remote sensing

Article

A Novel Pre-Processing Approach and Benchmarking Analysis for Faster, Robust, and Improved Small Object Detection Methods

Mohammed Ali Mohammed Al-Hababi [1], Ahsan Habib [2], Fursan Thabit [1] and Ying Liu [1,*]

[1] School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 101408, China; mohammed_al-hababi@mails.ucas.ac.cn (M.A.M.A.-H.)
[2] Technovative Solutions LTD (TVS), Manchester M15 6JJ, UK
* Correspondence: yingliu@ucas.ac.cn; Tel.: +86-(10)-6967-1791

Abstract: Detecting tiny objects in aerial imagery presents a major challenge regarding their limited resolution and size. Existing research predominantly focuses on evaluating average precision (AP) across various detection methods, often neglecting computational efficiency. Furthermore, state-of-the-art techniques can be complex and difficult to understand. This paper introduces a comprehensive benchmarking analysis specifically tailored for enhancing small object detection within the DOTA dataset, focusing on one-stage detection methods. We propose a novel data-processing approach to enhance the overall AP for all classes in the DOTA-v1.5 dataset using the YOLOv8 framework. Our approach utilizes the YOLOv8's darknet architecture, a proven effective backbone for object detection tasks. To optimize performance, we introduce innovative pre-processing techniques, including data formatting, noise handling, and normalization, in order to improve the representation of small objects and improve their detectability. Extensive experiments on the DOTA-v1.5 dataset demonstrate the superiority of our proposed approach in terms of overall class mean average precision (mAP), achieving 66.7%. Additionally, our method establishes a new benchmark regarding computational efficiency and speed. This advancement not only enhances the performance of small object detection but also sets a foundation for future research and applications in aerial imagery analysis, paving the way for more efficient and effective detection techniques.

Keywords: small object detection; remote sensing object detection; one stage; DOTA-v1.5; YOLO-v8

check for updates

Citation: Al-Hababi, M.A.M.; Habib, A.; Thabit, F.; Liu, Y. A Novel Pre-Processing Approach and Benchmarking Analysis for Faster, Robust, and Improved Small Object Detection Methods. *Remote Sens.* **2024**, *16*, 3753. https://doi.org/10.3390/rs16203753

Academic Editor: Konstantinos Topouzelis

Received: 11 September 2024
Revised: 4 October 2024
Accepted: 7 October 2024
Published: 10 October 2024

1. Introduction

Small object detection in remote sensing involves identifying and precisely locating small-sized features in large-scale aerial or satellite images [1]. In applications like urban monitoring, traffic planning, agriculture, maritime surveillance, disaster management, military operations, and environmental conservation, small objects offer crucial insights. These objects, such as small buildings, vehicles, trees, or specific land-use patterns, serve as indicators of significant phenomena or Earth's surface changes [2]. In urban planning, environmental monitoring, disaster management, and precision agriculture, small objects like buildings aid in assessing population density, monitoring vegetation health, estimating yields, and tracking land-use changes. They play a pivotal role in understanding transformations over time, enabling the observation of deforestation, climate change impact, and other crucial developments.

Detecting tiny instances in remote imagery is particularly challenging due to variations, occlusion, and low contrast in spectral and shape characteristics [3]. These challenges are further exacerbated by illumination variations, atmospheric effects, and noise, all of which underscore the importance of accurate detection for enabling informed decision making. Annotated training datasets, although requiring considerable labor for their creation, serve as critical resources for the development of effective methods [4]. Researchers commonly employ one-stage and two-stage methods for tiny object detection. One-stage methods, like SSD (single shot detector) and YOLO (You Only Look Once), predict object bounding boxes and class labels in a single pass, offering efficiency but potentially struggling with localizing small objects [5–7]. Two-stage methods, such as Faster R-CNN, follow a two-step approach, achieving significant success in small object detection tasks [8].

The choice between one- and two-phase methods depends on the specific requirements of the task. One-stage methods, with their simpler architecture, offer calculation efficacy and are well-suited for autonomous driving and video analysis [9,10]. They excel in detecting and localizing large, well-represented objects, demonstrating higher recall rates for significant-sized, clearly visible items. One-stage methods may enhance generalization to unseen data with fewer parameters and also reduce overfitting [11]. Additionally, one-stage methods simplify the training process by allowing for a direct, end-to-end approach, eliminating the requirement for separate stages of region proposals and object detection, which reduces the associated computational expenses. The complexity of small object detection in remote sensing arises from several factors such as low occlusion, contrast, and variations in spectral and shape characteristics [12–15]. Additionally, atmospheric effects, illumination variations, and noise further complicate the task [16]. Despite these challenges, accurate small object detection is vital for extracting meaningful information from remote sensing imagery and enabling informed decision making across various applications. Existing research for small object detection lacks proper investigations in terms of speed, time, and score analysis. Most of the existing studies focus either on two stages or both stages at a time. As a result, it is nearly impossible to identify the state-of-the-art one-stage method based on speed, time, and score.

Therefore, this study solely focuses on one-stage methods. In order to address the challenges of small object detection in remote sensing, this study proposes a novel pre-processing method to train YOLOv8 to achieve better performance metrics. This study also conducts a benchmarking analysis on one-stage methods for tiny object detection in terms of time, accuracy, and speed. The pre-processing methods focus on enhancing the quality of data for better object detection. This tackles noise reduction and data normalization, indirectly enhancing contrast by making object features more distinguishable from the background. Standardizing the dataset and removing noise improves the clarity of object boundaries, thereby aiding in handling occlusion to some extent. This method streamlines data formatting by converting object names into numerical representations, standardizing the representation of objects with different shapes and spectral characteristics and thus aiding in detection. This method explicitly addresses noise by employing regular expressions to remove extraneous strings from the dataset, ensuring that the model is trained on clean and relevant data. Additionally, data normalization helps mitigate the effects of illumination variations by scaling dataset values within a standardized range, making the model less sensitive to such variations during training. Moreover, the proposed method contributes to understanding the state of the art not only through the mean average precision but also in terms of speed and computational efficiency. This comprehensive evaluation aids in selecting the most practical method for real-world implementation, considering all relevant parameters.

