remote sensing

MDPI

Article

Automatic Detection of War-Destroyed Buildings from High-Resolution Remote Sensing Images

Yu Wang [1,2], Yue Li [1,2] and Shufeng Zhang [1,2,*]

[1] Science and Technology on Integrated Logistic Support Laboratory, National University of Defense Technology, Changsha 410073, China
[2] College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China
* Correspondence: sfzhang@nudt.edu.cn

Abstract: Modern high-intensity armed conflicts often lead to extensive damage to urban infrastructure. The use of high-resolution remote sensing images can clearly detect damage to individual buildings which is of great significance for monitoring war crimes and damage assessments that destroy civilian infrastructure indiscriminately. In this paper, we propose SOCA-YOLO (Sampling Optimization and Coordinate Attention–YOLO), an automatic detection method for destroyed buildings in high-resolution remote sensing images based on deep learning techniques. First, based on YOLOv8, Haar wavelet transform and convolutional blocks are used to downsample shallow feature maps to make full use of spatial details in high-resolution remote sensing images. Second, the coordinate attention mechanism is integrated with C2f so that the network can use the spatial information to enhance the feature representation earlier. Finally, in the feature fusion stage, a lightweight dynamic upsampling strategy is used to improve the difference in the spatial boundaries of feature maps. In addition, this paper obtained high-resolution remote sensing images of urban battlefields through Google Earth, constructed a dataset for the detection of objects on buildings, and conducted training and verification. The experimental results show that the proposed method can effectively improve the detection accuracy of destroyed buildings, and the method is used to map destroyed buildings in cities such as Mariupol and Volnovaja where violent armed conflicts have occurred. The results show that deep learning-based object detection technology has the advantage of fast and accurate detection of destroyed buildings caused by armed conflict, which can provide preliminary reference information for monitoring war crimes and assessing war losses.

Keywords: armed conflict; destroyed buildings; high resolution remote sensing images; object detection; convolutional neural network

Academic Editor: Jon Atli Benediktsson

Received: 28 November 2024
Revised: 21 January 2025
Accepted: 29 January 2025
Published: 31 January 2025

Citation: Wang, Y.; Li, Y.; Zhang, S. Automatic Detection of War-Destroyed Buildings from High-Resolution Remote Sensing Images. *Remote Sens.* **2025**, *17*, 509. https://doi.org/10.3390/rs17030509

1. Introduction

In contemporary armed conflicts, the extensive utilization of thermal weapons not only poses a grave threat to live targets but also presents a significant peril to the natural landscape and human environment [1]. Particularly, the devastation of urban infrastructure incurs substantial costs in terms of both material resources and humanitarian consequences such as triggering a "refugee wave" [2]. Consequently, the indiscriminate destruction of civilian infrastructure including residential homes, commercial buildings, and cultural landscapes is deemed both inhumane and constitutes a war crime.

Since the beginning of the 21st century, the international community has been in turmoil due to various conflicts such as the Iraq War, Syrian War, and civil war in Myanmar, which have inflicted significant damage on mankind's living environment and profoundly impacted global peace and development [3–5]. The ongoing Russian–Ukrainian conflict and the new round of Israeli–Palestinian conflict have also resulted in extensive destruction of non-military facilities like urban infrastructure and civilian buildings. Concerningly, the blockade of war zones and deteriorating security conditions pose challenges for United Nations peacekeeping efforts [6]. Restricted access for relevant personnel hampers field mapping activities while blocked information impedes the international community's supervision and mediation guidance within these areas. Therefore, there is an urgent need to conduct rapid and accurate mapping of destroyed buildings in war zones to facilitate timely intervention and protection of humanitarian rights [7].

The development of earth observation technology provides people with an observation perspective that is not restricted by the region so that the destroyed buildings can be observed more objectively and comprehensively in armed conflicts. However, the resolution of early remote sensing images is not high, and the application of destroyed building detection mainly focuses on the disaster assessment after the occurrence of natural disasters such as earthquakes and mudslides [8]. Currently, Interferometric Synthetic Aperture Radar (InSAR) technology and regional spectral change measurement have been used to perform rough regional damage perception [9]. Janalipour et al. [10] proposed an automatic building damage detection framework based on the LiDAR data after a disaster, which combines texture features with average digital projectors and can greatly improve the detection accuracy under the conditions of effective texture feature extraction. Huang et al. [11] combined synthetic aperture radar images with different simultaneous phases for coherence calculation, combined with open-source building vectors for classification extraction of destroyed buildings, and obtained regional detection results consistent with events. Based on optical remote sensing images, Ghandourj et al. [12] proposed a method to estimate building damage by using shadow features and gray co-occurrence matrix features and conducted building damage assessments in areas affected by the Syrian war near Damascus. In addition, some scholars have studied the collaborative detection of collapsed buildings by pre-disaster and post-disaster data from the perspective of multi-source data [13,14]. However, due to the diversity of data and the limitation of low resolution, there was no unified method for the detection of destroyed buildings during this period. Most of the methods were highly complex, and the classification features depended on artificial design, so the damage degree of buildings could only be roughly estimated.

Nowadays, countries all over the world have developed and launched high-resolution military, civilian, and commercial remote sensing satellites, such as Worldview, SPOT, and GF-2. The resolution of remote sensing images has been improved to the submeter level, enabling high-resolution and fine-grained imaging of individual buildings. Data resources are no longer the bottleneck restricting the acquisition of information in war zones. Therefore, how to quickly and accurately detect destroyed building individuals from wide-area remote sensing images has become the focus of attention [15].

Fortunately, advances in computer vision and artificial intelligence technology have made it possible to quickly detect destroyed buildings from a massive and large range of high-resolution remote sensing images. Many high-performance methods have been used or are being used in this field for research and application, the most typical cases are convolutional neural networks (CNNs) (e.g., Faster-RCNN [16], SSD [17], YOLOs [18–26]) and the detection with Transformers (e.g., ViT-FRCNN [27], RT-DET [28]). Ji et al. [29] took the lead in using a VGG [30] network to detect collapsed buildings after earthquakes in remote sensing images, and after fine-tuning, the detection accuracy of the network was

effectively improved. Aiming at the classification of collapsed buildings, Wu et al. [31] used the improved U-Net [32] network to segment collapsed buildings at the pixel level, and the classification accuracy reached 0.792. Shi et al. [33] propose an improved YOLOv4 [20] algorithm for detecting collapsed buildings in aerial images after earthquakes. However, the actual post-disaster scene is complex and diverse, the collapsed buildings and the background are easily confused, there are still some difficulties in extracting robust features, and the detection accuracy is still not high. To solve these problems, Bai et al. [34] proposed a pyramid-pool modular semi-twin network for detecting destroyed buildings and improved the detection accuracy by adding residual blocks with expansion convolution and extrusion excitation blocks into the network. Ma et al. [35] took ShuffleNet v2 [36] as the backbone network of YOLOv3 [19] and introduced a generalized intersection over Union (GIoU) [37] loss to improve the detection accuracy of the model under a complex background. Overall, the object detection method based on deep learning has been proven to be feasible in detecting destroyed buildings in satellite images. It is worth noting that the YOLO series algorithms have been widely adopted in recent years due to their advantages of both accuracy and efficiency. In addition, attention mechanisms [38–40], feature pyramids [41–43], sampling methods [44,45], and other methods are often used to improve the detection performance of networks in specific scenarios.

However, most of the current research focuses on the detection of destroyed buildings caused by natural disasters such as earthquakes and mudslides, and there are no studies on the detection of destroyed buildings in the context of armed conflict. Figure 1 shows the difference in collapse between buildings destroyed by earthquake and war. Earthquake usually leads to overall structural damage to buildings. In particular, a seismic wave will cause the building to lose the support of the facade and collapse as a whole, as shown in Figure 1a. In armed conflict, the damage caused by air strikes, shelling, and rockets to buildings is often localized and point-like. Large buildings often have partially or completely collapsed roofs and relatively intact facades, as shown in Figure 1b, while small buildings show irregular ruins, as shown in Figure 1c.

