens.* **2022**, *14*, 4939. [CrossRef]
8. Han, Z.; Fang, Z.; Li, Y.; Fu, B. A novel Dynahead-Yolo neural network for the detection of landslides with variable proportions using remote sensing images. *Front. Earth Sci.* **2023**, *10*, 1077153. [CrossRef]
9. Guo, H.; Yi, B.; Yao, Q.; Gao, P.; Li, H.; Sun, J.; Cheng, Z. Identification of landslides in mountainous area with the combination of SBAS-InSAR and Yolo model. *Sensors* **2022**, *22*, 6235. [CrossRef] [PubMed]
10. Peng, P.; Gao, L.; Li, J.; Zhang, H. Optimized YOLOv8 framework for intelligent rockfall detection on mountain roads. *Sci. Rep.* **2025**, *15*, 14007. [CrossRef] [PubMed]
11. Su, G.; Huang, W.; Li, C.; Huang, X. Rockfall-YOLO deep learning model for rockfall motion object detection. *J. Comput. Civ. Eng.* **2025**, *39*, 04025104. [CrossRef]
12. Terven, J.; Córdova-Esparza, D.M.; Romero-González, J.A. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Mach. Learn. Knowl. Extr.* **2023**, *5*, 1680–1716. [CrossRef]
13. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. DETRs Beat YOLOs on Real-Time Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 16965–16974.
14. Wang, C.Y.; Yeh, I.H.; Liao, H.Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Milan, Italy, 29 September–4 October 2024; pp. 1–10.
15. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 107984–108011. [CrossRef]
16. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.

17. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
18. Huang, H.; Chen, Z.; Zou, Y.; Lu, M.; Chen, C.; Song, Y.; Zhang, H.; Yan, F. Channel Prior Convolutional Attention for Medical Image Segmentation. *Comput. Biol. Med.* **2024**, *178*, 108784. [CrossRef] [PubMed]
19. Choudhury, K.; Murugan, R.; Azharuddin Laskar, M.; Laskar, R.H. A comparative analysis between late fusion of features approach and ensemble of multiple classifiers approach for image classification. *Concurr. Comput. Pract. Exp.* **2021**, *33*, e6371. [CrossRef]
20. Peng, P.; Gao, L.; Li, J. Road rockfall detection method based on enhanced Yolov8-GCB. *J. Chang. Univ. (Natural Sci. Ed.)* **2025**, *45*, 24–35.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.