In summary, the main objective of this research is to perform a benchmarking analysis of the existing one-stage object detection method under the same environment and to propose a pre-processing method aimed at achieving better mean average precision for all classes. The main contributions of this paper are as follows:

- Pre-processing: a novel pre-processing method is proposed to train the YOLOv8 model with the DOTA v1.5 dataset in order to achieve a better mean averaged precision.
- Benchmarking analysis: a benchmarking analysis is performed with one-stage methods for small object detection DOTA-v1.5 datasets.

The rest of this article is structured as follows. Related works are analyzed in Section 2. The methodology is delineated in Section 3. Section 4 illustrates the performance evaluation and discussion section. Finally, Section 5 concludes this article.

2. Related Work

This section describes the studies related to one-stage object detection. Table 1 shows the comparison between the proposed studies and other studies.

Table 1. An overview of existing research on one-stage object detection. Here, AP means average precision. FPS means frame per second. ✓ means that the studies support the parameter. - means the study does not support the parameter.

Studies	DOTAv	AP Analysis	Speed Analysis	Time Analysis	FPS Analysis	100 Epoch
Yang et al. [17]	1	✓	-	-	-	-
Wang et al. [18]	1	✓	-	-	✓	-
Qian et al. [19]	1	✓	-	-	-	-
Yassin et al. [20]	1	✓	-	-	-	-
Li et al. [21]	1	✓	-	-	✓	-
Qian et al. [22]	1	✓	-	-	-	-
Hou et al. [23]	1	✓	-	-	✓	-
Lin et al. [24]	1.5	✓	-	-	-	-
Cao et al. [25]	1	✓	-	-	-	-
This study	1.5	✓	✓	✓	✓	✓

Yang et al. [17] propose the Small, Cluttered, and Rotated Object Detector++ (SCRDet++), focusing on reducing noise in object detection, particularly for small and crowded objects. They perform individual-level denoising on the feature map in order to improve detection accuracy. Wang et al. [18] introduce feature-merged single-shot detection (FMSSD), a comprehensive framework that combines contextual information from various scales by using the atrous spatial feature pyramid (ASFP) module. In addition, they also adjust the loss function to give priority to small objects. Qian et al. [19] introduced rotated object detection with RSDet, offering advantages such as an adjusted rotation loss and predicting object corners and thus improving performance. Jiang et al. [26] present an Information Balanced Fusion Network (IBFF), a detector for small objects operating at multiple scales, featuring different attention-based context feature fusion (DACFF) modules. Zakaria et al. [20] integrate Instance Level Denoising (ILD) from SCRDet++ into S2A-Net.

Cheng et al. [27] present the Anchor-Free Oriented Proposal Generator (AOPG), eliminating horizontal box-related operations by utilizing a Coarse Location Module (CLM) for initial coarse-oriented box generation without anchors. A Fast Region-based Convolutional Neural Network (R-CNN) head refines these boxes for high-quality oriented proposals. Li et al. [21] propose the Dense Path Aggregation Feature Pyramid Network (DPAFPN) as a single-stage detector for remote sensing data. It aims to use both high-level semantic and low-level location information of the images. Qian et al. [22] suggest a Unified Transferring Strategy (UTS) for bounding box regression (BBR) in oriented object detection, introducing Rotated-Intersection of Union (RIoU) loss. Chen et al. [28] extend Faster R-CNN with Weighted Fusion and Refinement (WFR), Affine Transformation-Based Feature Decoupling (ATFD), and Post-Classification Regression (PCR) modules for improved performance.

Gao et al. [29] propose a repulsion constraint for point representation, assessing centeredness quality and introducing oriented repulsion regression for densely packed targets in remote sensing. Hou et al. [23] present G-Rep, a unified representation using Gaussian distributions for the OBB, QBB, and PointSet, optimizing parameters through maximum likelihood estimation. Wei et al. [30] offer a lightweight method for proposals of arbitrary-oriented objects, using a rotated region proposal network and a rotation-equivariant backbone. Lin et al. [24] augment foreground features in a one-stage object detection system by including a keypoint attention module and a prototype contrastive learning module. Cao et al. [25] integrate semantic edge detection with arbitrary-oriented object detection, introducing a feature-enhancement network and a rotation-invariant spatial pooling pyramid. Zheng et al. [31] proposed crossNet, an end-to-end deep neural network using cross-scale warping, which improves reference-based super-resolution accuracy and efficiency by performing spatial alignment at the pixel level. Law et al. [32] proposed cornerNet, a single convolution neural network, which effectively detects objects as paired key points, outperforming the existing one-stage detectors on MS COCO with a 42.2% accuracy. Duan et al. [33] proposed centerNet, which improves object detection precision and recall by detecting each object as a triplet of key points, outperforming existing one-stage detectors by at least 4.9%.

The studies referenced above primarily emphasize both one-stage and two-stage detection methods, focusing exclusively on the mean average precision (mAP) without considering other crucial factors such as speed, processing time, and additional relevant parameters. However, a comprehensive evaluation of all metrics is essential to gain a more thorough understanding of the performance of these approaches. To address this gap, the current study introduces a novel pre-processing approach designed to improve the training process of YOLOv8. Additionally, a comprehensive benchmarking evaluation was conducted on the DOTA-v1.5 dataset to assess the effectiveness of the proposed approach in enhancing the performance of small object detection against state-of-the-art one-stage methods.

3. Methodology

The computer environment utilized for the studies has an Intel(R) Core(TM) i7-9700 CPU running at 3.00 GHz, 32.0 GB of RAM (31.8 GB useable), and runs on a 64-bit Windows 11 Pro system with version 22H2 (OS build 22621.1702) and Windows Feature Experience Pack 1000.22641.1000.0. The experimental server configuration is far more robust, with improved connectivity and computational capabilities.

It comprises network connectivity with four InfiniBand 100 Gbps EDR and two 10 GbE connections. The server uses 8x NVIDIA Tesla V100 GPUs, each with 16 GB of RAM, for a total of 40,960 NVIDIA CUDA cores and 5120 Tensor cores. These GPUs are linked together via the NVIDIA NVLink Hybrid Cube Mesh, which ensures high-bandwidth communication between them. The system memory is significant, comprising 512 GB DDR4 LRDIMM, and the CPU configuration comprises two 20-core Intel Xeon E5-2698 v4 processors operating at 2.2 GHz.