(**a**) (**b**) (**c**)

Figure 1. Buildings destroyed by earthquakes or war. The red boxes represent the buildings destroyed by the earthquake, and the green boxes represent the war-destroyed buildings. (**a**) Buildings destroyed by the earthquakes (source: [35]); (**b**) large buildings destroyed by war (source: Google Earth); (**c**) small buildings destroyed by war (source: Google Earth).

Since the detection of destroyed buildings in high-resolution remote sensing images is highly dependent on the robustness of spatial detail feature extraction, we have made targeted improvements to CNNs and proposed an effective detection method. The main contributions of this work are as follows: (1) obtaining high-resolution Google images of the Russian–Ukrainian conflict area, and making the first dataset for the detection of destroyed buildings by visual interpretation and expert annotation; (2) in view of the tasks

dependence on spatial details, the Haar wavelet downsampling module (HWD), lightweight dynamic upsampling module (LDU), and coordinate attention mechanism (CA) were used to improve the convolutional neural network, effectively improving the detection accuracy of destroyed buildings; and (3) a comprehensive assessment and mapping of building damage in typical cities during the Russian–Ukrainian conflict is presented, and the advantages and limitations of our method for monitoring destroyed buildings in high-resolution optical remote sensing images are discussed.

2. Dataset

2.1. Remote Sensing Data Acquisition

In order to verify the effectiveness of the proposed method, we chose Mariupol, a city severely damaged by the Russian–Ukrainian conflict, and used its Google Earth images as the source of the production dataset. From February 2022 to May 2022, both sides fought with a large number of air missiles and ground artillery in the course of the armed conflict. In addition to military installations, residential and commercial buildings in the city were severely damaged. According to a report by UN High Commissioner for Human Rights Michelle Bachelet [46], the 82-day armed conflict resulted in the destruction of 90% of apartment buildings and 60% of private homes in Mariupol, with many neighborhoods razed to the ground.

Therefore, in this study, we obtained images with a resolution of about 0.3 m from Google Earth provided by Maxar, which covers the main urban area of Mariupol and contains a large number of destroyed buildings, as shown in Figure 2.

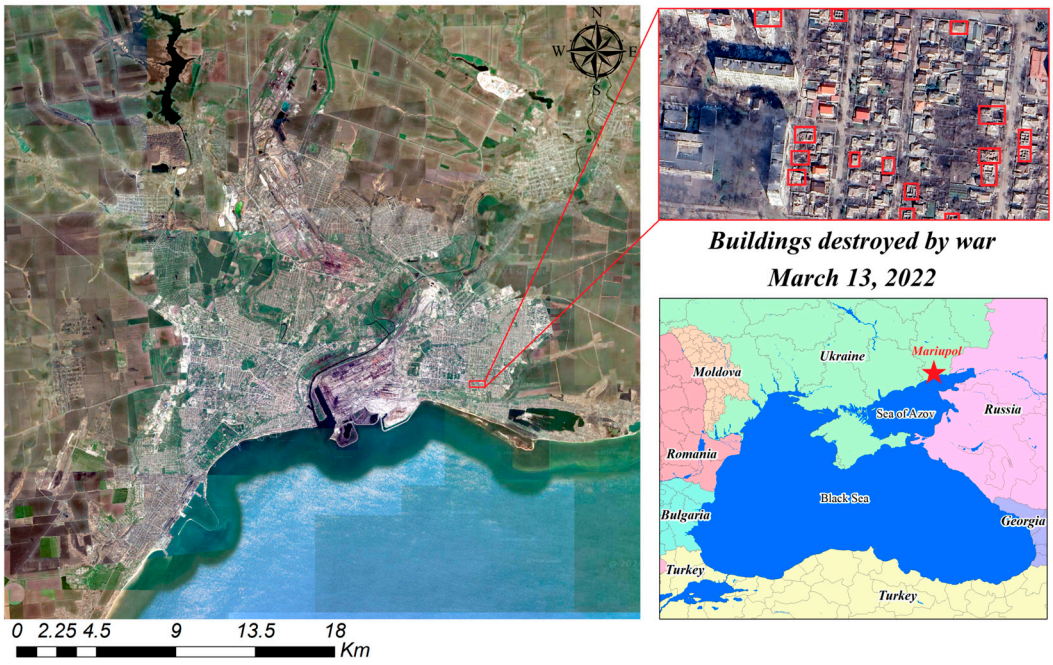


Figure 2. Collection area of the dataset, the red boxes in the upper right sub-image indicate the labels of the destroyed building. (Source: Google Earth.)

2.2. Dataset Production

Since the size of the original image is too large to be directly input into the CNN, the image needs to be pre-sliced. Meanwhile, in order to ensure the integrity of each individual building in the sample image, the image was cut into 640 × 640-pixel image slices according to the overlap degree of 25%, and the destroyed building was marked by

Labelimg Tools [47]. Figure 3 shows the pre-slicing process in the production of the dataset and the labeling information of the destroyed buildings.

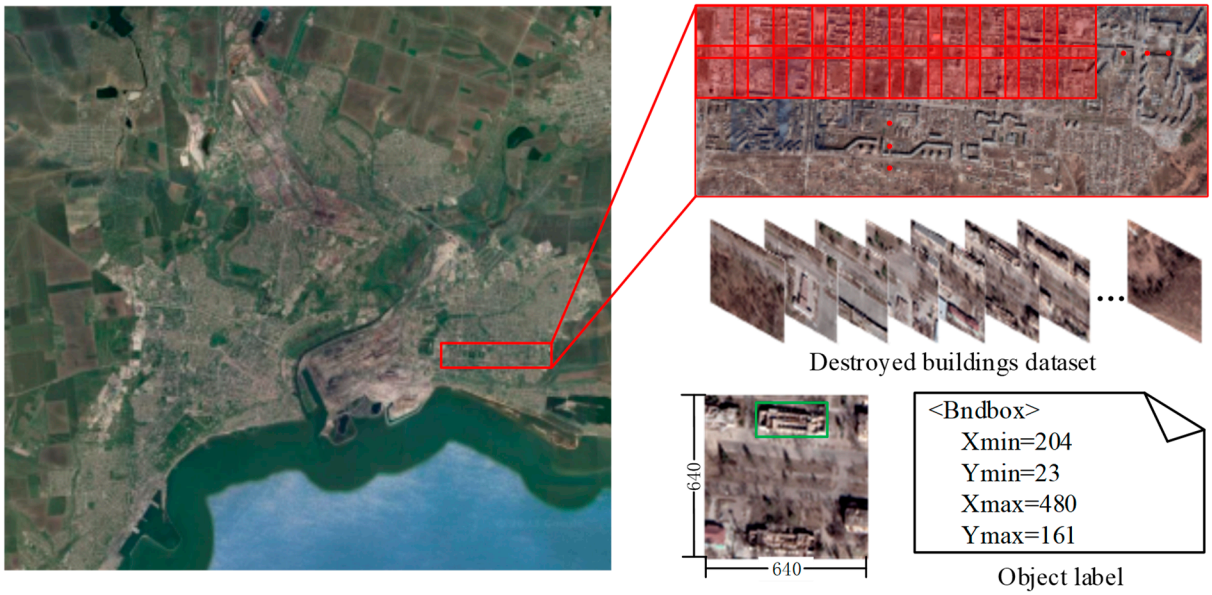


Figure 3. The production process of the destroyed buildings dataset; The red boxes in the upper right sub-image indicate the sliding windows, the green box in the lower right sub-image indicates the label of the destroyed building. (Source: Google Earth.)

In general, we cut the collected high-resolution remote sensing images of Mariupol into 14,023 slice samples, screened out 4251 sample images containing destroyed buildings, and marked 11,604 destroyed building objects. In addition, we collected the same resolution remote sensing images of the Vornovakha region as test data to verify that the model can stably detect destroyed buildings in different regions.

