



Article

Precision and Efficiency in Dam Crack Inspection: A Lightweight Object Detection Method Based on Joint Distillation for Unmanned Aerial Vehicles (UAVs)

Hangcheng Dong ^{1,†}, Nan Wang ^{2,†} , Dongge Fu ¹, Fupeng Wei ³ , Guodong Liu ¹ and Bingguo Liu ^{1,*}

¹ School of Instrumentation Science and Engineering, Harbin Institute of Technology, Harbin 150001, China; hunsen_d@hit.edu.cn (H.D.); 1190302620@stu.hit.edu.cn (D.F.); lgd@hit.edu.cn (G.L.)

² School of Information Science and Technology, Hainan Normal University, Haikou 571158, China; nanwang.ac@hainnu.edu.cn

³ School of Information Engineering, North China University of Water Resources and Electric Power, Zhengzhou 450045, China; weifupeng@ncwu.edu.cn

* Correspondence: liu_bingguo@hit.edu.cn

† These authors contributed equally to this work.

Abstract: Dams in their natural environment will gradually develop cracks and other forms of damage. If not detected and repaired in time, the structural strength of the dam may be reduced, and it may even collapse. Repairing cracks and defects in dams is very important to ensure their normal operation. Traditional detection methods rely on manual inspection, which consumes a lot of time and labor, while deep learning methods can greatly alleviate this problem. However, previous studies have often focused on how to better detect crack defects, with the corresponding image resolution not being particularly high. In this study, targeting the scenario of real-time detection by drones, we propose an automatic detection method for dam crack targets directly on high-resolution remote sensing images. First, for high-resolution remote sensing images, we designed a sliding window processing method and proposed corresponding methods to eliminate redundant detection frames. Then, we introduced a Gaussian distribution in the loss function to calculate the similarity of predicted frames and incorporated a self-attention mechanism in the spatial pooling module to further enhance the detection performance of crack targets at various scales. Finally, we proposed a pruning-after-distillation scheme, using the compressed model as the student and the pre-compression model as the teacher and proposed a joint distillation method that allows more efficient distillation under this compression relationship between teacher and student models. Ultimately, a high-performance target detection model can be deployed in a more lightweight form for field operations such as UAV patrols. Experimental results show that our method achieves an mAP of 80.4%, with a parameter count of only 0.725 M, providing strong support for future tasks such as UAV field inspections.

Keywords: dam crack inspection; object detection; lightweight model; knowledge distillation



Citation: Dong, H.; Wang, N.; Fu, D.; Wei, F.; Liu, G.; Liu, B. Precision and Efficiency in Dam Crack Inspection: A Lightweight Object Detection Method Based on Joint Distillation for Unmanned Aerial Vehicles (UAVs). *Drones* **2024**, *8*, 692. <https://doi.org/10.3390/drones8110692>

Academic Editor: Pablo Rodríguez-González

Received: 29 September 2024

Revised: 2 November 2024

Accepted: 15 November 2024

Published: 19 November 2024



Copyright: © 2024 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Dams are susceptible to various factors such as natural environments and water scouring during their operation, which can more readily lead to the development of cracks and other forms of damage [1]. If these cracks are not detected and remedied promptly, they may progressively worsen, diminishing the structural integrity of the dam. Under extreme weather conditions, these cracks can rapidly expand, potentially leading to catastrophic dam failures [2]. Therefore, the detection of dam cracks is of significant importance for ensuring the structural integrity and operational safety of the dam.

To address the aforementioned challenges, this paper proposes an intelligent detection scheme integrated with unmanned aerial vehicle (UAV) inspections as an alternative to the previously labor-intensive and inefficient manual inspection methods [3]. UAVs, with their

compact size and maneuverability, can perform complex and hazardous tasks that would otherwise require human intervention [4]. More importantly, their high degree of freedom and portability allows them to navigate over various terrains, enabling a rapid response to diverse inspection needs. To achieve the goal of automated inspection, this study also introduces a lightweight crack detection method designed for automatic identification and target analysis.

Drones, as convenient aerial units, have long been applied in various industries for inspection tasks [5], such as power infrastructure maintenance [6], assisting archaeology and cultural heritage [7], agriculture [8], and the energy sector [9]. Regarding dam crack detection, some studies have designed algorithms for low-resolution defect images [10,11], but they have not considered directly processing high-resolution remote sensing images from drones, nor have they further focused on field operations under limited computing resources.

In this study, deep neural networks are employed to implement dam surface crack defect detection for high-definition images captured by UAVs. In order to construct a high-performance and easily deployable crack defect detection algorithm, we first used a UAV to collect a crack dataset from a key reservoir in the South-to-North Water Transfer Central Project. A key challenge faced is that the resolution of the UAV images is high, reaching 6000×4000 , while the crack defects account for a very small proportion, only ranging from a dozen to several dozen pixels. In situations where the region of interest accounts for a small proportion, it would be difficult for training to converge, and performance would be poor if the original image is directly input into the network. Therefore, we design a moving window to preserve the crack defect features while maintaining partial overlap. Additionally, to ensure that the final defect detection on the original image does not result in false positives, we propose a corresponding fusion method to eliminate redundant frames.

After obtaining data with appropriate resolution and sufficient quantity, we propose a lightweight model using a distillation-after-pruning approach. First, we enhance the performance of the object detection model as much as possible, then prune the high-performance model, and finally restore accuracy through model distillation. Specifically, we first improve the object detection model based on YOLO_v5. To address the issue of varying image scales, we introduce a Gaussian distribution in the loss function to calculate the similarity of prediction boxes, achieving the comprehensive detection of targets of different sizes, while also making the regression boxes more accurate. In response to situations where target features are weak and there is much interference from background information, we introduce a self-attention mechanism into the spatial pooling module, which better enables the recognition of diverse features. Then, we use layer-adaptive pruning to reduce the number of parameters as much as possible to facilitate model deployment. Finally, to enhance the performance of the compressed model, we propose a joint distillation method to address the issue that models that have been overly compressed are difficult to distill. In summary, the key contributions of this work can be encapsulated as follows:

- This paper proposes an intelligent inspection process that combines UAV patrol with object detection and constructs a dataset of UAV aerial photography images;
- We propose a novel lightweight and high-performance object detection scheme tailored to the characteristics of UAV inspection tasks.
- We propose a joint distillation method to enhance the performance of compressed models, alleviating the issue where traditional distillation methods struggle to adapt to overly compressed models.

2. Related Works

2.1. Crack Detection

Crack detection is a critical aspect of maintaining the integrity and safety of surfaces in pavements and concrete structures [12]. The degradation of these surfaces due to cracking can pose safety hazards and reduce the service life of the structures. Traditional methods of assessing structural health, such as visual inspection, are costly, labor-intensive,

economically inefficient, and inconsistent in terms of effectiveness. With the advancement of deep learning technology, image-based computer vision techniques have emerged as a promising solution for the automated detection of cracks [13–15].

Ref. [16] pioneered the application of deep learning to crack detection, identifying five categories of cracks in concrete images. Ref. [17] compared crack detection methods using handcrafted features with those using deep architectures, evaluating the performance of various edge detectors such as Sobel, Canny, Prewitt, and Butterworth against the application of deep learning frameworks. Kim et al. [18] conducted a comparative study on the use of Hessian matrix and Haar wavelet-based accelerated robust feature methods and the application of convolutional neural networks (CNNs) for automatic feature extraction in crack detection tasks. Li et al. [19] designed a multi-scale defect region proposal network (RPN) that extracts candidate boxes at different layers to improve detection accuracy. They also used ultra-deep architectures and geographically tagged image databases for crack geolocation. Ref. [20] improved the R-CNN and proposed CrackDN, which uses a pre-trained CNN to extract deep features, along with a sensitivity detection network, achieving a faster training process. They integrated two additional modules into the Faster R-CNN architecture to reduce the false detection rate. The first module is a deep CNN responsible for estimating the direction of the identified crack patches. The second module is a Bayesian integration algorithm that uses local consistency and reduces the uncertainty of the detected crack patches. Maeda et al. employed the SSD framework [21], using Inception V2 [22] and MobileNet [23] as the main feature extraction modules in the SSD framework.

In addition to two-stage object detection algorithms, one-stage object detection algorithms focus more on inference speed and are more suitable for fields that require higher inference speeds. The representative method is the You Only Look Once (YOLO) algorithm. YOLOv3 [24], capable of performing target detection tasks at a significantly faster processing time with comparable accuracy to other methods, is a representative algorithm of the YOLO family. Mandal et al. utilized YOLO for crack detection in various datasets, including different types of cracks and defects [25]. Fan et al. [26] proposed a modified version of the UNet, incorporating dilated convolution modules and hierarchical feature learning modules to enhance the performance of crack detection. They introduced a method known as CrackGAN, drawing on the concept of generative adversarial networks, and utilized an asymmetric U-Net network as the backbone architecture to process images of arbitrary sizes. SegNet, an encoder–decoder architecture, employs VGG-16 to form the encoder–decoder. During the decoding process, maximum pooling indices are invoked for upsampling on the decoder module. This makes SegNet faster than UNet. Chen et al. [27] proposed a road and bridge crack segmentation network inspired by the SegNet [28] architecture, achieving end-to-end detection.