The server's storage subsystem has four 1.92 TB SSDs deployed in a RAID 0 array, giving fast data access and a total storage capacity of 7.68 TB. The power needs are handled by four 1600 W power supply units (PSUs) with a combined thermal design power (TDP) of 3500 W, which provides enough power for the high-performance components. The system's cooling is tuned for optimal front-to-back airflow, ensuring stable operation even under high computational loads. This high-performance configuration shown in Table 2 allows for full benchmarking and analysis, which supports the study's need to efficiently handle massive amounts of data and sophisticated computations.

Table 2. The experimental setup for the benchmarking analysis.

System 1	Windows 11 Pro edition, version 22H2, 64-bit
GPU	4 GB NVIDIA GeForce GTX 1050 Ti
RAM	32.0 GB DDR4
CPU	Intel(R), Core(TM), i7-9700 CPU @ 3.00 GHz
System 2	Ubuntu 18.4 server
GPUs	8x 32 GB NVIDIA Tesla V100
RAM	512 GB DDR4 LRDIMM
CPUs	2x 20-Core Intel Xeon E5-2698 v4 2.2 GHz
GPU interconnect	NVIDIA NVLink Hybrid Cube Mesh
Storage	4x 1.92 TB SSDs RAID
Cooling	Efficient Front-to-Back Airflow

Table 2. *Cont.*

Network interconnect	4x InfiniBand 100 Gbps EDR 2x 10 GbE
Power	4x 1600 W PSUs (3500 W TDP)

3.1. Datasets

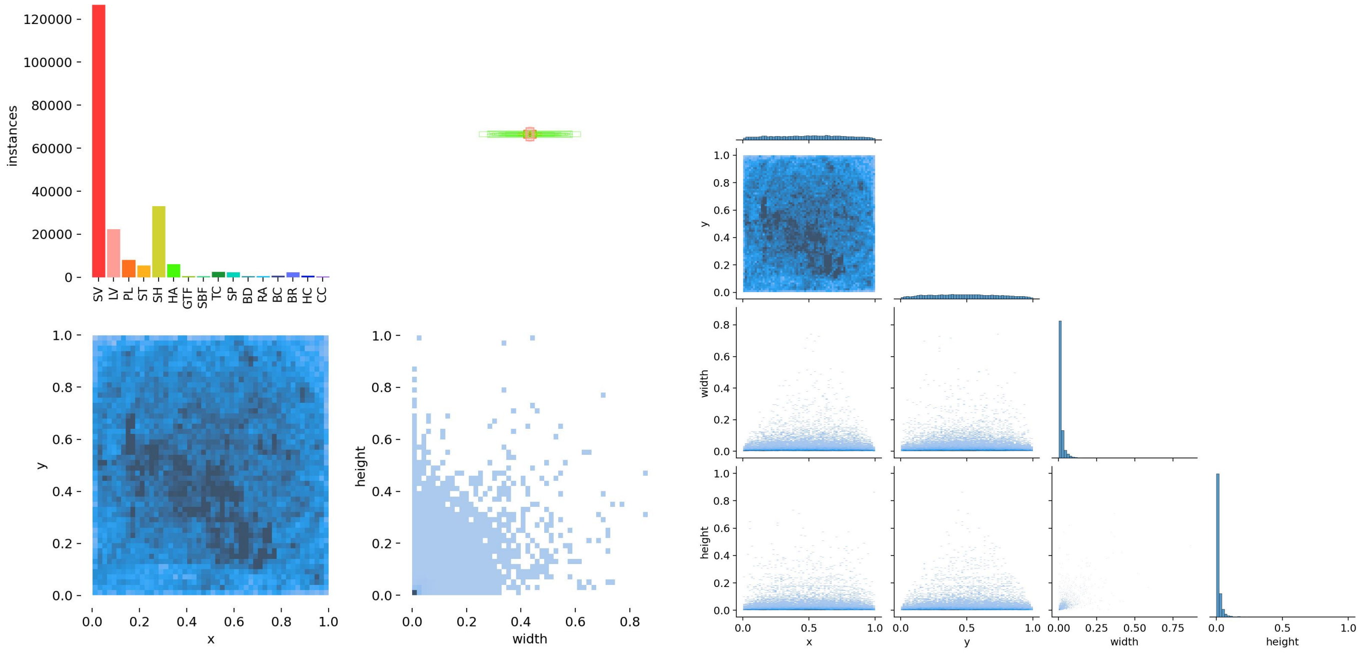
The identification in aerial images (DOTA) dataset is a well-known benchmark in the field of object identification, designed particularly for high-resolution aerial images. It has contributed significantly to the development and assessment of object detection algorithms. The DOTA dataset has gone through multiple revisions, with each iteration bringing new features that improve its usability for academics and practitioners. The following is a complete summary of the many versions of the DOTA dataset, highlighting their contributions and advancements.

DOTA-v1.0, released in 2018, was the first version of the DOTA dataset. This first edition includes 2806 aerial photographs taken in a variety of geographic regions and settings, including urban and rural areas. The collection includes annotations for 15 item categories, covering a wide range of real-world things typically seen in aerial images. The categories include airplanes (PL), ships (SH), storage tanks (ST), and basketball courts (BC), among others. Each object in the photos is tagged with bounding box coordinates and categorization names, making it easier to create and test object recognition algorithms. DOTA-v1.0 provided a fundamental dataset for assessing object identification algorithms in aerial photos, answering the demand for high-resolution, diversified, and annotated datasets. The annotations in this version were created to help researchers train and test object detection algorithms, allowing them to compare their predictions to a consistent collection of data.

Building on the success of DOTA-v1.0, DOTA-v1.5 was released as an expansion of the original dataset. DOTA-v1.5, which included enhanced annotations, was designed to improve both the precision and dependability of item labeling. While the dataset size remained comparable with DOTA-v1.0, improved annotations resulted in higher coverage and more exact classifications of items inside the photos. DOTA-v1.5 aimed to solve problems identified in the previous version, notably in terms of annotation quality and object categorization. This version sought to remove ambiguities and inconsistencies in the annotations, which would improve the performance of detectors for objects trained on the dataset. The improved annotations made it easier to evaluate model performance and helped to progress object recognition algorithms in aerial photography.