3. Methodology

In order to effectively take advantage of the rich detail information of high-resolution remote sensing images, we improved the sampling method and feature expression of the CNN, including the extraction of shallow spatial features and the key screening and utilization of deep semantic features of the CNN. The network structure of SOCA-YOLO is shown in Figure 4.

In the process of feature extraction, geometric features such as the shape and texture of the object are crucial for detecting destroyed buildings in high-resolution remote sensing images. In order to avoid excessive loss of shallow spatial details, we introduced the Haar wavelet 2D decomposition to replace the early stage of the backbone network for downsampling [44]. In addition, we combined the coordinate attention [40] with the C2f module in the backbone network to enhance feature representation with spatial detail information earlier, promoting cross-scale feature fusion.

In the stage of feature fusion, it is usually necessary to upsample the feature map and connect it with the shallow feature map to enrich the multi-scale object information and improve the detection performance. In this section, we introduce a lightweight dynamic upsampling module [45] into the network. By combining point sampling and offset, a feature map with more spatial position perception can be obtained and the details of semantic features in the reconstructed feature map can be better retained.

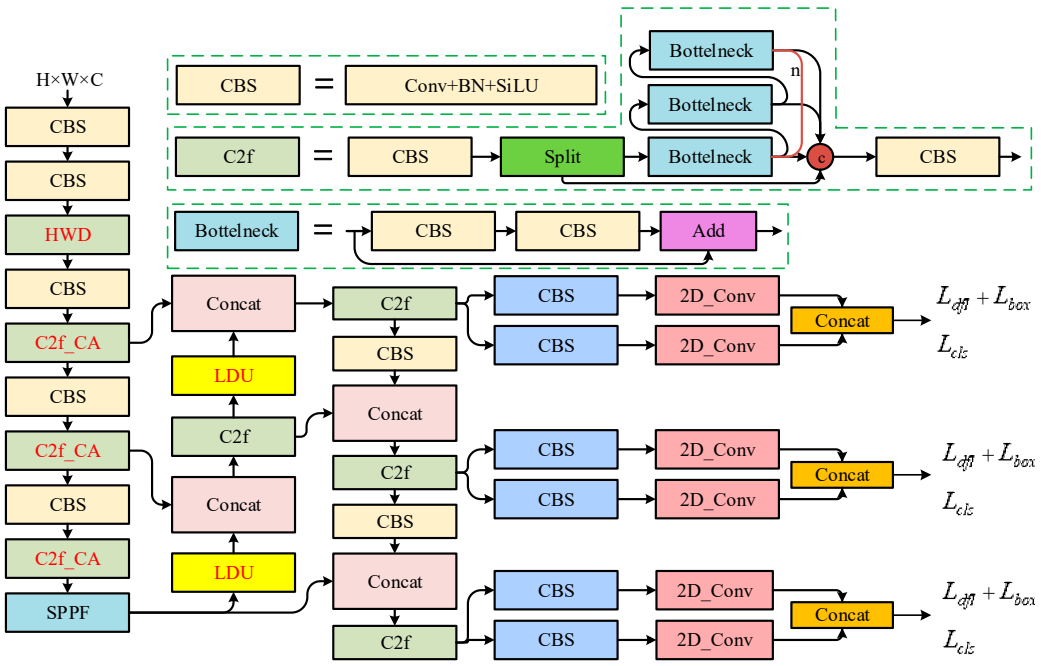


Figure 4. The network structure of SOCA-YOLO.

3.1. Haar Wavelet Downsample

The structure of the Haar wavelet downsampling (HWD) module is shown in Figure 5, which is composed of the Haar wavelet transform (HWT) and the convolution block (CBS).

Assume that the input image I has C channels of size $H \times W$. Row i can be represented as $I_i = [I_{i,0}, I_{i,1}, \cdots I_{i,W-1}]$. According to the sequence of channels, each row of the image is transformed by the one-dimensional Haar wavelet:

$$a_{i,j} = \frac{I_{i,2j} + I_{i,2j+1}}{\sqrt{2}}, \qquad j = 0,1,\cdots,\frac{W}{2}-1 \tag{1}$$

$$d_{i,j} = \frac{I_{i,2j} - I_{i,2j+1}}{\sqrt{2}}, \qquad j = 0,1,\cdots,\frac{W}{2}-1 \tag{2}$$

where $a_{i,j}$ and $d_{i,j}$ are the low-frequency and high-frequency coefficients of pixels (i,j), respectively.

Furthermore, $a_{i,j}$ and $d_{i,j}$ are made into low-frequency matrix A and high-frequency matrix D, and then each column of them is transformed by a one-dimensional Haar wavelet:

$$CA_{k,l} = \frac{A_{2k,l} + A_{2k+1,l}}{\sqrt{2}}, \qquad k = 0,1,\cdots,\frac{H}{2}-1, \quad l = 0,1,\cdots,\frac{W}{2}-1 \tag{3}$$

$$CH_{k,l} = \frac{A_{2k,l} - A_{2k+1,l}}{\sqrt{2}}, \qquad k = 0,1,\cdots,\frac{H}{2}-1, \quad l = 0,1,\cdots,\frac{W}{2}-1 \tag{4}$$

$$CH_{k,l} = \frac{D_{2k,l} + D_{2k+1,l}}{\sqrt{2}}, \qquad k = 0,1,\cdots,\frac{H}{2}-1, \quad l = 0,1,\cdots,\frac{W}{2}-1 \tag{5}$$

$$CD_{k,l} = \frac{D_{2k,l} - D_{2k+1,l}}{\sqrt{2}}, \qquad k = 0,1,\cdots,\frac{H}{2}-1, \quad l = 0,1,\cdots,\frac{W}{2}-1 \tag{6}$$

where k and l represent row and column indexes, respectively. CA, CH, CV, and CD represent the approximate component, horizontal detail component, vertical detail component, and diagonal detail component, respectively. These components form a temporary feature set $I'[H/2, W/2, 4C]$. Since this can be seen as a lossless coding process, the four components obtained after decomposition can contain more spatial features. In particular, the high-frequency detail features that damage the building are preserved. With

X' as the input of the CBS block, after convolution, batch normalization, and nonlinear activation of the SiLU function, the output feature set $F'(x)$ of HWD is obtained. After HWD, the image can be downsampled, while the spatial details are fully preserved, and the application of the CBS block also enhances the training stability of the model.

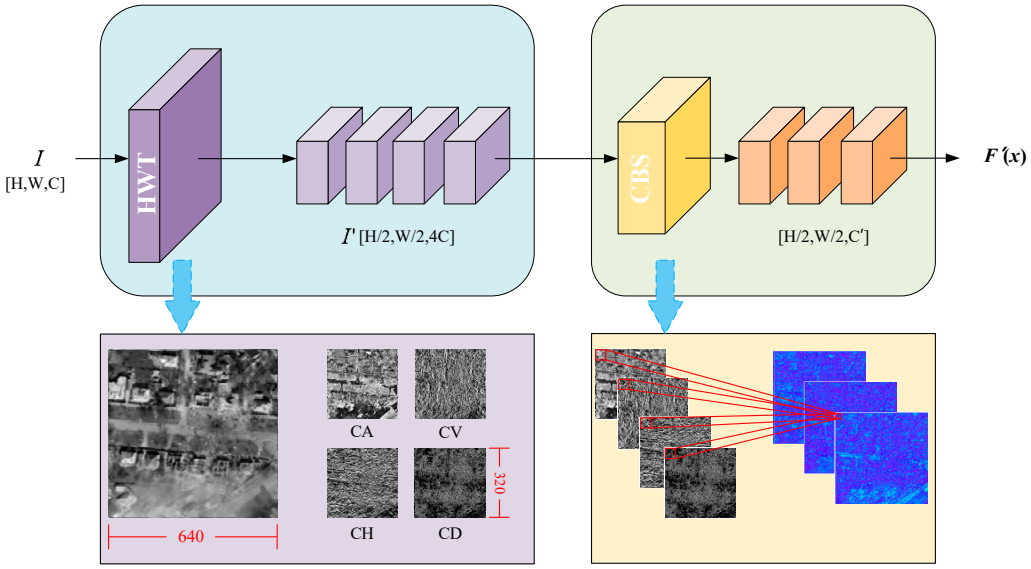


Figure 5. Structure of Haar wavelet downsampling module.