In addition to the crack detection methods from different scenarios mentioned in the above literature [10,11] directly address the study of cracks in dams. Ref. [10] considers the issue of precision and speed in crack detection, which is extremely important for the problem of dam crack detection. Ref. [11] investigates the role of Vision Transformers in dam crack detection. However, these studies are all based on cropped crack images and do not consider the convenience and challenges of direct detection based on unmanned aerial vehicle (UAV) remote sensing imagery.

2.2. Object Detection

The concept of object detection algorithms refers to the ability to not only delineate the location of objects in an image with bounding boxes but also to classify those objects, determining their categories [29]. This makes object detection algorithms highly suitable for crack detection tasks. Object detection algorithms can generally be divided into one-stage and two-stage algorithms. RCNN [30] is a classic two-stage object detection algorithm that uses selective search methods to extract a large number of candidate boxes and uses a CNN network to extract features from each candidate region in the corresponding image area. It then employs an SVM classifier to detect whether an object exists in each region. Two-stage

algorithms typically perform better but are slower in inference speed, even though some research attempts to improve on this. Therefore, one-stage object detection algorithms are more widely used in production environments that emphasize inference speed. One-stage algorithms eliminate the candidate box generation and use a single network to achieve feature extraction, object regression, and prediction. YOLO [31] is a single-stage object detector designed to treat object detection as a regression problem. It uses a limited number of candidate regions, widely uses features from the entire image, directly predicts the coordinates of the object bounding boxes, and calculates the probabilities of belonging to various categories. It has spawned a series of classic algorithms such as YOLOv3, YOLOv5, and YOLOv8. SSD is a fast single-shot multi-box detector suitable for multi-category detection [32]. It constructs a unified detection framework with detection speeds comparable to YOLO and accuracy on par with Faster RCNN.

2.3. Knowledge Distillation

Knowledge distillation is a process that involves using a large network to guide the training of a smaller network, abstracting knowledge from the larger model to impart it to the smaller one, thereby enabling the smaller model to approach or even surpass the performance of the larger model. This method does not alter the original structure of the network, effectively achieving model compression indirectly. Wang et al. pointed out that due to the inconsistency in optimization objectives, there is a significant difference between the predictions of the teacher and student models, leading to contradictory supervisory information during the imitation process by the student model [33]. To address this, the intermediate features of the student model's detection head are passed to the teacher's detection head, and the cross-head predictions are then forced to mimic the teacher's predictions. This prevents the student from receiving contradictory information from both the true labels and the teacher's predictions, thereby significantly enhancing the detection performance of the student model. Ref. [34] proposed decoupled knowledge distillation, reconstructing the classical knowledge distillation loss into two parts: target category knowledge distillation and non-target category knowledge distillation. This is because the logical information of non-target categories contains important "dark knowledge", whereas in traditional logical distillation, the importance of these two types of losses is coupled.

Knowledge distillation has transitioned from logical distillation to feature distillation. Shu C. et al. proposed channel-level knowledge distillation, which, unlike previous methods that aligned feature maps in the spatial domain, this method normalizes the feature maps of each channel to obtain soft probability maps. By calculating the KL divergence between the channel probability maps of the two networks, the differences between the two networks are measured, focusing on the most significant areas in each channel through the minimization of KL divergence, achieving knowledge transfer, which is suitable for dense detection tasks [35]. Ref. [36] proposed a simple and effective distillation strategy. Directly aligning the feature maps of the teacher and student models may impose too strict constraints on the student model, leading to a decline in performance. The authors found that aligning the feature maps of the teacher and student models along the channel dimension is equally effective in solving the problem of feature mismatch, using a multi-layer perceptron (MLP) to align the features of the student and teacher models. Ref. [37] proposed an attention-guided feature distillation method for semantic segmentation tasks. Drawing on the idea of the CBAM convolution block attention module, it is divided into CAM and SAM two modules, which, respectively, refine key information on the feature map from the spatial and channel dimensions while maintaining computational efficiency and suppressing the interference of background noise.

3. Methods

3.1. Overall Scheme

In this paper, we have established a set of automatic defect detection methods for dam cracks that can be combined with unmanned aerial vehicles (UAVs). The overall

research plan of this paper is shown in the Figure 1. First, we collected a sufficient amount of UAV aerial photography data at the target dam using UAVs. Then, the object detection model was trained and optimized on the server. Next, we proposed a pruning-first and then distillation scheme to compress the already optimized model while maintaining comparable performance. Finally, our lightweight model can be deployed online or offline for subsequent field inspection operations. The following sections of this paper will introduce the corresponding key technologies in order. Sections 3.2 and 3.3 introduce the details of data collection and construction, Section 3.4 explains in detail how to optimize the object detection model, Section 3.5 shows the network pruning method used, and Section 3.6 explains in detail the knowledge distillation technology proposed.

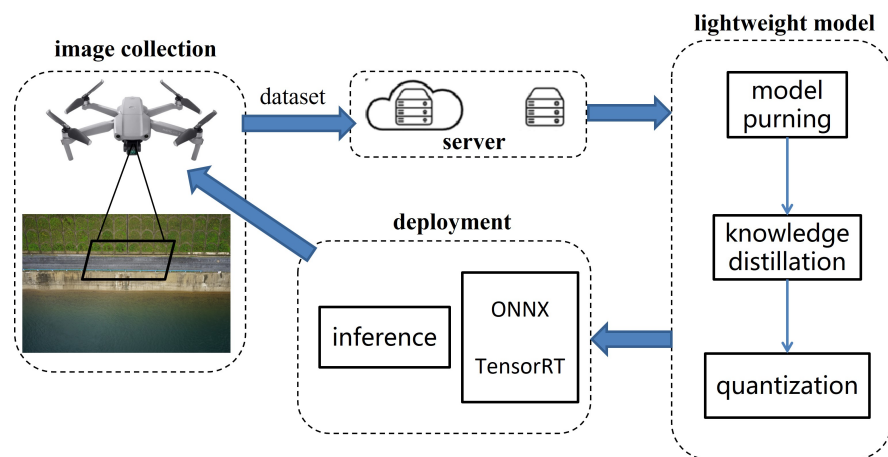


Figure 1. Flowchart of the overall scheme of this work.

3.2. Data Collection and Dataset Construction

The dam crack dataset, collected by unmanned aerial vehicle (UAV) aerial photography, primarily focuses on a specific area of the South-to-North Water Diversion Project. It is a dataset that includes various images of surface cracks. Due to the characteristics of the ultra-high-definition camera on the UAV, the aerial photographs are high-resolution images. As shown in Figure 2, the size of the original data for each image is 6000×4000 RGB color images saved in JPG format. The dataset contains comprehensive, rich, and clear high-definition crack samples.

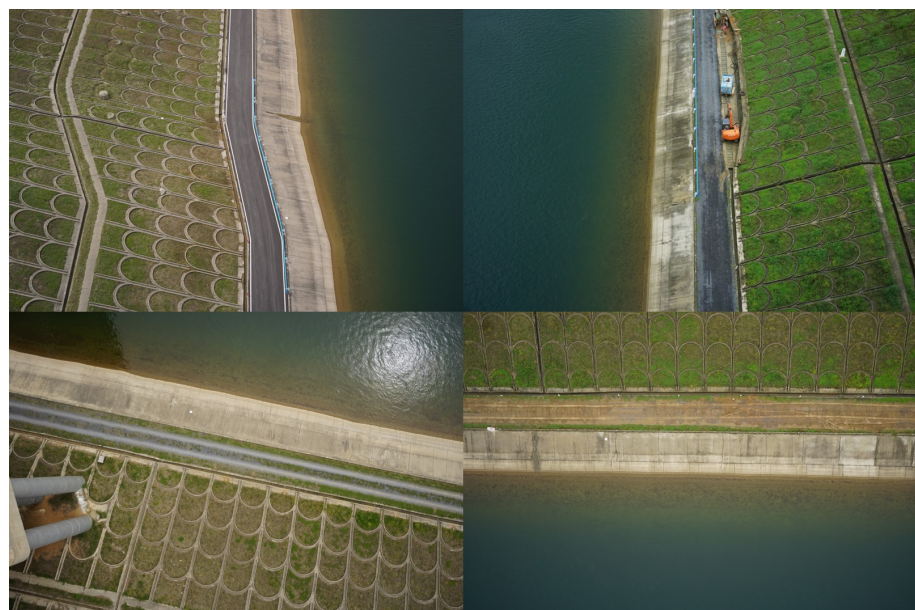


Figure 2. Remote sensing images captured by unmanned aerial vehicles.