The DOTA-v2.0 version, published in 2019, significantly expanded the dataset. This iteration retains the original 2806 photos while making numerous significant changes. One of the most important innovations to DOTA-v2.0 was the introduction of a new object category, the backdrop class, which increased the overall number of object categories to 15 + 1. This update was intended to offer a more thorough portrayal of the many objects and backdrops found in aerial images. The annotations in DOTA-v2.0 were improved, increasing both the accuracy and coverage of item tagging. This version also added a wider range of item categories and enhanced annotation consistency, resulting in a more rigorous benchmark for assessing object detection methods. Better annotations and an enlarged dataset made it possible to compare and analyze model performance in more detail, which aided in the creation of increasingly sophisticated object recognition techniques.

The current work makes use of DOTA-v1.5, which provides a comprehensive collection of classifications of objects for evaluation and building models. This version includes the following object categories: bridge (BR), helicopter (HC), storage tank (ST), soccer ball field (SBF), small vehicle (SV), plane (PL), large vehicle (LV), ground track field (GTF), tennis court (TC), ship (SH), swimming pool (SP), container crane (CC), basketball court (BC), harbor (HA), roundabout (RA), and baseball diamond (BD). This wide set of categories includes a variety of items and buildings typically seen in aerial images, making the dataset extremely useful for training and assessing object identification algorithms. The DOTA-v1.5 dataset is described in full in Figure 1. Figure 1a depicts the frequency of the various item labels, while Figure 1b displays a correlogram of the labels. The frequency plot displays the distribution of object labels in the dataset, indicating how frequently every group appears throughout the photos. The correlogram, on the other hand, shows the associations between multiple labels, demonstrating linkages and combination patterns across different item types. The DOTA-v1.5 dataset is separated into subsets for model training and assessment, with 70% for training, 20% for validation, and 10% for testing [34]. This segmentation enables a thorough evaluation of object detection algorithms, guaranteeing that models are evaluated on previously unknown data and their performance is correctly measured at various phases of development.



(**a**) Frequency of labels (**b**) Correlogram of labels

Figure 1. Description of DOTA-v1.5 dataset.

3.2. Method

The existing research on tiny item recognition frequently skips a thorough examination of one-stage approaches, especially when it comes to critical performance variables like speed, computing time, and detection scores. Because of the absence of extensive assessments, it is difficult to identify and use the most effective strategies for detecting tiny objects. To solve these deficiencies, our research focuses on a detailed benchmarking analysis and the development of innovative pre-processing algorithms for the YOLOv8 model. The DOTA-v1.5 dataset, which is notable for its wide range of item categories and high-resolution photos, is used to assess the efficacy of one-step algorithms. This dataset provides a solid foundation for evaluating how well different algorithms perform under difficult settings, such as spotting tiny, densely packed objects. Using DOTA-v1.5, we want to give a complete comparison of existing one-stage approaches, highlighting their merits and limitations while taking into account both speed and accuracy.

Further, our work provides novel pre-processing strategies for YOLOv8 that improve its performance, particularly for tiny object recognition. Pre-processing is crucial for enhancing the quality of input data and, hence, the accuracy of detection models. Traditional pre-processing approaches may be insufficient to address the special issues of tiny object identification, resulting in an inferior performance. Our suggested solutions shown in Figure 2 include enhanced noise reduction and adaptive histogram equalization to improve picture contrast, allowing for the better separation of tiny objects. These pre-processing stages are combined with YOLOv8, which was chosen for its higher efficiency and accuracy than earlier one-stage models. YOLOv8's sophisticated design and training capabilities make it ideal for processing refined input data. The creation of models consists of three stages: data processing, model training, and assessment metrics. Data processing focuses on removing noise and reformatting pictures in order to increase input quality. YOLOv8 is then trained on these processed photos to determine its performance in spotting tiny things. The assessment step entails employing extensive metrics to assess not just the detection accuracy but also speed and computing economy. By concentrating on these characteristics, we want to give a more nuanced understanding of one-stage approaches and their practical consequences, ultimately leading to more effective and efficient tiny-item identification solutions.

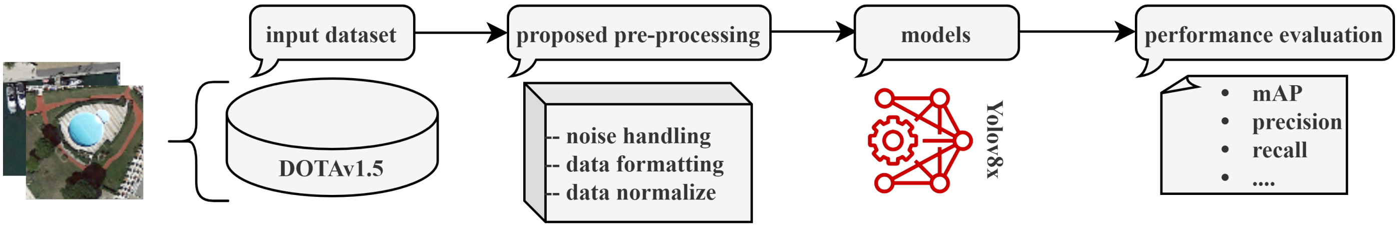


Figure 2. System overview of the proposed model.

3.2.1. The Proposed Pre-Processing Approach

The DOTA dataset uses an annotation style for every item indicated by an orientated bounding box (OBB). The coordinates of the i-th vertex of the oriented bounding box (OBB) are represented by (xi, yi), while the overall format includes (x1, y1, x2, y2, x3, y3, x4, y4, categories, complex). These vertices are ordered clockwise to establish the object's bounding box. This work describes a new pre-processing strategy for improving a DOTA-v1.5 dataset in order to train the YOLOv8 algorithm, which is critical for good object detection. The pre-processing approach consists of three critical steps: noise reduction, data presentation, and data standardization. Each of these actions is intended to improve the dataset's efficiency in the YOLOv8 model.