Figure 6 shows the visualization results of four downsampling methods, including Average pooling, Maximum pooling, Strided convolution, and HWD. It is evident that the image with Haar wavelet downsampling has clearer texture and shape features. In particular, the spatial details of the destroyed buildings in the red boxes are preserved to a greater extent.

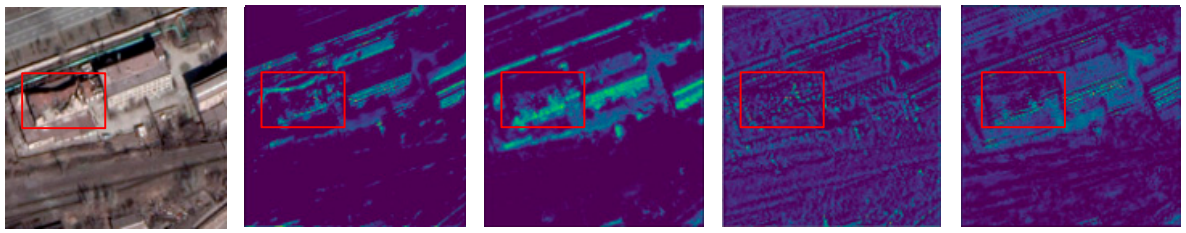


Figure 6. Visualization of different downsampling methods. The red box marks a damaged building in the input image. (**a**) Input image, (**b**) result of average pooling, (**c**) result of maximum pooling, (**d**) result of step convolution, (**e**) result of HWD.

3.2. Coordinate Attention

To further enhance the ability of the CNN to extract and locate the features of destroyed buildings, we constructed a C2f_CA module integrating coordinate attention, as shown in Figure 7. The CA module was input into Boottleneck through two stages of coordinate information generation and coordinate attention calculation to enhance spatial features and analyze dependency of inter-channel and inter-position dependency. The model's space awareness and feature representation ability are improved by stacking multiple CA_Bottleneck.

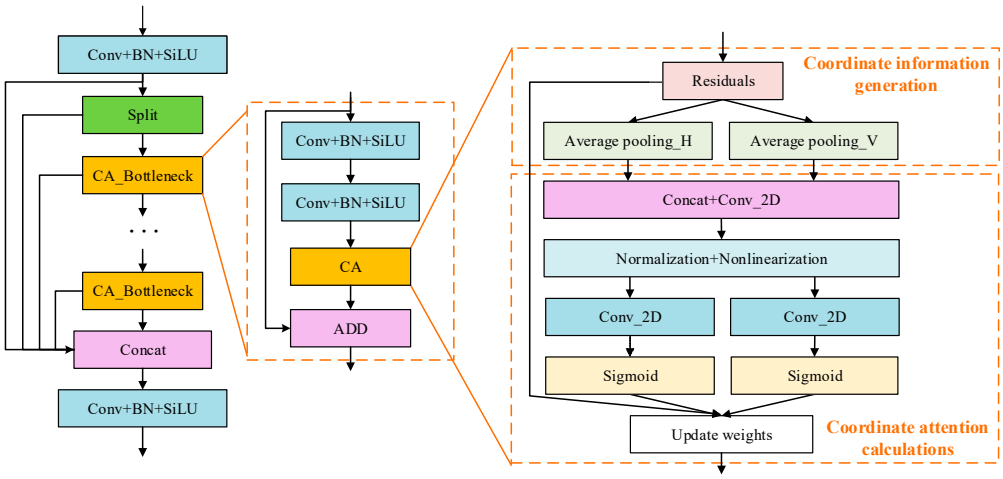


Figure 7. Structure of C2f_CA module.

In the generation stage of coordinate information, the CNN carries out one-dimensional spatial pooling in the horizontal and vertical directions of the input feature map, respectively, so that the attention module can capture accurate spatial position perception in different directions. In other words, while effectively capturing long-range dependencies in one spatial direction, the network can also obtain precise location information in the other direction, as shown in Equations (7) and (8):

$$z_c^h(h) = \frac{1}{W} \sum_{0 \le i < W} x_c(h, i) \tag{7}$$

$$z_c^w(w) = \frac{1}{H} \sum_{0 \le i < H} x_c(j, w) \tag{8}$$

where $z_c^h(h)$ and $z_c^w(w)$, respectively, represent the output by the c channel with height H in the vertical direction and the c channel with width W in the horizontal direction; $x_c(h, i)$ is the i pixel value in row h of channel c for the input feature map; and $x_c(j, w)$ is the j pixel value in column w of channel c for the feature map.

In the coordinate attention calculation stage, the channel dimension of the feature graph output in the previous stage is combined to make it have two independent spatial direction feature perceptions at the same time. The intermediate feature graph is further generated through 2D convolution, normalization, and nonlinearization:

$$f = \sigma(F_1([z^h, z^w])) \tag{9}$$

where f is the middle feature map; σ is the h_swish activation function; F_1 is a convolution operation; $[\cdot,\cdot]$ is the splicing channels; and z^h and z^w are the feature maps generated in the vertical and horizontal directions, respectively, in the previous stage.

The intermediate feature map is decomposed again into vertical and horizontal vectors f^h and f^w, and the number of channels is adjusted by convolution and activated by a Sigmoid function to generate the attention weights g^h and g^w in the corresponding directions. Finally, the attention feature map is obtained by weighting the two directions with the input feature map:

$$y(i, j) = x(i, j) \times g^h(i) \times g^w(j) \tag{10}$$

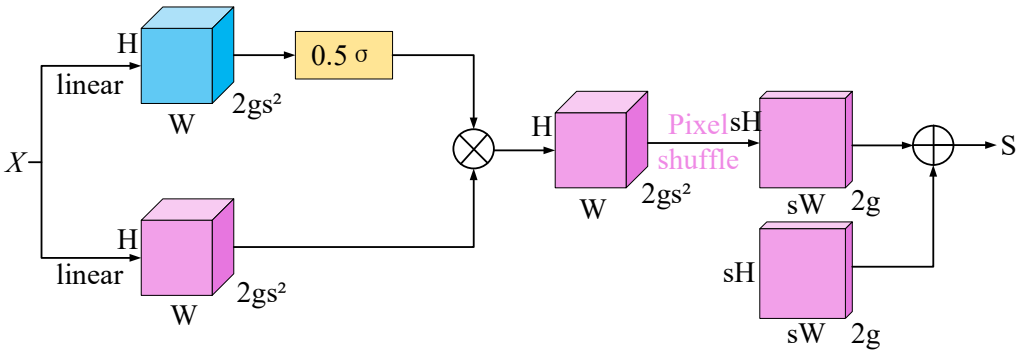
where $y(i, j)$ is the output attention feature map, $x(i, j)$ is the input feature map, and $g^h(i)$ and $g^w(j)$ are the vertical and horizontal attention weights, respectively.