3.3. Preprocessing of Remote Sensing Images

The images directly captured by drones have a size of 6000×4000 , while the crack areas that need to be detected are generally ten or even tens of pixels, indicating that the objects of interest are very small and scattered, and the targets occupy a very small proportion in the image. Directly inputting images into the network results in poor training results and an unstable convergence process. However, if downsampling methods are used, a lot of detail will be lost, affecting the accuracy of the detection results. Therefore, a sliding window method is used to cut and map the original image.

First, a window of a specified size is set. During the process of sliding and cutting the high-resolution image, small targets may be chopped, as shown in Figure 3. Therefore, an overlap rate can be set to make adjacent sub-graphs have overlapping parts, which can better solve the problem of small targets being divided. However, due to the characteristics of the cracks, there may still be situations where the target box on the cut sub-graph is not complete. A specified IOU value is set, and when the IOU value of the new target box and the target box on the original image is greater than a certain value, the sub-graph target box information is saved; if it is less than the value, it is removed. When there is no target box object on the sub-graph label, the graph is directly removed from the dataset.

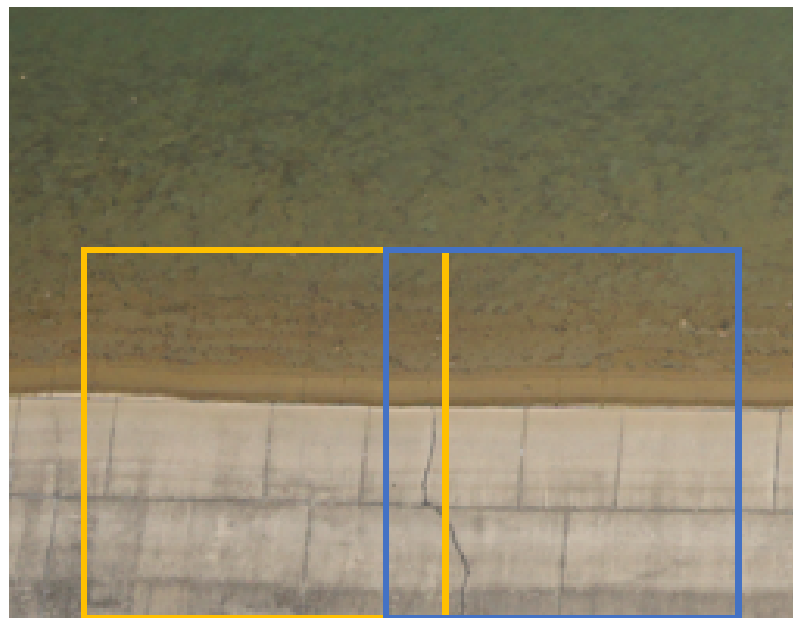


Figure 3. Schematic diagram of overlapping sliding window cutting.

During the inference phase, the process commences with the prediction of sub-images, followed by the employment of post-processing algorithms to stitch and restore the original image. For object detection results obtained from the sub-images, the coordinates are mapped back to the entire image. Subsequently, overlapping bounding boxes for the detected objects will appear in the overlapping regions. Non-maximum suppression (NMS) is then applied to eliminate redundant bounding boxes.

3.4. Improved YOLO_v5 Algorithm

3.4.1. Backbone

YOLOv5 is a widely used object detection model [31], which we use as the base model, and the network structure is shown in Figure 4. The network structure mainly includes the backbone network, prediction head, and neck network. The backbone, implemented by stacking multiple layers of the same module, can be used to extract features from the input dam images. The neck network is responsible for fusing the multi-scale features extracted by the backbone and passing these features to the prediction layer. The prediction

head is responsible for the final regression prediction, outputting the category information, location information, and confidence information of the cracks.

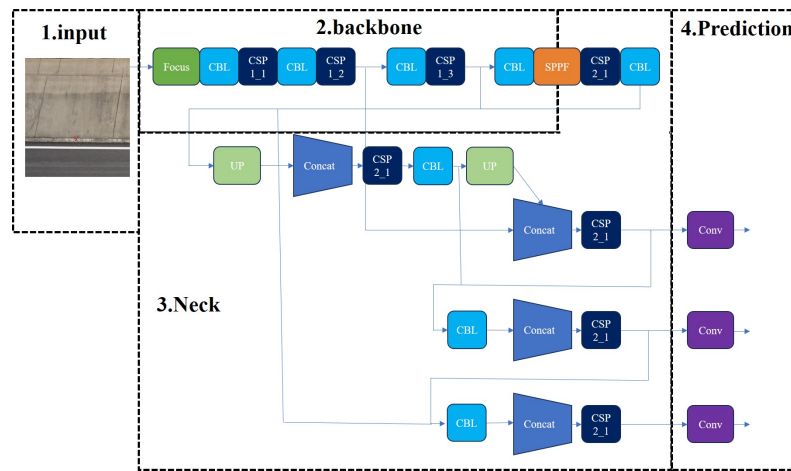


Figure 4. The network composition of YOLO_v5.

The feature extraction network structure in YOLO_v5 is modularly designed, and these networks are relatively lightweight, ensuring high detection accuracy while minimizing computational load and memory usage. The feature extraction network of YOLO_v5 consists of a total of 50 layers, and the overall network structure mainly includes the backbone network, prediction head, and neck network. The backbone, which is implemented by stacking multiple layers of the same module, can be used to extract features from the input dam images. The neck network is responsible for fusing the multi-scale features extracted by the backbone and passing these features to the prediction layer. The prediction head is responsible for the final regression prediction, outputting the category information, location information, and confidence level of the cracks. Figure 5 shows the structure of the feature extraction backbone.

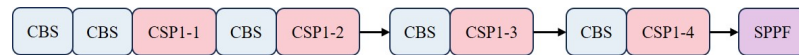


Figure 5. The structure of the feature extraction backbone in YOLO_v5.

3.4.2. Gaussian Distribution Loss Function

The dam’s cracks vary in size, and the differences can be enormous, as shown in Figure 6. To address this challenge, we analyzed the impact of this issue on object detection algorithms. The Intersection over Union (IoU) is commonly used to assess the similarity between two bounding boxes. Compared to larger objects, smaller objects, which occupy only a small fraction of the image space in terms of size and pixel area, are more susceptible to interference from adjacent objects or background noise. This leads to a reduction in the accuracy of similarity calculations, as illustrated in Figure 7.



Figure 6. Dam cracks vary widely in shape and size.

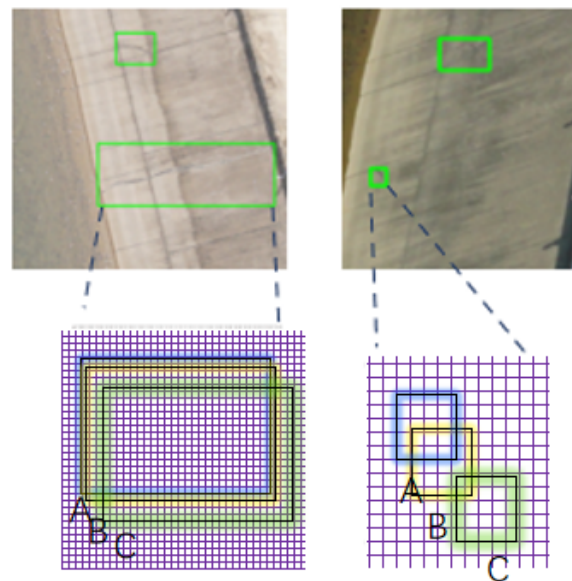


Figure 7. Targets of different sizes are inconsistently sensitive to IOU calculations.

We introduce the Wasserstein distance [38] as a new metric to replace the traditional intersection–parity ratio metric. The core idea is to model bounding boxes as two-dimensional Gaussian distributions. Even if the bounding boxes do not overlap or overlap very little in pixel space, their distributions may still have some similarity, which can be calculated by the Wasserstein distance. In these bounding boxes, foreground and background pixels are concentrated at the center and boundary of the bounding box, respectively. In order to better describe the weights of different pixels in the bounding boxes, especially those of tiny objects, they are modeled as 2D Gaussian distributions. The normalized Wasserstein distance (NWD) measures the similarity of these derived Gaussian distributions. Bounding boxes with no or very small overlap can be handled. Formally, let N_a and N_b be the two Gaussian distributions corresponding with two bounding boxes $R(cx_a, cy_a, w_a, h_a)$ and $R(cx_b, cy_b, w_b, h_b)$; then, the Wasserstein distance between N_a and N_b is

$$W_2^2(N_a, N_b) = \|[cx_a, cy_a, w_a, h_a]^T, [cx_b, cy_b, w_b, h_b]^T\|_F^2. \quad (1)$$

Thus, the loss representation for fitting the bounding box based on Gaussian distribution is

$$L_{NWD} = 1 - NWD(N_a, N_b), \quad (2)$$

where $NWD(N_a, N_b) = -\frac{\sqrt{W_2^2(N_a, N_b)}}{C}$, and C is a constant.