Noise handling: The first part of the pre-processing procedure handles the issue of noise in the dataset. The DOTA-v1.5 dataset includes two sorts of files: photos and their labels. The label files include not only the locations and class names of objects but also unnecessary text and information, which might inject noise into the dataset. To clean up the dataset, regular expressions are utilized to detect and delete any extraneous strings. This challenge uses two regular expressions, (1) and (2):

$$\hat{}\,.(imagesource).\backslash n? \tag{1}$$

$$\hat{}\,.(gsd).\backslash n? \tag{2}$$

After dealing with noise, the next stage is data formatting. During this step, the label file's last column, which provides further labeling information, is removed. Instead, a different strategy is used: each item is allocated a unique identification number using a dictionary. This dictionary converts object names, which are initially in string format, into numerical values. This transformation produces a new labeling column to replace the previous one. The new column, which contains numerical IDs, is subsequently added as the first column in the dataset. This update simplifies the dataset and guarantees that it meets the criteria of the YOLOv8 training procedure. By translating item names to numerical representations, the dataset becomes more effective and standardized, making the training process easier and the model more accurate.

Data normalization is the last stage of the pre-processing technique. This phase involves dividing each value in the label files by the height and width of the relevant picture, with the exception of the recently added labeling column. Through this process of normalization, the values are scaled to fall between 0 and 1. Normalization is used to minimize problems that could arise during training, like burst gradients. The model becomes less sensitive to changes in the input data and more resilient as a result of scaling the values. By ensuring that every input feature is on the same scale, this phase stops certain characteristics from controlling the learning process because of their higher values. Because normalization keeps the model from being too sensitive to specific characteristics, it promotes faster convergence and a more seamless training procedure.

The comparison between the original DOTA dataset and the dataset following the use of the suggested pre-processing strategy is shown in Table 3. The processed dataset demonstrates how the data formatting, normalization, and noise-management processes were applied successfully. The addition of a new labeling column including normalized values and numerical representations suggests that the dataset is now well-structured and ready for YOLOv8 training. Finally, by addressing noise, ensuring appropriate normalization, and improving data formatting, this thorough pre-processing method raises the overall standard of the DOTA-v1.5 dataset. Using the YOLOv8 algorithm for accurate and reliable object recognition is made possible by this improved dataset.

Table 3. A comparison of the dataset is presented before and after the application of the pre-processing approach. The ground sample distance (GSD), which represents the physical size of a single image pixel in meters, is also provided.

Explanation of Original Dataset
imagesource:GoogleEarth gsd:0.145268458746 846.0 569.0 541.0 775.0 854.0 567.0 752.0 874.0 plane 0 459.0 785.0 574.0 468.0 627.0 518.0 851.0 797.0 ship 0
Explanation of Processed Dataset
0 0.48... 0.24... 0.64... 0.87... 0.54... 0.61... 0.45... 0.78... 1 0.54... 0.78... 0.65... 0.12... 0.47... 0.57... 0.87... 0.87...

3.2.2. YOLOv8 Model

The YOLOv8 model is a cutting-edge object identification model that forecasts bounding boxes and class probabilities for every grid cell by dividing the input image into a grid. Localization loss, categorization loss, and confidence loss are combined to form the total loss function. Figure 3 [35] illustrates the structure of YOLOv8.

The model divides the input into N grid cells. For each cell i and corresponding bounding box j, it predicts four coordinates (x, y, w, h) that define the bounding box's location, along with a confidence score c. The class probabilities are encoded in the vector P. The predicted coordinates of the bounding box, $(\hat{x}_i^j, \hat{y}_i^j, \hat{w}_i^j, \hat{h}_i^j)$, are calculated according to the following equations:

$$\hat{x}_i^j = \sigma(b_{x,i}^j) + i \tag{3}$$

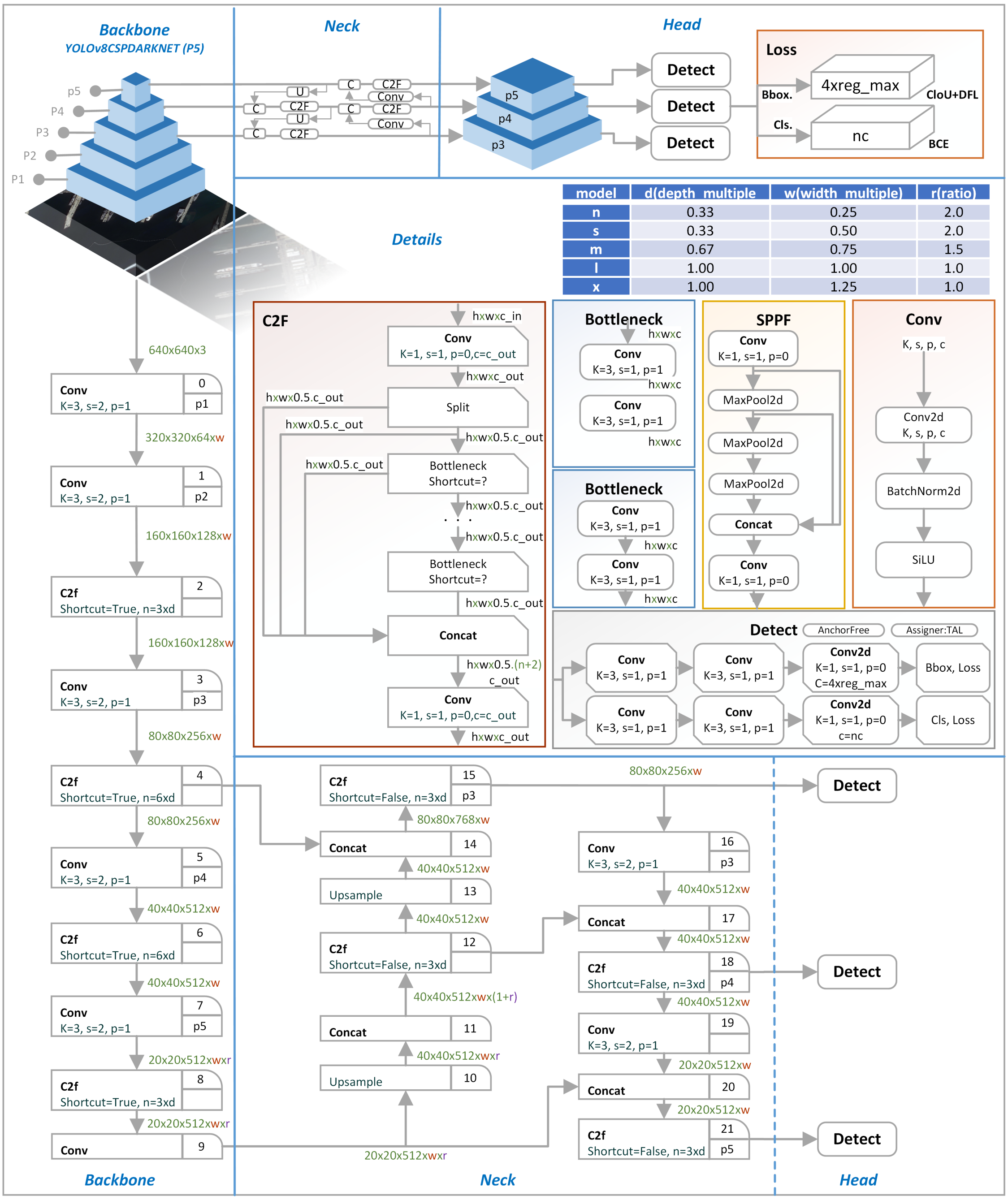
$$\hat{y}_i^j = \sigma(b_{y,i}^j) + i \tag{4}$$