With embedding coordinate attention in C2f, different weights can be obtained to different spatial positions of input feature maps, enhancing or suppressing feature

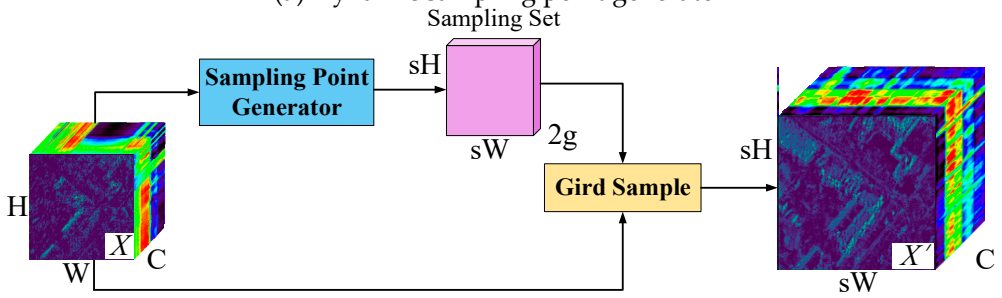
information. More importantly, the spatial context of local features can be better understood by the network, improving the utilization efficiency and recognition accuracy of spatial structure information.

3.3. Lightweight Dynamic Upsampling

The lightweight dynamic upsampling module consists of two parts: the Dynamic sampling point generator and Dynamic upsampling calculator, whose structures are shown in Figure 8a and Figure 8b, respectively.



(**a**) Dynamic sampling point generator



(**b**) Dynamic upsampling calculator

Figure 8. The structure of lightweight dynamic upsampling module.

The input feature map is denoted by X, whose size is $C \times H_1 \times W_1$, and the sample set S has size $2 \times H_2 \times W_2$, where C is the number of channels, H is the height of the image, W is the width of the image, and 2 in the sample set S is the x and y coordinates. Given the premise that the upsampling rate is s, the upsampling process can be expressed as

$$X' = grid\_sample(X, S) \tag{11}$$

where X' is the upsampling output.

Based on this, a linear offset layer O with the number of input channels C and the number of output channels $2s^2$ is introduced into the sampling module on Dysample. The coordinate S of each point in the feature graph X' can be obtained by

$$S = O + G \tag{12}$$

$$O = linear(X) \tag{13}$$

where G is the corresponding sampling point and O is its corresponding offset.

When O is non-0, the domain information of the sampling point can be taken into account. However, too large offset results in mixing other semantic cluster information when sampling the edge points of the semantic cluster, leading to the boundary being

indivisible. Therefore, multiplying the scaling factor γ in Equation (13) is used to limit the range of offset. After scaling and adjusting the offset, the coordinates are reordered by normalization and pixel recombination, and the final position coordinates are obtained by adding the positions of the corresponding sampling points. Finally, in order to make the offset process more flexible, the scaling factor is generated point by point by way of linear projection to adapt to different feature distributions:

$$0 = 0.5 \cdot ssigmoid(linear_1(X) \times linear_2(X)) \quad (14)$$

In this case, the offset range is [0, 0.5]. Compared with the traditional upsampling method, the lightweight dynamic upsampling module can retain the geometric feature information of the feature graph, and be used against boundary discontinuity and detail ambiguity caused by simple upsampling.

3.4. Slicing-Aided Hyper Inference

To detect destroyed buildings rapidly in a wide range of remote sensing images, the slice-assisted super inference strategy [48] is adopted in the process of network inference. The method cuts a wide range of remote sensing images into manageable sub-images, detects the destroyed building in sub-images one by one, and splices the detection results. It can greatly optimize memory usage and simplify the process of reasoning to visualize the geospatial location distribution of destroyed buildings.

4. Experimental Setting

4.1. Experimental Environment and Dataset Configuration

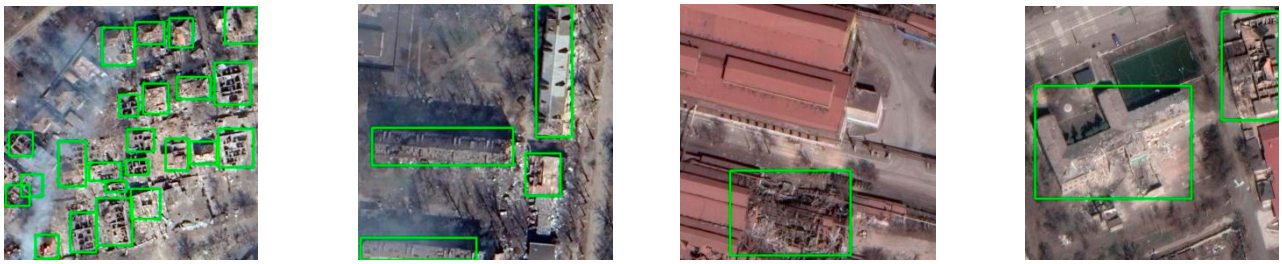
The experimental environment of this paper is as follows: Windows 10 operating system and Pytorch deep learning framework. The computer is configured as 12th Gen Intel(R) Core (TM) i7-12700H, 2.30 GHZ; 16 GB RAM, NVIDIA GeForce RTX 3050 graphics card, and 4G video memory.

In this work, we use 80% of the samples as the training set and the remaining 20% as the validation set. The data distribution is shown in Table 1.

Table 1. Distribution of experimental datasets.

	The Number of Sample Images	**The Number of Destroyed Buildings**	**Average Number of Destroyed Buildings per Image**
Train	3400	9348	2.75
Valid	851	2256	2.65

There are various types of scenes and objects in the dataset, including a high-density low-rise building area, low-density high-rise building area, industrial park, non-residential area, etc. At the same time, the size and shape of destroyed buildings differ greatly, and the characteristics of damage are not inconsistent (such as wall facade collapse and roof collapse), as shown in Figure 9, where the green boxes represent the ground truth of the destroyed building. Therefore, detecting destroyed buildings in complex and diverse scenarios is a challenging task.



Scene: Dense building region **Scene:** Sparse building region **Scene:** Industrial region **Scene:** Commercial region
Objects: Small house **Objects:** Large apartment **Objects:** Factory **Objects:** Large irregular building

Figure 9. The various scenes and various objects contained in the dataset. The green boxes represent the ground truth of the destroyed building.

4.2. Evaluation Indicators

The experiment verifies the effectiveness of the algorithm according to the evaluation indexes commonly used in object detection, including recall (R), precision rate (P), Average Precision (AP), $F1$ score, model size, parameter number, GFLOPs, and inference time. As shown in Formulas (15)–(18). Among them, the recall rate reflects the ability to detect the object from the image, and the higher the recall rate, the less missed detection. Precision reflects the ability to detect the correct object, and the higher the precision, the less false detection. AP is the definite integral of the $P-R$ curve, the $F1$ score is the harmonic average of recall and precision; they are all comprehensive indexes to evaluate the performance of model detection. model size, parameter number, GFLOPs, and inference time measure the efficiency of the model in real-world applications.

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

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

$$AP = \int_0^1 P(R)dR \tag{17}$$

$$F1 = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision} \tag{18}$$

where TP is the correctly predicted positive samples; $FN$is the incorrectly predicted negative samples; FP is the incorrectly predicted positive samples, and $P(R)$ is the curve of accuracy and recall rate. The larger the average precision, the better the detection performance of the model.

5. Result

5.1. Comparison of Different Improvement Strategies

Considering that HWD modules and the coordinate attention mechanism have various embeddedness positions and embeddedness ways, we designed several sets of comparative experiments to discuss the validity and rationality of the embeddedness positions and methods in the above methods.

(1) The embedding position of HWD

Previous experiments have proved that HWD can fully retain the spatial detail features of destroyed buildings when downsampling shallow feature maps. Therefore, in this section, we discuss whether consecutively embedding different numbers of HWD modules in a backbone network contributes to network performance. To be specific, we still use YOLOv8 as the baseline. HWD replaces the first C2f layer (Group A experiment, ours), the first two C2f layers (Group B experiment), and all C2f layers (Group C experiment) in the backbone network according to the sequence. The comparison experiment results are shown in Table 2.

Table 2. Experimental results of embedding HWD in different positions.