3.4.3. Improved Feature Fusion Module

To enhance the feature detection capability of the target detection model for small targets, we propose an improved feature fusion module called LSKA-SPFF. First, we introduce the large kernel attention module (LKA) into the feature fusion layer to address the limitations of the traditional convolution in capturing the long-range relationships by integrating the strengths of the self-attention mechanism and the large kernel convolution. It employs a decomposition strategy to decompose a large convolution kernel into multiple smaller convolution operations focusing on different spatial scales and channel dimensions while capturing local structural information, long dependencies, and inter-channel correlations. Specifically, as shown in Figure 8, a large convolution kernel can be decomposed into three parts: spatial local convolution (deep convolution), which is responsible for extracting the local structural information of the image; spatial remote convolution (deep extended convolution), where a larger sensory field captures remote spatial dependencies;

and channel convolution (1×1 convolution), which focuses on inter-channel correlations to allow the network to integrate the information in the channel dimension.

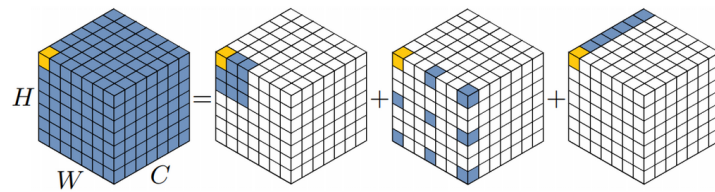


Figure 8. Decomposition of LKA convolutional structures.

Meanwhile, to solve the problem of memory increase caused by LKA, depth-separable convolution, LSKA [39], is introduced based on LKA. As shown in Figure 9, the two-dimensional weight kernel of the deep convolution is first decomposed into two cascaded one-dimensional separable convolution kernels:

$$\bar{Z}^C = \sum_{H,W} W_{(2d-1) \times 1}^C * \left(\sum_{H,W} W_{1 \times (2d-1)}^C * F^C \right). \tag{3}$$

Then, the two-dimensional weight kernel is decomposed into two cascaded one-dimensional separable convolution kernels as

$$Z^C = \sum_{H,W} W_{\lfloor \frac{k}{d} \rfloor \times 1}^C * \left(\sum_{H,W} W_{1 \times \lfloor \frac{k}{d} \rfloor}^C * \bar{Z}^C \right). \tag{4}$$

Next, we can obtain the attention map as

$$A^C = W_{1 \times 1} * Z^C, \tag{5}$$

$$\bar{F}^C = A^C \otimes F^C. \tag{6}$$

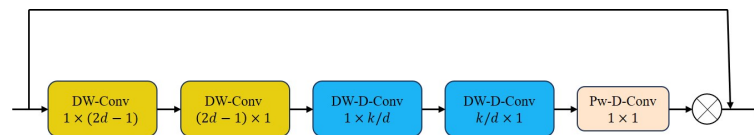


Figure 9. The convolutional structure of LSKA.

Finally, we obtain the improved SPPF structure by fusing this module with the SPPF module, as shown in Figure 10, which focuses more on the shape features of the target and excludes the interference of texture information in the background.

3.5. Network Slimming Based on Scaling Factors in BN Layers

For the trained target detection model, we apply a distillation-after-pruning strategy to adapt to the scenario of UAVs with limited computational resources. Specifically, we will first prune the improved model obtained in Section 3.4 and then use the high-performance model as the teacher model and the pruned model as the student model for model distillation. We employ a pruning method [40] that utilizes scaling factors derived from Batch Normalization (BN) layers. Given that BN layers constitute a fundamental component in many classical neural network architectures, the characteristics of BN layers can be leveraged to gauge the significance of channels. During the training process, the Batch Normalization (BN) layer first computes the mean and variance of the mini-batch data. Subsequently, the data are normalized using these mean and variance values as:

$$\hat{x} = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}}. \tag{7}$$

Then, two parameters are introduced, scaling factor γ and bias β . After normalization, the data are transformed by applying a scale factor and a bias, resulting in the following transformation:

$$y^{(k)} = \gamma^{(k)} \hat{x}^{(k)} + \beta^{(k)}. \tag{8}$$

When training the BN layer in the network, the automatically learned scale scaling factor γ is optimized according to the input data, and the trained loss function in the different channels contains information to evaluate its own importance. Initially, the original model is trained normally to establish a baseline. Subsequently, L_1 regularization is employed by adding the sum of the scaling factors of the weights as a penalty term to the loss function. This encourages the scaling factors of the Batch Normalization (BN) layers within the network to approach zero. As a result, a sparsely weighted network is obtained. After training is completed, to determine which channels can be safely removed, the distribution of the scaling factors is analyzed, and a global pruning threshold is established. This threshold is typically determined based on a specific percentile of the values of the scaling factors. Channels with scaling factors below the threshold are eliminated, and the corresponding weights are removed, along with all input and output connections involved in the convolutional computations. After the removal of a substantial number of parameters, a compact model is obtained. This model is then fine-tuned to compensate for any potential accuracy loss due to pruning. The aforementioned process is repeated iteratively to gradually compress the model size, thereby achieving a model that meets the pre-set compression ratio.

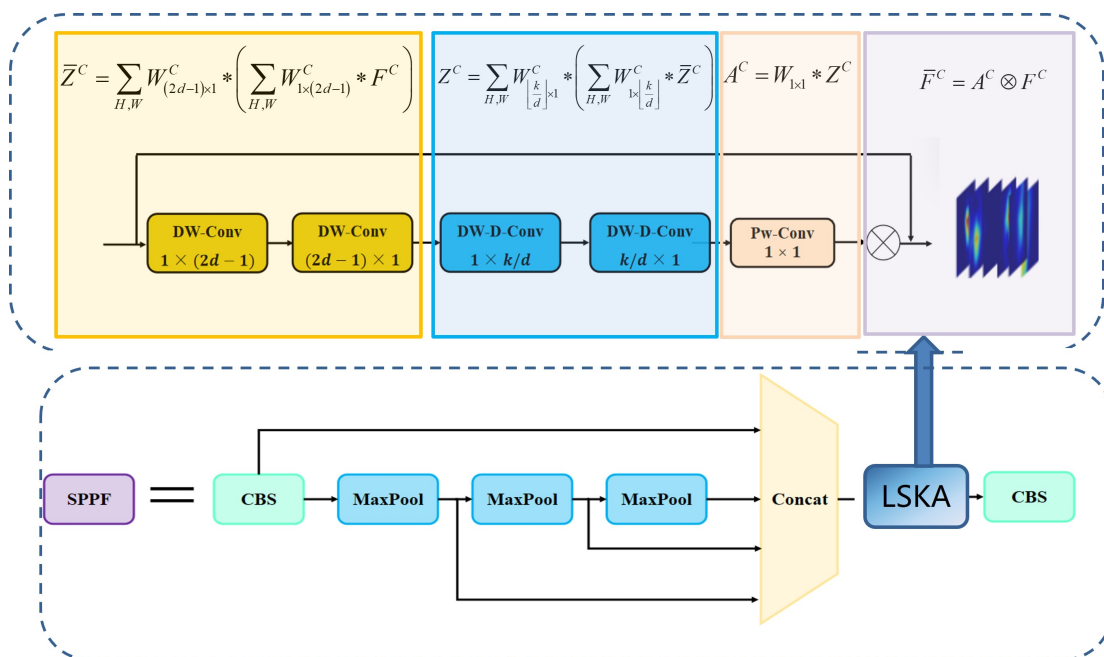


Figure 10. Feature fusion module with the incorporation of LSKA module.

3.6. Joint Feature Distillation Algorithm

In this section, we propose the joint feature distillation method, which consists of two parts, output information-based knowledge distillation and feature graph-based knowledge distillation. As shown in Figure 11, the first part extracts direct information from the output results, which includes object loss and iou loss. Since the object detection model’s output includes the target category, target confidence, and target location, this task has only one category, and the role of category probability is not considered. The student network is encouraged to learn the output information distribution of the teacher network. The other part considers extracting solutions to the imbalance between the background pixels and

foreground pixels in the image from the feature map and designs knowledge distillation that separates the foreground from the background. The attention map is used to encourage the student network to learn the attention weight distribution of the teacher network while obtaining the attention-based mask matrix loss, which is used to encourage the student network to learn the output feature distribution of the teacher network.