$$\hat{w}_i^j = p_{w,i}^j \cdot e^{b_{w,i}^j} \tag{5}$$

$$\hat{h}_i^j = p_{h,i}^j \cdot e^{b_{h,i}^j} \tag{6}$$

Given the sigmoid function σ; the predicted parameters $b_{x,i}^j$, $b_{y,i}^j$, $b_{w,i}^j$, and $b_{h,i}^j$; as well as the dimensions of the anchor box $p_{w,i}^j$ and $p_{h,i}^j$, the confidence score c_i^j for each bounding box is defined as

$$c_i^j = \sigma(b_{c,i}^j) \tag{7}$$



model	d(depth_multiple	w(width_multiple)	r(ratio)
n	0.33	0.25	2.0
s	0.33	0.50	2.0
m	0.67	0.75	1.5
l	1.00	1.00	1.0
x	1.00	1.25	1.0

Figure 3. Internal architecture of YOLOv8.

The predicted confidence parameter is denoted by $b^j_{c,i}$. The class probabilities P_i are obtained by applying the softmax activation function:

$$P_i = \text{softmax}(\mathbf{b}_{\text{class},i}) \tag{8}$$

The term $\mathbf{b}_{\text{class},i}$ represents the vector of predicted class parameters. The total loss function is formulated as a linear combination of three components: the localization loss, confidence loss, and classification loss:

$$\mathcal{L} = \lambda_{\text{coord}} \sum_{i,j} (\text{Localization Loss}) + \tag{9}$$

$$\lambda_{\text{conf}} \sum_{i,j} (\text{Confidence Loss}) + \lambda_{\text{class}} \sum_{i} (\text{Classification Loss}) \tag{10}$$

The hyperparameters λ_{coord}, λ_{conf}, and λ_{class} are used to control the weighting of each individual loss component within the overall loss function.

4. Results Analysis

This section provides a performance analysis and discussion of the proposed pre-processing approaches. Table 4 illustrates the performance analysis based on the mean average precision (mAP) and average precision (AP). The evaluation metrics were the AP for each class. It is important to note that all the experiments were executed under the same conditions such as 50 epochs where 80% of the data are set for training and 20% of the data are set for testing.

Table 4. A performance evaluation and comparison of the proposed pre-processing techniques utilizing YOLOv8 is carried out by comparing the mean average precision (mAP) against leading one-stage small object detection models. In this context, TS denotes this study, which refers to the suggested approach. The categories considered in the evaluation include bridge (BR), helicopter (HC), storage tank (ST), soccer ball field (SBF), small vehicle (SV), plane (PL), large vehicle (LV), ground track field (GTF), tennis court (TC), ship (SH), swimming pool (SP), container crane (CC), basketball court (BC), harbor (HA), roundabout (RA), and baseball diamond (BD).

Study	PL	SH	ST	BD	TC	BC	GTF	HA	BR	LV	SV	HC	RA	SBF	SP	CC	mAP
[17]	87.9	65.8	71.4	85.2	90.8	87	73.7	69.3	49.7	77.8	42.8	51.5	62.9	61	65.5	1.4	65.2
[18]	88.2	73.4	68.4	81.4	90.7	82.8	66.5	67.1	43.2	68.5	41.9	47.3	62.5	47.4	75.6	1.9	62.9
[19]	86.4	74	70.6	82.5	87.4	83.4	71.4	65.8	49.6	66.2	43.3	49.8	57.9	67.2	67.3	2.1	64.1
[20]	80	73.9	70.3	85	90.8	81.3	80	73.4	53.2	75.5	44.5	52.9	64.7	64.1	71	1.8	66.4
[21]	79.5	68.2	67.5	74.8	81	65.5	54.7	60.2	45.7	44.8	33.2	49	61.5	39.8	68.2	1.3	55.9
[22]	88.8	74	71	83.2	90.8	81.2	80.2	74.7	51.2	78.6	44.7	50.7	60.1	65.3	67.5	3.6	66.6
[23]	86.6	73.8	72.1	81.4	89.9	80.9	77.4	70.1	56.3	78	44.8	48.6	62.4	60.9	74.2	3.4	66.3
[24]	80.2	70.2	69.9	83.8	90.8	80.1	76.2	71.3	48.3	77.5	39.4	50.6	59.6	56.7	70.2	4.8	64.4
[25]	88.4	71.6	69.8	85.7	90.8	84.7	72.3	67.8	45.53	68.9	43.6	49.9	62.8	59.8	71.6	3.7	64.8
TS	90	74.1	72.3	84	96.6	64	75.1	83	57.5	78.1	44.9	53.1	65.2	54.1	69.2	6.1	66.7

The proposed pre-processing approach outperforms other one-stage methods for the majority of the object classes in terms of the mAP with YOLOv8. This table presents a thorough performance comparative of the proposed pre-processing approach (denoted as "TS") against various state-of-the-art one-stage object detection algorithms utilizing YOLOv8, focusing on the mean average precision (mAP) across multiple object categories. The aim is to highlight the effectiveness of the proposed method in improving detection accuracy for a range of objects including planes, ships, storage tanks, baseball diamonds, tennis courts, basketball courts, ground track fields, harbors, bridges, large vehicles, small vehicles, helicopters, roundabouts, soccer ball fields, swimming pools, and container cranes.