	Recall	Precision	AP50	Model Size/M	Number of Parameters	GFLOPs	Inference Time/ms

Baseline	0.661	0.667	0.705	5.96	3,005,843	8.2	5.2
Group A	0.698	0.660	0.712	5.95	3,002,771	7.8	5.6
Group B	0.687	0.652	0.708	5.88	2,970,131	7.3	6.3
Group C	0.660	0.634	0.682	4.73	2,380,563	6.3	7.0

It can be seen that the HWD module can significantly reduce the model size, model parameters, and required GFLOPs, which has the great advantage of being lightweight. When the HWD was used to replace the first C2f layer, the AP50 of the model increased by 0.7%, but with the continuous embedding of the HWD, the AP50 decreased. The reason is that the HWD can improve the expression of shallow spatial features such as edges and textures in the initial stage of the network and provide more effective feature input for subsequent layers. However, the description of deep feature maps is more dependent on semantic features, and the HWD is weak in extracting semantic features. Therefore, the continuous use of the HWD to replace the C2f layer will destroy the compatibility and consistency of the network structure, and thus reduce the detection accuracy.

(2) The embedding method of CA

To discuss the effectiveness of different embedding methods of the CA, we designed two sets of comparative experiments based on the Group A experiment. The first method is to take the CA as a single layer of the network (Group D experiment) and embed it into the last layer of the network's detection head, enhancing the detection head's focus on important features. The second method is to fuse the CA with a C2f layer in the backbone to constitute a C2f_CA module (Group E experiment, ours). The comparative experimental results are shown in Table 3.

Table 3. Experimental results of embedding CA in different ways.

	Recall	Precision	AP50	Model Size/M	Number of Parameters	GFLOPs	Inference Time/ms
Baseline	0.661	0.667	0.705	5.96	3,005,843	8.2	5.2
Group A	0.698	0.660	0.712	5.95	3,002,771	7.8	5.6
Group D	0.681	0.662	0.718	5.97	3,014,491	7.8	6.0
Group E	0.692	0.669	0.723	5.99	3,012,747	7.8	6.1

It can be seen that the AP50 of Group E is 0.5% higher than that of Group D. This is because C2f_CA can make full use of spatial information to enhance feature representation in the early stage of feature extraction and then make more effective use of global context information to generate richer spatial concern graphs on feature maps. In the follow-up inspection process, it is more helpful to accurately locate the destroyed building.

5.2. Ablation Experiment

Here, we design a group of ablation experiments to examine the effect of each component, as shown in Table 4. Experimental parameters were set as epochs: 100; initial learning rate: 0.01; IoU threshold: 0.5; batch size: 16; input image size: 640 × 640; the optimizer adopts SGD; no pre-training weights are used.

Table 4. The results of the ablation experiment.

HWD	C2f_CA	LDU	Recall	Precision	AP50	Model Size/M	Number of Parameters	GFLOPs	Inference Time/ms
×	×	×	0.661	0.667	0.705	5.96	3,005,843	8.2	5.2
√	×	×	0.691	0.660	0.712	5.95	3,002,771	7.8	5.6
×	√	×	0.657	0.671	0.708	5.41	2,695,075	6.9	5.6
×	×	√	0.688	0.649	0.710	5.39	2,696,715	6.8	5.3

√	√	×	0.692	0.669	0.723	5.99	3,012,747	7.8	6.1
√	√	√	0.670	0.669	0.730	6.01	3,025,099	7.9	6.0

(AP50 represents the detection accuracy of the destroyed building under the condition that the IoU is 0.5.)

It can be seen that adding the HWD, C2f_CA, and LDU separately improves the detection accuracy of the network, but the reasons are different. By retaining more spatial details at the beginning of the network calculation, the model with the HWD improved recall by 3.0%, indicating that it learned to more fully express the characteristics of the destroyed building. The model with C2f_CA can focus on the accurate expression of features, ensure the network learns more robust features, and improve the precision of the network. The introduction of the LDU can help the model to distinguish feature edges more effectively in the later stage, mainly in the improvement of recall.

In addition, in the ablation experiment, we observed that after adding the HWD, CA, and LDU to the baseline successively, the AP50 is successively increased by 0.7%, 1.8%, and 2.5%. It proves that the HWD module provides more spatial details for the subsequent feature extraction, and on this basis, C2f_CA can focus more on the expression of important features, and LDU can further distinguish the boundary distinction during the sampling process on the feature map. The information exchange of each module plays an active role in improving the detection accuracy of destroyed buildings.

At the same time, it is observed that the introduction of each module does not bring too much parameter number and calculation overhead, the model size is basically maintained at about 6.0 M, the number of model parameters is only increased by 0.59% compared with that of the baseline, and GFLOPs decreases from 8.2 to 7.9. It shows that our method can significantly improve the detection accuracy of destroyed buildings, as well as limit the increase in parameter number and algorithm complexity.

5.3. Visual Inspection

Figure 10 shows the detection results of each model in the ablation experiment under different scenarios. For large individual buildings, the improved model can more fully assess the size of the box, envelop destroyed buildings more completely, and improve confidence. Furthermore, it can also improve the recall rate in dense and small destroyed building scenes.

Figure 10. Visualization results of ablation experiments. The green boxes represent the ground truth of the destroyed building, the red boxes represent the correctly detected destroyed buildings, the yellow boxes represent the undetected destroyed buildings, and the blue circles represent the false alarm objects.

To illustrate the influence of each module on the attention area and degree of the network, we visualize the attention of the network detection layer for different sizes of objects, as shown in Figure 11.

For large destroyed buildings, we visualize the attention of the large-scale object detection layer. It is found that the attention of the baseline has inaccurate positioning and excessive noise response. With the HWD, the noise response near the object is significantly reduced. With the C2f_CA and LDU, the attention is further focused on the destroyed area, indicating that the introduction of each module can make the network pay more attention to the important features of the destroyed building area. In the detection of small destroyed buildings, heatmaps have the same form. Particularly, C2f_CA and LDU modules enable the network to reconstruct large images with more details on the basis of enhanced attention and improve the detection accuracy of small destroyed buildings.

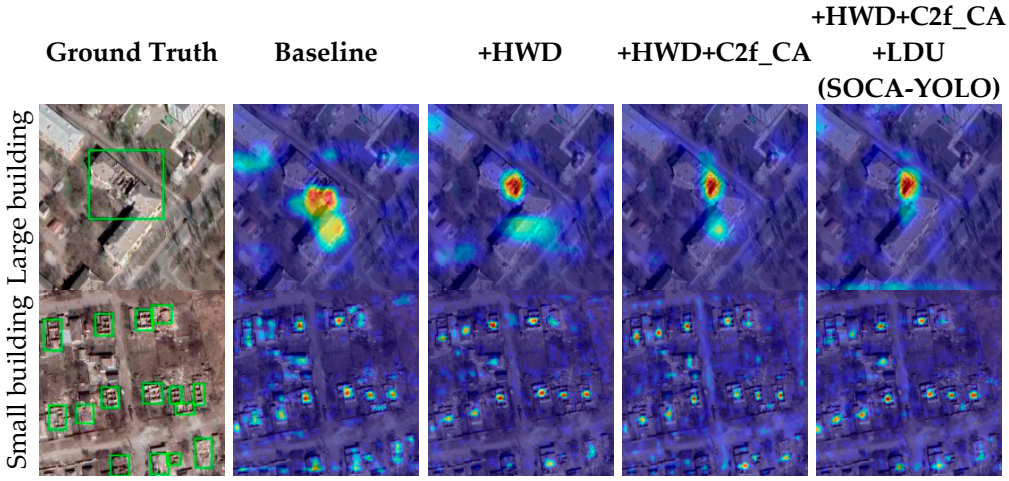


Figure 11. Attention visualization of CNN detection heads. The green boxes represent the ground truth of the destroyed building.