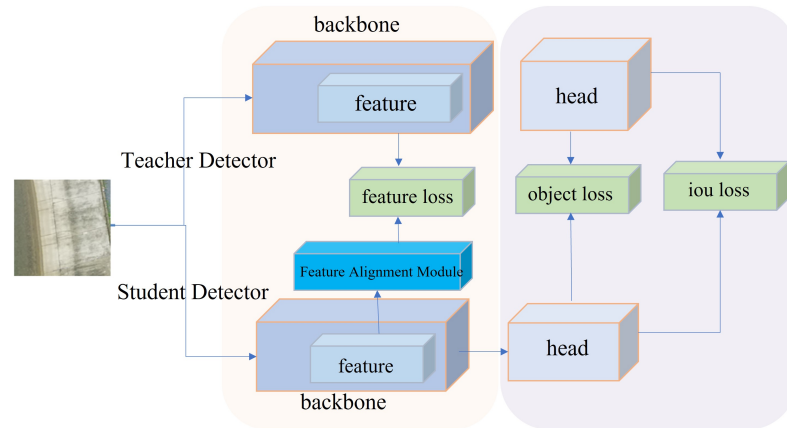


Figure 11. The overall framework of the joint feature knowledge distillation algorithm.

3.6.1. Knowledge Distillation Based on Outputs

For the first part of our method, as shown in Figure 12, there are two components. The first one is about the knowledge distillation of confidence. In the field of target detection, traditional classification distillation methods may not be suitable for dense detection tasks because they are usually designed for class-balanced scenarios. The softmax function is suitable for calculating the relationships between classes and is commonly used in the calculation of classification distillation. However, in target detection tasks, due to the extreme imbalance between foreground and background classes, directly applying the distillation loss from image classification may not bring the optimal classification performance of the student model in dense target detection, especially when the foreground class is extremely imbalanced. Therefore, the classification logits from both the teacher and student models are mapped into multiple binary classification mappings, and a binary classification distillation loss is applied to each mapping. Here, the distillation based on classification is directly transferred to the confidence levels, treating the confidence mappings as multiple binary mappings during the distillation process. Formally, the confidence loss is

$$L_{obj}^{dis}(p_{i,j}^s, p_{i,j}^t) = -\left((1 - p_{i,j}^t) \cdot \log(1 - p_{i,j}^s) + p_{i,j}^t \cdot \log(p_{i,j}^s) \right), \tag{9}$$

where $p^t = \text{sigmoid}(l_t)$ and $p^s = \text{sigmoid}(l_s)$. In addition, in order to enhance the focus on key regions, a loss weighting strategy is used to focus on extracting important samples. The importance weight w for sample x is calculated as $w = |p^t - p^s|$.

Therefore, the equation of the confidence loss equation in our work is

$$L_{obj}^{dis}(x) = \sum_{i=1}^n \sum_{j=1}^K w_{i,j} \cdot L_{BCE}(p_{i,j}^s, p_{i,j}^t). \tag{10}$$

The other one is localization distillation [41], which transforms the bounding boxes into probability distributions for distillation. However, this requires the use of a specific discrete detection head, which most models do not have and would require specialized training. To address this issue, the most basic positional relationship between two bounding boxes is directly utilized to transfer the location information from the teacher model to the student model. Specifically, localization results are obtained from both the teacher and student models, and the corresponding localization predictions from the teacher and

student models for the given input sample x at the i -th location are denoted as o_i^t and o_i^s , respectively. Then, by using the position of the anchor boxes and the localization predictions, the bounding boxes b_i^t and b_i^s for x are obtained, where A_i represents the i -th anchor box.

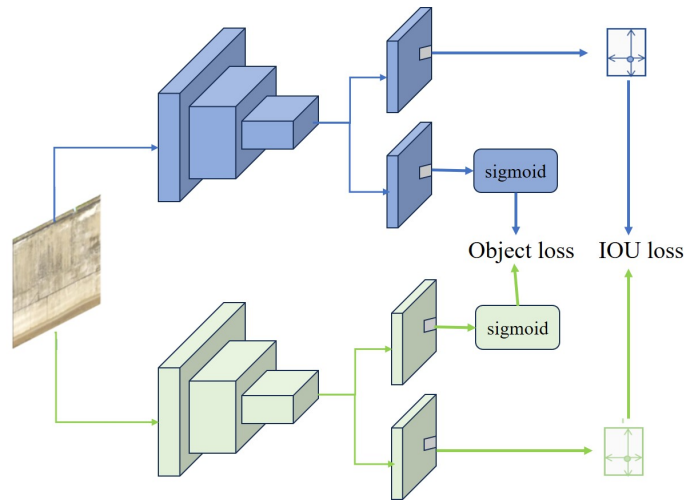


Figure 12. Distillation strategy based on output information.

The formula for calculating distillation loss based on location information is

$$L_{loc}^{dis}(x) = \sum_{i=1}^n (1 - u_i), \tag{11}$$

where $u_i = IoU(b_i^t, b_i^s)$. In summary, the overall loss function for the first part is

$$L_{output} = L_{loc}^{dis} + \alpha L_{obj}^{dis} \tag{12}$$

3.6.2. Knowledge Distillation Based on Feature Maps

For the second part of our method, as shown in Figure 13, we considered two types of features, foreground features and attention features [42]. Introducing the feature loss function L_{focal} from [42] can further help our lightweight model to focus on the features of the target.

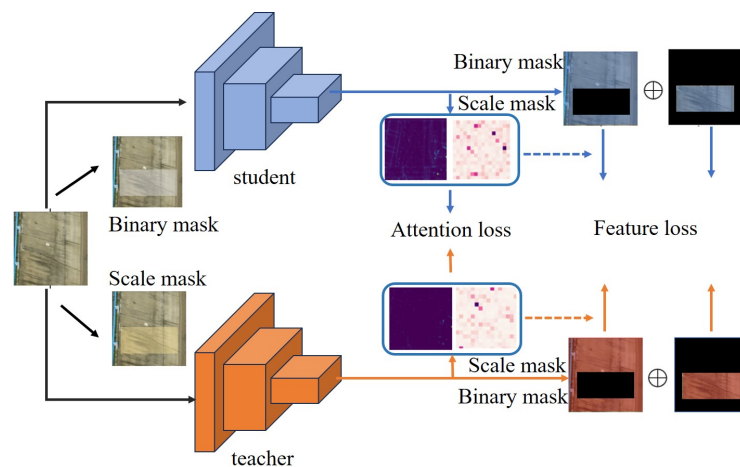


Figure 13. Distillation strategy based on feature maps.

By considering the above two types of information together, we obtain the final joint loss function:

$$L_{joint} = L_{output} + \beta L_{focal}. \tag{13}$$

4. Results

4.1. Dataset and Experimental Setup

In accordance with the methodologies outlined in Sections 3.1 and 3.2, we prepared a dataset consisting of aerial photographs of dam cracks captured by unmanned aerial vehicles (UAVs). The dataset comprises a total of 3157 images, which are categorized into three distinct sets: a training set with 2336 images, a validation set with 406 images, and a test set with 415 images. Each image in the dataset has a uniform resolution of 640×640 pixels.

In our experiments, the learning rate for the dataset was uniformly set to 0.001, with a batch size of 32. We employed the stochastic gradient descent (SGD) algorithm for iterative optimization. To facilitate faster convergence of the network, we configured the momentum weight to 0.937, which accelerates the gradient updates in the direction of the relevant axes. All algorithms were developed using the PyTorch deep learning framework, leveraging the computational capabilities of an A6000 GPU. The experiments were conducted on a server running Python 3.8, with the environment configured for PyTorch version 2.0.1, CUDA 11.7. The hyperparameter settings for the experiments are summarized in Table 1.

Table 1. The setting of hyperparameters in our experiments.

Hyperparameter	Value
Input size	640×640
Iterations	250
Learning rate	0.01
Batchsize	32

4.2. Evaluate Metric

To assess the efficacy of the methodologies proposed in this paper, we have employed a suite of standard metrics to evaluate the experimental outcomes. Firstly, the formulas for calculating recall and precision are as follows:

$$Precision = \frac{TP}{TP + FP} \quad (14)$$

$$Recall = \frac{TP}{TP + FN} \quad (15)$$

where TP , FP and FN denote true positive, false positive, and false negative, respectively. Mean Average Precision (mAP) is used to comprehensively evaluate the performance of the object detection model, which means the average value of the average precision (AP) of each category. The mean precision (AP) refers to the average value of the detection accuracy under different detection rates, and its value is equal to the area under the PR curve. Formally, AP and mAP are computed as

$$AP = \int_0^1 P(r)dr, \quad (16)$$

$$mAP = \frac{\sum_{j=1}^M AP(j)}{M}, \quad (17)$$

where M is the total number of categories.

Additionally, we use the model's parameter count (Params) and the number of operations to reflect the size of the model. One GFLOPs is equivalent to one billion floating-point operations per second.

4.3. Comparison Experiment

In this section, we will compare the performance of the proposed object detection model and the lightweight version. Meanwhile, we will also compare the performance of our proposed distillation method with the state-of-the-art method.