The table provides mAP scores for each algorithm across these categories, reflecting how well each method performs in detecting and classifying objects. The mAP is a crucial metric in object detection, representing the average precision across all classes and thus giving a comprehensive measure of a model's performance. The comparison data consist of many studies, each with a reference number that indicates how well it detected distinct object types.

The study by [17] achieves an mAP score of 65.2, with its highest scores in detecting planes (87.9) and its lowest in container cranes (1.4). Similarly, [18] scores 62.9 overall, with its best performance in detecting planes (88.2) and a lower score for container cranes (1.9). The performance of [19] is noteworthy with an overall mAP of 64.1, excelling in detecting planes (86.4) but with less effectiveness in detecting container cranes (2.1). The authors of [20] present an mAP of 66.4, showing competitive results across most categories, particularly in detecting baseball diamonds (85) and tennis courts (90.8), although the score for container cranes is relatively low at 1.8. The study [21] demonstrates an overall mAP of 55.9, with strengths in detecting larger objects like storage tanks (67.5) but a weaker performance in detecting smaller objects like container cranes (1.3). The authors of [22] report an overall mAP of 66.6, highlighting its efficacy in detecting several object types, notably achieving high scores for tennis courts (90.8) and large vehicles (78.6), yet with a lower score for container cranes (3.6). The study by [23] shows an overall mAP of 66.3, with a good performance in detecting large vehicles (78) and tennis courts (89.9), but its detection of container cranes is also on the lower side (3.4). The study [24] achieves an mAP of 64.4, with a notable performance in detecting tennis courts (90.8) but a lower score for container cranes (4.8). Lastly, [25] reports an mAP of 64.8, excelling in detecting tennis courts (90.8) and planes (88.4) but with a relatively lower performance in container cranes (3.7).

The proposed method, TS, achieves an overall mAP score of 66.7, making it the top performer among the compared methods. The detailed breakdown reveals that TS excels particularly in detecting tennis courts (96.6) and planes (90), showing substantial improvements over other methods. It maintains a competitive performance across several categories, including storage tanks (72.3), baseball diamonds (84), and harbors (83), with varying effectiveness in detecting smaller objects like container cranes (6.1) and soccer ball fields (54.1). The exceptional performance of TS in several categories provides evidence that the pre-processing methods employed in this approach greatly improve the YOLOv8 model's capacity to reliably detect and classify objects. The proficiency of the suggested approach in attaining the best scores in specific categories, such as tennis courts and planes, highlights its efficacy in enhancing detection accuracy; this may be ascribed to the improved data representation and feature extraction procedures employed in TS.

The comparison demonstrates that the proposed pre-processing method (TS) not only achieves the highest overall mAP score but also exhibits significant improvements in specific categories where other methods demonstrate an inferior performance. For example, whereas TS has outstanding accuracy in identifying tennis courts and airplanes, properly detecting container cranes remains difficult, as shown by the lower score of 6.1. This underscores the possible domains in which additional improvements in the pre-processing practices might result in even better detection results. Furthermore, the table also demonstrates that while other methods exhibit robust performance in specific categories, they frequently fail to meet expectations in others. Methods such as [20,22] demonstrate improved mean average precision (mAP) scores in identifying tennis courts and large vehicles but exhibit a poor performance in detecting smaller items such as container cranes. In contrast, the proposed method demonstrates a more balanced performance across various categories, therefore highlighting its overall efficacy and adaptability.

In summary, the table compellingly illustrates the benefits of the newly proposed pre-processing approach (TS) in enhancing the efficacy of object detection using the YOLOv8 framework. The notable increase in the mean average precision (mAP) score and the method's exceptional ability to detect particular object classes indicate the effectiveness of the employed pre-processing approach. Such enhancements are primarily due to the improved data management and feature extraction techniques, leading to increased accuracy in detection. The results emphasize the potential of TS to enhance the state of the art in object detection, offering valuable insights for future research and progress in this field. Overall, the proposed approach is a substantial improvement in object detection technology, offering a reliable solution for precisely detecting and categorizing a diverse array of objects. The comprehensive comparison of the performance highlights the efficacy of the approach and establishes a standard for future enhancements in object detection systems.

The speed and time analyses of the suggested technique are presented in Table 5. None of the research that focuses on one-stage tiny object detection on the DOTA dataset has addressed their Giga Floating-point Operations Per Second (GFLOPs), speed, epoch, gradients, and other necessary evaluation parameters. Based on Table 5, it is clear that the proposed pre-processing methods are suitable for real-time applications. This paper provides a comprehensive comparison of various studies focusing on small object detection within the DOTA dataset, specifically emphasizing their GFLOPs (Giga Floating-point Operations Per Second), speed, epochs, gradients, and pre-processing, inference, loss, and postprocessing times. In this analysis, the goal is to highlight the efficiency and effectiveness of the proposed method relative to existing methods. Here is a detailed explanation of each aspect presented in the table.

Table 5. Time and speed analysis of the proposed method.

Studies	Gradients	GFLOPs	Epoch	Pre-Process	Inference	Loss	Postprocess
[17]	0.3	268.7	50	1.5 ms	59.7 ms	0.5 ms	7.2 ms
[18]	0.1	274.1	50	1.6 ms	60.2 ms	0.5 ms	9.3 ms
[19]	0.5	270.4	50	1.4 ms	69.1 ms	0.5 ms	8.4 ms
[20]	0.5	265.9	50	1.6 ms	75.3 ms	0.4 ms	7.5 ms
[21]	0.9	264.5	50	1.4 ms	71.4 ms	0.5 ms	7.1 ms
[22]	0.2	260.2	50	1.6 ms	63.9 ms	0.5 ms	6.9 ms
[23]	0.1	259.3	50	1.7 ms	67.7 ms	0.5 ms	6.8 ms
[24]	0.5	263.8	50	1.5 ms	64.5 ms	0.5 ms	7.5 ms
[25]	0.3	266.1	50	1.7 ms	60.1 ms	0.7 ms	8.2 ms
This study	0	263.2	50	1.2 ms	57.4 ms	0 ms	6.0 ms