5.4. Comparison with Other Methods

To evaluate the comprehensive performance of the proposed method in terms of detection accuracy, model size, and detection efficiency, we compared the current mainstream same model size YOLO series algorithms, including YOLOv3-tiny, YOLOv5-n, YOLOv6-n, et al. As shown in Table 5. During the experiment, the parameter settings of each algorithm are consistent with those in Section 5.2. From the comparison results, it can be seen that the AP50 of SOCA-YOLO reaches 0.73, which is 0.3% higher than that of the second-best model, YOLOv3-tiny, but the model volume is much smaller than it.

Overall, SOCA-YOLO achieves a balance of model size, number of parameters, GFLOPs, and inference time while ensuring high-precision detection, and its comprehensive performance is the best.

Table 5. The detection results of various methods.

	Recall	Precision	AP50	Model Size/M	Number of Parameters	GFLOPs	Inference Time/ms
YOLOv3-tiny [19]	**0.695**	0.661	0.727	23.2	12128178	18.9	6.3
YOLOv5-n [21]	0.684	0.652	0.701	5.03	2503139	7.1	5.1
YOLOv6-n [22]	0.666	0.661	0.700	8.29	4233843	11.8	5.2
YOLOv8-n [24]	0.661	0.667	0.705	5.96	3005843	8.2	**5.0**
YOLOv9-t [25]	0.669	0.644	0.688	**4.41**	**1970979**	7.6	6.7
YOLOv10-n [26]	0.673	0.658	0.707	5.47	2694806	8.2	6.5
YOLOv11-n [24]	0.662	0.664	0.704	5.22	2582347	**6.3**	5.5
SOCA-YOLO	0.670	**0.669**	**0.730**	6.01	3025099	7.9	6.0

Bold represents the optimal value, underlining represents the sub-optimal value, and wavy lines represents the third optimal value.

5.5. Generalization Ability Test

To test the generalization ability of the model on high-resolution remote sensing images outside the dataset, we selected high-resolution remote sensing images from the Vornovakha region of Ukraine as test data, which were also sourced from Google Earth. The image size was 34,318 × 27,634 and was taken on 2 September 2022, containing a large number of buildings destroyed by the war. In the process of the generalization ability test, a slice-assisted super reasoning strategy was adopted to conduct sliding window detection on the entire remote sensing image and Mosaic it into a large-scale image. The detection results are shown in Figure 12, where the red boxes represent the correctly detected destroyed buildings, the yellow boxes represent the undetected destroyed buildings, and the blue boxes represent the false alarm objects. According to the results of the generalization ability test, as shown in Table 6, the overall recall rate is 0.708, indicating that there is a certain number of missed tests. In addition, the missing objects were mostly small destroyed buildings (recall rate was 0.655), indicating that the detection ability of the model for small destroyed buildings needs to be improved. On the other hand, the overall precision is 0.986 with fewer false alarms, indicating that the model has good robustness for feature extraction of destroyed buildings. The comprehensive index F1 score is 0.824, which verifies that the proposed method has a certain generalization ability for the detection of destroyed buildings under different scenarios.

Table 6. The results of generalization ability test.

Objects	Recall	Precision	F1 Score
Large destroyed buildings	1.000	0.941	0.967
Small destroyed buildings	0.655	0.950	0.775
ALL	0.708	0.986	0.824

According to the visualization results of geospatial information, the destroyed buildings on 2 September 2022 were mainly distributed on the east side of the railway line. The main positions of the warring parties in the city were located on the east side of the railway line, and the destroyed buildings in the north and the south were especially concentrated. The above results show that the method proposed in this paper can provide an important reference for rapid disaster assessment and humanitarian relief.

Figure 12. Detection results of destroyed buildings in Vornovakha (2 September 2022). The red boxes represent the correctly detected destroyed buildings, the yellow boxes represent the undetected destroyed buildings, and the blue boxes represent the false alarm objects.

6. Conclusions

In this study, we designed SOCA-YOLO, an object detection algorithm combining Haar wavelet downsampling, lightweight dynamic upsampling, and a coordinate attention mechanism to detect war-destroyed building objects in high-resolution remote sensing images. In the stage of feature extraction, Haar wavelet is used to decompose high-resolution remote sensing images in two dimensions to realize downsampling and retain the details of remote sensing images to a greater extent. In addition, coordinate attention is combined with C2f to make full use of spatial feature information, and a coordinate attention module is introduced in the early stage of feature extraction to achieve feature optimization and focus to a greater extent. Finally, a lightweight dynamic upsampling module is used in the feature fusion phase to further enrich and accurately detail information on the feature map.

To evaluate the effectiveness of the method, we collected high-resolution optical remote sensing images of the Mariupol region, Ukraine, and produced the world's first datasets for the detection of buildings destroyed by war. The proposed method was verified by ablation experiments, and the validity and reasons for the embedding positions and methods of HWD and CA were discussed through multiple sets of comparative experiments. Compared with the classical deep learning object detection algorithm, the results show that the proposed method has better comprehensive performance in detection accuracy, model size, and inference speed.

It is worth noting that SOCA-YOLO mainly focuses on detecting individual destroyed buildings by making full use of spatial details and enhancing spatial perception. However, it still has some limitations: it is limited by the size and region of the training samples, the architectural styles of different countries and regions have obvious differences, and the generalization performance of the model needs to be further improved. In addition, the model has limited recognition of difficult objects, some destroyed buildings that look like ruins are still missed, and some structurally complex buildings are susceptible to false detection. In addition, the lightweighting and reasoning speed of the model need to be improved. In our following work, we will consider classifying destroyed buildings according to the degree of damage. By combining a feature recombination strategy and difficult object mining, fine-grained identification of buildings with different degrees of damage has become another exciting issue.

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

Funding: This research received no external funding

Data Availability Statement: The data presented in this study are available on request from the corresponding author due to (specify the reason for the restriction).

Acknowledgments: We would like to thank Google Earth and MAXAR for providing high-resolution optical remote sensing image data. We would also like to thank ultralytics and its users for their contributions to the YOLO series of object detection algorithms. Finally, thanks to the reviewers and editors who put forward valuable suggestions for improving the quality of this paper.