4.3.1. Comparative Analysis of Object Detection Models

In order to substantiate the efficacy of the methodologies proposed in this paper, a comprehensive set of baseline approaches has been selected for comparative analysis. The comparison encompasses not only the detection performance but also the efficacy of model lightweighting. The methodologies selected include various variants of the YOLOv5 series, as well as the state-of-the-art (SOTA) approaches YOLOv7 and YOLOv8. It is noteworthy that the primary focus of this comparison is on models such as the YOLO series, which are well suited for offline tasks.

According to the experimental results in Table 2, we found that on specific tasks, the YOLOv8 series is not necessarily superior to the earlier YOLOv5 series. From the perspective of the number of parameters alone, as the number of parameters increases, the model's performance tends to get better. However, this increase is not sustainable. This phenomenon also provides a basis for us to make trade-offs between precision and lightweight design. As indicated in Table 2, the improved YOLO method, referred to as yolov5_ns, which we have proposed, achieves optimal performance in terms of mean Average Precision (mAP), surpassing the YOLOv5 series as well as YOLOv8n by 4.9%. Moreover, the parameter count is approximately two-thirds that of YOLOv8n. When compared with the YOLOv5 series, our method has a parameter count close to YOLOv5n but outperforms it by 3%.

Table 2. Performance comparison of object detection models.

Model	mAP_{50}	$mAP_{50:100}$	Precision	Recall	Params	Gflops
Yolov5l	0.81	0.441	0.798	0.797	46,108,278	107.6
Yolov5s	0.81	0.441	0.798	0.797	46,108,278	107.6
Yolov5n	0.785	0.409	0.784	0.776	1,760,518	4.1
Yolov7	0.795	0.42	0.771	0.79	36,481,772	103.2
Yolov7_tiny	0.658	0.298	0.64	0.705	6,007,596	13
Yolov8s	0.766	0.422	0.794	0.782	11,125,971	28.4
Yolov8n	0.765	0.417	0.78	0.783	3,005,843	8.1
Yolov5_ns(ours)	0.815	0.435	0.809	0.782	2,033,414	4.3
drone-Yolov5(ours)	0.804	0.410	0.800	0.770	725,248	2.1

Furthermore, focusing on the lightweight version of our proposed model, drone-YOLOv5, the parameter count is reduced to 35.7% of the original yolov5_ns, with only a 1.1% decrease in performance. When compared with other models, drone-YOLOv5 demonstrates superior performance with a notably lower parameter count. For instance, the parameter count is less than a quarter of YOLOv8n, yet the performance is 3.9% higher.

In summary, the lightweighting approach proposed in this paper exhibits its superiority by maintaining high performance while significantly reducing the parameter count to one-third or even less.

4.3.2. Comparative Analysis of Knowledge Distillation

In the experimental setup of knowledge distillation, the well-optimized model yolov5_ns was selected as the teacher model, while the model post-pruning was designated as the student model. Comparative analysis of the results presented in Table 3 reveals that our proposed distillation method outperforms the state-of-the-art (SOTA) knowledge distillation methods across multiple metrics.

We can observe that traditional distillation methods have overly focused on knowledge distillation of feature regions. In our architecture, the student model is obtained by pruning the teacher model; thus, the effect of feature distillation is limited, and if not handled properly, it could even yield opposite effects. In response to this, we propose a joint distillation method that combines feature distillation with output information distillation, enriching the diversity of information sources and effectively enhancing the efficiency of knowledge distillation. In the end, we developed the Drone-YOLO model with a parameter

count of less than 0.73 M, which achieved a 1.8% improvement over the original student model, and its performance is close to that of a model with a parameter count as high as 46 M.

Table 3. Performance comparison of knowledge distillation methods.

Method	mAP_{50}	$mAP_{50:100}$	Precision	Recall
teacher	0.815	0.435	0.809	0.782
student	0.786	0.394	0.763	0.801
CWD [35]	0.796	0.405	0.782	0.775
FGD [42]	0.791	0.397	0.775	0.776
MGD [43]	0.794	0.4	0.78	0.775
AMD [44]	0.785	0.394	0.76	0.787
ours	0.804	0.41	0.8	0.77

4.4. Ablation Experiment

In the ablation study, we aim to assess the contribution of various modules within an object detection model by systematically removing or modifying them. This approach allows us to verify the role and significance of each module in the model.

According to the data analysis from Table 4, after improving the loss function, the model's detection capability for small objects has been enhanced, with the mAP increasing from 78.5% to 80.3%, while the model complexity remained unchanged. Following the improvement of the feature joint module, the model's detection capability for targets with complex shape features has been strengthened, with the mAP rising from 78.5% to 79.6%, and the model complexity increased slightly. These two improvement strategies achieved increases of 1.8% and 1.1%, respectively, and finally, applying both improvement strategies simultaneously, the mAP achieved a 3% increase. The number of parameters and the amount of floating-point operations increased from the original 176,051 to 2,033,414 and from 4.1 Gflops to 4.3 Gflops, respectively. This shows that while the model's performance was enhanced, the computational cost did not increase significantly.

Table 4. Ablation experiment of improved YOLOv5.

Method	mAP_{50}	$mAP_{50:100}$	Precision	Recall
Yolov5n	0.785	0.409	0.784	0.776
Yolov5n+nwd	0.803	0.41	0.803	0.768
Yolov5n+lska_spff	0.796	0.418	0.787	0.775
Yolov5_ns(ours)	0.815	0.435	0.809	0.782

Selecting the pruned model as the student model, the overall parameter count is 35.7% of the improved model, and the floating-point operations are 50% of the original. The teacher model's mAP_{50} value is 2.9% higher than that of the student model. According to the data analysis from Table 5, when output information-based distillation acts alone, the student model's mAP_{50} increases by 0.8%. When the feature distillation acts alone, the student model's mAP_{50} also increases by 0.8%. When multiple strategies are integrated into knowledge distillation, the student model's mAP_{50} increases by 1.8%.

Table 5. Ablation experiment of our knowledge distillation method.

Method	mAP_{50}	$mAP_{50:100}$	Precision	Recall
teacher	0.815	0.435	0.809	0.782
student	0.786	0.394	0.763	0.801
with L_{output}	0.794	0.405	0.774	0.782
with L_{focal}	0.794	0.401	0.784	0.775
with $L_{output} + L_{focal}$	0.804	0.41	0.8	0.77

4.5. Visual Analysis

As shown in Figure 14, we present the final detection results. By cutting a 6000×4000 aerial image captured by a drone into several 640×640 sub-images, we have achieved dam crack target detection under remote sensing conditions. It can be observed that even the tiny crack area in the upper right corner can be detected by our method. Additionally, the post-processing algorithm we designed is capable of eliminating the overlap of target area boxes that occurs during the merging of sub-images.

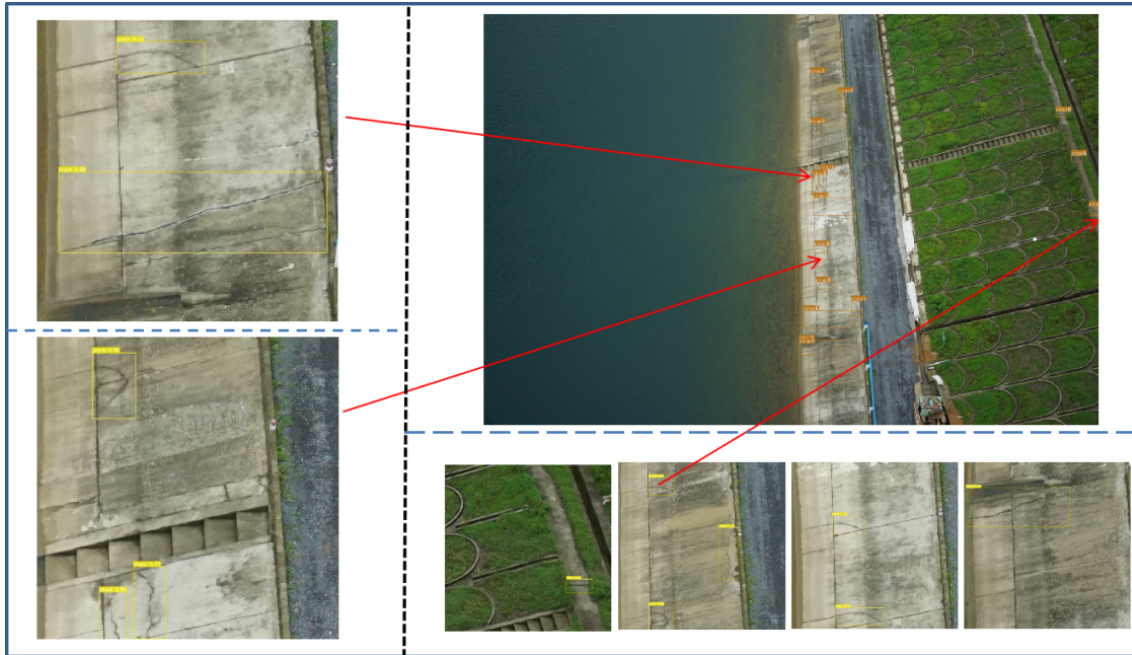


Figure 14. Detection results of original images of dam cracks.