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

References

1. Serhii, A.S.; Vyshnevskyi, V.I.; Olena, P.B. The Use of Remote Sensing Data for Investigation of Environmental Consequences of Russia-Ukraine War. *J. Landsc. Ecol.* **2022**, *15*, 36–53. https://doi.org/10.2478/jlecol-2022-0017.
2. ICRC. *Urban Services during Protracted Armed Conflflict: A Call for a Better Approach to Assisting Affected People*; International Committee of the Red Cross: Geneva, Switzerland, 2015.
3. Alacevich, M. Planning Peace: The European Roots Of The Post-War Globel Development Cchallenge. *Past & Present*. **201**8, 239, https://doi.org/10.1093/pastj/gtx065.
4. Kashtelan, S.O. About analysis of global trends of modern armed struggle. *Collect. Sci. Work. Mil. Inst. Kyiv Natl. Taras Shevchenko Univ.* **2023**, *79*, 7–12. https://doi.org/10.17721/2519-481x/2023/79-01.
5. A history of the Israeli-Palestinian conflict. *Choice Rev. Online* **2010**, *47*, 47–3990. https://doi.org/10.5860/choice.47-3990.
6. Sarjoon, A.; Yusoff, M.A. The United Nations peacekeeping operations and challenges. *Acad. J. Interdiscip. Stud.* **2019**, *8*, 202–211. https://doi.org/10.36941/ajis-2019-0018.
7. Mello, P.A. The Oxford handbook of United Nations peacekeeping operations. *Political Stud. Rev.* **2017**, *15*, 106–107. https://doi.org/10.1177/1478929916676956.
8. Dong, L.; Shan, J. A comprehensive review of earthquake-induced building damage detection with remote sensing techniques. *ISPRS J. Photogramm. Remote Sens.* **2013**, *84*, 85–99. https://doi.org/10.1016/j.isprsjprs.2013.06.011.
9. Boloorani, A.D.; Darvishi, M.; Weng, Q.; Liu, X. Post-War Urban Damage Mapping Using InSAR: The Case of Mosul City in Iraq. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 140. https://doi.org/10.3390/ijgi10030140.
10. Janalipour, M.; Mohammadzadeh, A. A novel and automatic framework for producing building damage map using post-event LiDAR data. *Int. J. Disaster Risk Reduct.* **2019**, *39*, 101238. https://doi.org/10.1016/j.ijdrr.2019.101238.
11. Huang, Q.; Jin, G.; Xiong, X.; Ye, H.; Xie, Y. Monitoring Urban Change in Conflict from the Perspective of Optical and SAR Satellites: The Case of Mariupol, a City in the Conflict between RUS and UKR. *Remote Sens.* **2023**, *15*, 3096. https://doi.org/10.3390/rs15123096.
12. Ghandour, A.J.; Jezzini, A.A. Post-War Building Damage Detection. *Proceedings* **2018**, *2*, 359. https://doi.org/10.3390/ecrs-2-05172.
13. Zitzlsberger, G.; Podhoranyi, M. Monitoring of Urban Changes With Multimodal Sentinel 1 and 2 Data in Mariupol, Ukraine, in 2022/23. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 5245–5265. https://doi.org/10.1109/JSTARS.2024.3362688.
14. Aimaiti, Y.; Sanon, C.; Koch, M.; Baise, L.G.; Moaveni, B. War Related Building Damage Assessment in Kyiv, Ukraine, Using Sentinel-1 Radar and Sentinel-2 Optical Images. *Remote Sens.* **2022**, *14*, 6239. https://doi.org/10.3390/rs14246239.
15. Gupta, R.; Goodman, B.; Patel, N.; Hosfelt, R.; Sajeev, S.; Heim, E.; Doshi, J.; Lucas, K.; Choset, H.; Gaston, M. Creating xBD: A Dataset for Assessing Building Damage from Satellite Imagery. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Long Beach, CA, USA, 16–20 June 2019. Available online: https://arxiv.org/pdf/1911.09296 (accessed on 11 October 2024).
16. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031.
17. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot MultiBox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37. https://doi.org/10.1007/978-3-319-46448-0_2.
18. Redmon, J.; Farhadi, A. YOLO9000: Better faster stronger. In Proceedings of the IEEE Conference On Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
19. Farhadi, A.; Redmon, J. YOLOv3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
20. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
21. yolov5. Available online: https://github.com/ultralytics/yolov5 (accessed on 24 October 2024).
22. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; Li, Y. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
23. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bagof- freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
24. ultralytics. Available online: https://github.com/ultralytics/ultralytics (accessed on 24 October 2024).
25. Wang, C.Y.; Yeh, I.H.; Mark Liao, H.Y. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. In *Computer Vision—ECCV 2024*; Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2025; Volume 15089. https://doi.org/10.1007/978-3-031-72751-1_1.

26. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. *arXiv* **2024**, arXiv:2405.14458.
27. Beal, J.; Kim, E.; Tzeng, E.; Park, D.H.; Zhai, A.; Kislyuk, D. Toward transformer-based object detection. *arXiv* **2020**, arXiv:2012.09958.
28. 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 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 17–21 June 2024; pp. 16965–16974. https://doi.org/10.1109/CVPR52733.2024.01605.
29. Ji, M.; Liu, L.; Zhang, R.; Buchroithner, M.F. Discrimination of Earthquake-Induced Building Destruction from Space Using a Pretrained CNN Model. *Appl. Sci.* **2020**, *10*, 602. https://doi.org/10.3390/app10020602.
30. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
31. Wu, C.; Zhang, F.; Xia, J.; Xu, Y.; Li, G.; Xie, J.; Du, Z.; Liu, R. Building Damage Detection Using U-Net with Attention Mechanism from Pre- and Post-Disaster Remote Sensing Datasets. *Remote Sens.* **2021**, *13*, 905. https://doi.org/10.3390/rs13050905.
32. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015*; Springer International Publishing: Berlin/Heidelberg, Germany, 2015. https://doi.org/10.1007/978-3-319-24574-4_28.
33. Shi, L.; Zhang, F.; Xia, J.; Xie, J.; Zhang, Z.; Du, Z.; Liu, R. Identifying Damaged Buildings in Aerial Images Using the Object Detection Method. *Remote Sens.* **2021**, *13*, 4213. https://doi.org/10.3390/rs13214213.
34. Bai, Y.; Hu, J.; Su, J.; Liu, X.; Liu, H.; He, X.; Meng, S.; Mas, E.; Koshimura, S. Pyramid Pooling Module-Based Semi-Siamese Network: A Benchmark Model for Assessing Building Damage from xBD Satellite Imagery Datasets. *Remote Sens.* **2020**, *12*, 4055. https://doi.org/10.3390/rs12244055.
35. Ma, H.; Liu, Y.; Ren, Y.; Yu, J. Detection of Collapsed Buildings in Post-Earthquake Remote Sensing Images Based on the Improved YOLOv3. *Remote Sens.* **2020**, *12*, 44. https://doi.org/10.3390/rs12010044.
36. Ma, N.; Zhang, X.; Zheng, H. ShuffleNet v2: Practical guidelines for efficient CNN architecture design. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 116–131.
37. Rezatofighi, H.; Tsoi, N.; Gwak, J.Y. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Los Angeles, CA, USA, 16–19 June 2019. 658–666.
38. Lu, W.; Wei, L.; Nguyen, M. Bitemporal Attention Transformer for Building Change Detection and Building Damage Assessment. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 2024, 17, 4917. https://doi.org/10.1109/JSTARS.2024.3354310.
39. Liu, C.; Sepasgozar, S.M.; Zhang, Q.; Ge, L. A novel attention-based deep learning method for post-disaster building damage classification. *Expert Syst. Appl.* **2022**, *202*, 117268. https://doi.org/10.1016/j.eswa.2022.117268.
40. Hou, Q.; Zhou, D.; Feng, J. Coordinate Attention for Efficient Mobile Network Design. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 19–25 June 2021; pp. 13708-13717. https://doi.org/10.1109/CVPR46437.2021.01350.
41. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944. https://doi.org/10.1109/CVPR.2017.106.
42. 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. 10778–10787. https://doi.org/10.1109/CVPR42600.2020.01079.
43. Zhang, H.; Zhang, Y.; Wang, D.; Ma, G. Damaged Building Object Detection From Bi-temporal Remote Sensing Imagery: A Cross-task Integration Network and Five Datasets. *IEEE Trans. Geosci. Remote Sens.* **2024**, *60*, 5648827. https://doi.org/10.1109/TGRS.2024.3493886.
44. Xu, G.; Liao, W.; Zhang, X.; Li, C.; He, X.; Wu, X. Haar wavelet downsampling: A simple but effective downsampling module for semantic segmentation. *Pattern Recognit.* **2023**, *143*, 109819. https://doi.org/10.1016/j.patcog.2023.109819.
45. Liu, W.; Lu, H.; Fu, H.; Cao, Z. Learning to Upsample by Learning to Sample. In Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, Paris, France, 2–6 October 2023; pp. 6004–6014. https://doi.org/10.1109/ICCV51070.2023.00554.
46. High Commissioner Updates the Human Rights Council on Mariupol, Ukraine. Available online: https://www.ohchr.org/en/statements/2022/06/high-commissioner-updates-human-rights-council-mariupol-ukraine (accessed on 11 October 2024).

47. labelimg. Available online: https://github.com/HumanSignal/labelImg (accessed on 7 October 2024).
48. Akyon, F.C.; Altinuc, S.O.; Temizel, A. Slicing Aided Hyper Inference and Fine-Tuning for Small Object Detection. In Proceedings of the 2022 IEEE International Conference on Image Processing (ICIP), Bordeaux, France, 16–19 October 2022; pp. 966–970. https://doi.org/10.1109/ICIP46576.2022.9897990.