As depicted in Figure 15, we present a comparison between the YOLOv5n model and our optimized YOLOv5_ns model. Figure 15a represents the original image, while Figure 15b illustrates the detection results of the pre-improved YOLOv5n model. Figure 15c displays the detection outcomes of the YOLOv5_ns model. It is evident that after the enhancement, smaller targets that were previously undetected are now identified, indicating an improved model adaptability to targets of varying scales.

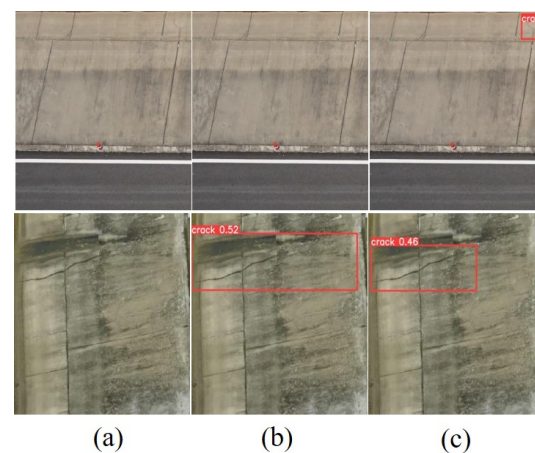


Figure 15. Comparison of detection results for cracks in dams. (a) Original image; (b) results of YOLOv5n; (c) results of YOLOv5_ns.

To investigate the effectiveness of the proposed method, we use the GradCAM [45] technique to interpret the model, which involves generating a heatmap. The highlighted areas in the heatmap represent the parts that the network focuses on; the more red the color, the more attention the network pays to that area, while the darker the color, the less attention it receives. As shown in Figures 16 and 17, our proposed joint distillation method enhances the model's ability to recognize small target areas, giving more attention to targets that are relatively shallow and easily submerged by noise.

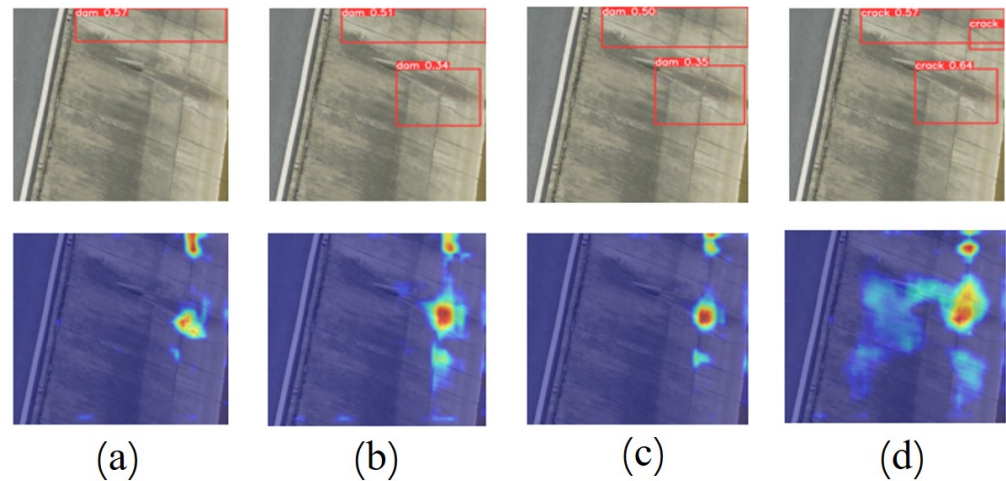


Figure 16. Comparison of results and heatmap from different distillation methods. (a) represents the model after pruning; (b) represents the model after local distillation based on feature maps; (c) represents the model with knowledge distillation based on output information; and (d) represents the model with multi-strategy joint distillation algorithm.

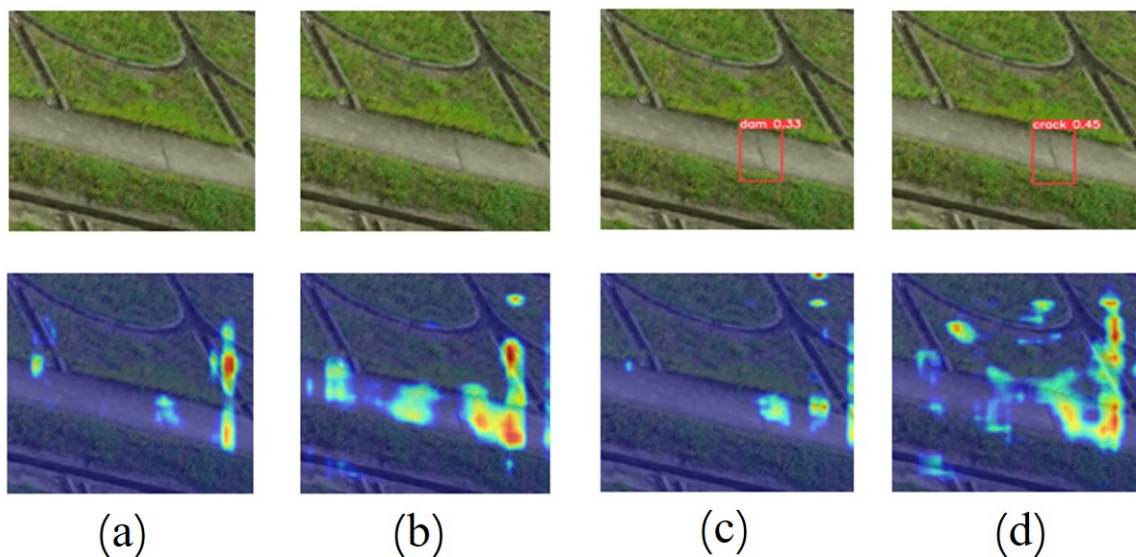


Figure 17. Comparison of results and heatmap from different distillation methods on another image. (a) represents the model after pruning; (b) represents the model after local distillation based on feature maps; (c) represents the model with knowledge distillation based on output information; and (d) represents the model with multi-strategy joint distillation algorithm.

5. Discussion

In this work, we propose a lightweight object detection algorithm that can directly process high-resolution images. Our scheme mainly addresses two key issues, high resolution

and light weight, which provide a strong algorithmic guarantee for the future implementation of field inspections using drones. The first major challenge faced by field inspections is the limitation of computational resources; hence, the lightweight performance of the model is crucial. Additionally, the ability to directly process high-resolution input images is also related to the time consumption of the entire inspection process. The proposed method can also be applied to similar scenarios, such as a bridge defect detection and power line inspection. However, the current research is still in its infancy, and future deployment practices will need to be coordinated with issues such as drone path planning and drone hardware design.

6. Conclusions

In this work, we propose an automatic dam inspection method for unmanned aerial vehicles (UAVs) suitable for field operations. We have implemented the entire process from UAV image collection to training and introduced optimization ideas for object detection models. On this basis, we have proposed a lightweight method more suitable for field operations. Compared to other dam crack inspection methods that only work at low resolutions, our algorithm can be directly applied to high-resolution remote sensing image inspection. More importantly, our method can effectively maintain the performance of the original model while significantly reducing the number of model parameters. On our own collected actual dataset, our final model has only 35.7% of the parameter volume before pruning, and the performance has only decreased by 1.1%. Compared with other SOTA models, our overall solution has achieved a minimal parameter volume while maintaining comparable performance. In the future, we will further consider the direct deployment of intelligent detection algorithms on hardware.

Author Contributions: Conceptualization, H.D. and N.W.; methodology, H.D.; software, D.F.; validation, B.L., G.L. and D.F.; formal analysis, N.W.; investigation, F.W.; resources, F.W.; data curation, F.W.; writing—original draft preparation, H.D.; writing—review and editing, H.D.; visualization, D.F.; supervision, B.L.; project administration, B.L. Funding, N.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the research funding of Hainan Normal University(HSZK-KYQD-202431).

Data Availability Statement: Restrictions apply to the datasets. The dataset provided in this article is not easily accessible as it is part of an ongoing project. The request to access the dataset should be directed to Fupeng Wei.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhao, Z.; Chen, B.; Wu, Z.; Zhang, S. Multi-sensing investigation of crack problems for concrete dams based on detection and monitoring data: A case study. *Measurement* **2021**, *175*, 109137. [[CrossRef](#)]
2. Dai, F.; Dong, W.; Huang, Z.; Zhang, X.; Wang, A.; Cao, S. Study on Shear Strength of Undisturbed Expansive Soil of Middle Route of South-to-North Water Diversion Project. *Adv. Eng. Sci* **2018**, *50*, 123–131.
3. Hu, Q.; Wang, P.; Li, S.; Liu, W.; Li, Y.; Lu, W.; Kou, Y.; Wei, F.; He, P.; Yu, A. Research on Intelligent Crack Detection in a Deep-Cut Canal Slope in the Chinese South–North Water Transfer Project. *Remote Sens.* **2022**, *14*, 5384. [[CrossRef](#)]
4. Zhang, Z. Drone-YOLO: An efficient neural network method for target detection in drone images. *Drones* **2023**, *7*, 526. [[CrossRef](#)]
5. Nooralishahi, P.; Ibarra-Castanedo, C.; Deane, S.; López, F.; Pant, S.; Genest, M.; Avdelidis, N.P.; Maldague, X.P. Drone-based non-destructive inspection of industrial sites: A review and case studies. *Drones* **2021**, *5*, 106. [[CrossRef](#)]
6. Memari, M.; Shakya, P.; Shekaramiz, M.; Seibi, A.C.; Masoum, M.A. Review on the advancements in wind turbine blade inspection: Integrating drone and deep learning technologies for enhanced defect detection. *IEEE Access* **2024**, *12*, 33236–33282. [[CrossRef](#)]
7. Fernández-Hernandez, J.; González-Aguilera, D.; Rodríguez-González, P.; Mancera-Taboada, J. Image-based modelling from unmanned aerial vehicle (UAV) photogrammetry: An effective, low-cost tool for archaeological applications. *Archaeometry* **2015**, *57*, 128–145. [[CrossRef](#)]
8. Elijah, O.; Rahman, T.A.; Orikumhi, I.; Leow, C.Y.; Hindia, M.N. An overview of Internet of Things (IoT) and data analytics in agriculture: Benefits and challenges. *IEEE Internet Things J.* **2018**, *5*, 3758–3773. [[CrossRef](#)]

9. Sudevan, V.; Shukla, A.; Karki, H. Current and Future Research Focus on Inspection of Vertical Structures in Oil and Gas Industry. In Proceedings of the 18th International Conference on Control, Automation and Systems (ICCAS), PyeongChang, Korea, 17–20 October 2018; pp. 144–149.
10. Li, L.; Zhao, H.; Liu, R.; Nayyar, A.; Ali, R.; Li, Y.; Zhang, H. CiC-NET: A real-time semantic segmentation network for dam surface crack detection. *Multimed. Tools Appl.* **2024**, 1–23. [[CrossRef](#)]
11. Zhou, J.; Zhao, G.; Li, Y. Vison Transformer-Based Automatic Crack Detection on Dam Surface. *Water* **2024**, *16*, 1348. [[CrossRef](#)]
12. Zou, Q.; Zhang, Z.; Li, Q.; Qi, X.; Wang, Q.; Wang, S. Deepcrack: Learning hierarchical convolutional features for crack detection. *IEEE Trans. Image Process.* **2018**, *28*, 1498–1512. [[CrossRef](#)] [[PubMed](#)]
13. Mohan, A.; Poobal, S. Crack detection using image processing: A critical review and analysis. *Alex. Eng. J.* **2018**, *57*, 787–798. [[CrossRef](#)]
14. Oliveira, H.; Correia, P.L. Automatic road crack detection and characterization. *IEEE Trans. Intell. Transp. Syst.* **2012**, *14*, 155–168. [[CrossRef](#)]
15. Hsieh, Y.A.; Tsai, Y.J. Machine learning for crack detection: Review and model performance comparison. *J. Comput. Civ. Eng.* **2020**, *34*, 04020038. [[CrossRef](#)]
16. Yokoyama, S.; Matsumoto, T. Development of an automatic detector of cracks in concrete using machine learning. *Procedia Eng.* **2017**, *171*, 1250–1255. [[CrossRef](#)]
17. Nhat-Duc, H.; Nguyen, Q.L.; Tran, V.D. Automatic recognition of asphalt pavement cracks using metaheuristic optimized edge detection algorithms and convolution neural network. *Autom. Constr.* **2018**, *94*, 203–213. [[CrossRef](#)]
18. Kim, H.; Ahn, E.; Shin, M.; Sim, S.H. Crack and noncrack classification from concrete surface images using machine learning. *Struct. Health Monit.* **2019**, *18*, 725–738. [[CrossRef](#)]
19. Li, R.; Yuan, Y.; Zhang, W.; Yuan, Y. Unified vision-based methodology for simultaneous concrete defect detection and geolocalization. *Comput. Aided Civ. Infrastruct. Eng.* **2018**, *33*, 527–544. [[CrossRef](#)]
20. Huyan, J.; Li, W.; Tighe, S.; Zhai, J.; Xu, Z.; Chen, Y. Detection of sealed and unsealed cracks with complex backgrounds using deep convolutional neural network. *Autom. Constr.* **2019**, *107*, 102946. [[CrossRef](#)]
21. Maeda, H.; Sekimoto, Y.; Seto, T.; Kashiyama, T.; Omata, H. Road damage detection and classification using deep neural networks with smartphone images. *Comput. Aided Civ. Infrastruct. Eng.* **2018**, *33*, 1127–1141. [[CrossRef](#)]
22. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
23. Howard, A.G. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
24. Redmon, J. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
25. Mandal, V.; Uong, L.; Adu-Gyamfi, Y. Automated Road Crack Detection Using Deep Convolutional Neural Networks. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; pp. 5212–5215.
26. Fan, Z.; Li, C.; Chen, Y.; Wei, J.; Loprencipe, G.; Chen, X.; Di Mascio, P. Automatic crack detection on road pavements using encoder–decoder architecture. *Materials* **2020**, *13*, 2960. [[CrossRef](#)] [[PubMed](#)]
27. Chen, T.; Cai, Z.; Zhao, X.; Chen, C.; Liang, X.; Zou, T.; Wang, P. Pavement crack detection and recognition using the architecture of segNet. *J. Ind. Inf. Integr.* **2020**, *18*, 100144. [[CrossRef](#)]
28. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder–decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [[CrossRef](#)]
29. Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object detection in 20 years: A survey. *Proc. IEEE* **2023**, *111*, 257–276. [[CrossRef](#)]
30. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
31. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
32. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single Shot Multibox Detector. In *Computer Vision—ECCV 2016, Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016*; Proceedings, Part I; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
33. Wang, J.; Chen, Y.; Zheng, Z.; Li, X.; Cheng, M.M.; Hou, Q. CrossKD: Cross-Head Knowledge Distillation for Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 16520–16530.
34. Zhao, B.; Cui, Q.; Song, R.; Qiu, Y.; Liang, J. Decoupled Knowledge Distillation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 11953–11962.
35. Shu, C.; Liu, Y.; Gao, J.; Yan, Z.; Shen, C. Channel-Wise Knowledge Distillation for Dense Prediction. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 5311–5320.
36. Liu, Z.; Wang, Y.; Chu, X.; Dong, N.; Qi, S.; Ling, H. A Simple and Generic Framework for Feature Distillation via Channel-Wise Transformation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 1–6 October 2023; pp. 1129–1138.
37. Mansourian, A.M.; Jalali, A.; Ahmadi, R.; Kasaei, S. Attention-guided Feature Distillation for Semantic Segmentation. *arXiv* **2024**, arXiv:2403.05451.
38. Wang, J.; Xu, C.; Yang, W.; Yu, L. A normalized Gaussian Wasserstein distance for tiny object detection. *arXiv* **2021**, arXiv:2110.13389.

39. Lau, K.W.; Po, L.M.; Rehman, Y.A.U. Large separable kernel attention: Rethinking the large kernel attention design in cnn. *Expert Syst. Appl.* **2024**, *236*, 121352. [[CrossRef](#)]
40. Liu, Z.; Li, J.; Shen, Z.; Huang, G.; Yan, S.; Zhang, C. Learning Efficient Convolutional Networks Through Network Slimming. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2736–2744.
41. Zheng, Z.; Ye, R.; Wang, P.; Ren, D.; Zuo, W.; Hou, Q.; Cheng, M.M. Localization Distillation for Dense Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 9407–9416.
42. Yang, Z.; Li, Z.; Jiang, X.; Gong, Y.; Yuan, Z.; Zhao, D.; Yuan, C. Focal and Global Knowledge Distillation for Detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 4643–4652.
43. Yang, Z.; Li, Z.; Shao, M.; Shi, D.; Yuan, Z.; Yuan, C. Masked generative distillation. In *Computer Vision—ECCV 2022, Proceedings of the 17th European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 53–69.
44. Yang, G.; Tang, Y.; Li, J.; Xu, J.; Wan, X. Amd: Adaptive Masked Distillation for Object Detection. In Proceedings of the 2023 International Joint Conference on Neural Networks (IJCNN), Gold Coast, Australia, 18–23 June 2023; pp. 1–8.
45. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual Explanations from Deep Networks via Gradient-Based Localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.

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.