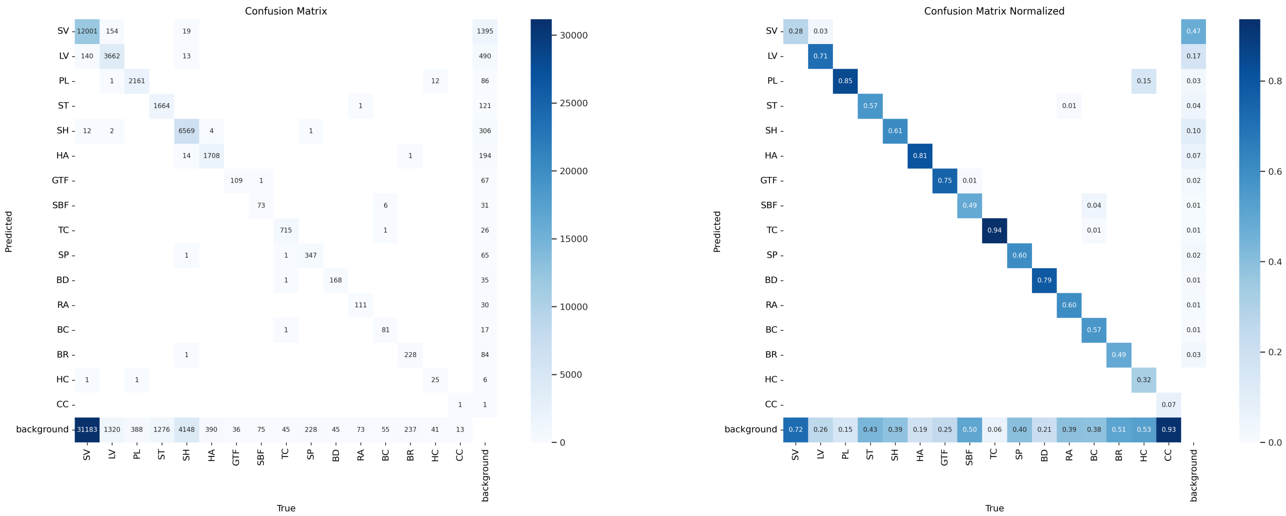
Table 5 summarizes key performance metrics for different studies and the proposed approach in the context of small object detection. The table includes columns for the following: Studies: references to the various studies evaluated. Gradients: the gradient computation time or amount, which reflects the amount of information used during the learning phase. GFLOPs: this indicates the computational complexity of the method, with lower values suggesting more efficient algorithms. Epoch: the number of times the learning algorithm iterates over the entire dataset. Pre-process: the time taken for data pre-processing. Inference: the time required to make predictions on new data. Loss: the time to compute the loss function. Postprocess: the time required for any additional processing after inference.

The study by [17] demonstrates a relatively balanced approach with gradients taking 0.3 ms, GFLOPs at 268.7, and a pre-processing time of 1.5 ms. The inference time is 59.7 ms, and postprocessing takes 7.2 ms. The loss calculation is quick at 0.5 ms. In [18], the gradients are slightly lower at 0.1 ms with GFLOPs of 274.1. This study shows a marginal increase in the pre-processing and inference times, but the postprocessing time is notably higher at 9.3 ms compared to other studies. The study by [19] shows the highest gradient computation time at 0.5 ms and GFLOPs of 270.4. The pre-processing time is the lowest among the studies (1.4 ms), but the inference time is the highest at 69.1 ms. The postprocessing time is 8.4 ms. In [20], with 0.5 ms for gradients and GFLOPs of 265.9, this study maintains a reasonable pre-processing time of 1.6 ms. The inference time is quite high at 75.3 ms, and postprocessing takes 7.5 ms. The study by [21] features a gradient computation time of 0.9 ms, GFLOPs of 264.5, and a pre-processing time of 1.4 ms. The inference time is 71.4 ms, and the postprocessing time is relatively low at 7.1 ms. In [22], gradients take 0.2 ms, the GFLOPs are 260.2, and the pre-processing time is 1.6 ms. The inference time is lower at 63.9 ms, with the postprocessing time at 6.9 ms. In [23], with the lowest gradient time of 0.1 ms and GFLOPs of 259.3, this study has a slightly higher pre-processing time of 1.7 ms. The inference time is 67.7 ms, and postprocessing is 6.8 ms. In [24], the gradient computation time is 0.5 ms, the GFLOPs are 263.8, and the pre-processing time is 1.5 ms. The inference time is 64.5 ms and postprocessing takes 7.5 ms. In [25], the gradient time is 0.3 ms, the GFLOPs are 266.1, and the pre-processing time is 1.7 ms. The inference time is 60.1 ms, with a postprocessing time of 8.2 ms.

The proposed method shows the following metrics: Gradients: 0 ms, indicating that gradient computation is either negligible or integrated differently, possibly through optimized methods or precomputed gradients. GFLOPs: 263.2, a value that is competitive with other methods, suggesting efficient computation. Epoch: 50, consistent with the other studies, providing a comparable basis for training duration. Pre-process: 1.2 ms, which is the lowest pre-processing time among all methods listed, highlighting efficient data handling and preparation. Inference: 57.4 ms, which is the lowest inference time, indicating faster prediction capabilities compared to other methods. Loss: 0 ms, suggesting that the loss calculation might be embedded within the training loop or otherwise optimized. Postprocess: 6.0 ms, the lowest postprocessing time, further emphasizing efficiency.

Comparative Analysis and Implications: Speed and efficiency: The proposed method demonstrates a superior efficiency in pre-processing, inference, and postprocessing times. Specifically, the proposed method's pre-processing time of 1.2 ms is notably faster than other studies, which range from 1.4 ms to 1.7 ms. Furthermore, the inference time of 57.4 ms is the lowest, suggesting faster object detection capabilities. Additionally, the postprocessing time of 6.0 ms is also the shortest, therefore enhancing the overall efficiency of the system. Computational complexity (GFLOPs): the proposed method's GFLOPs value of 263.2 is competitive and shows that while the method is computationally efficient, it does not sacrifice the complexity of the operations required for detection. Gradient computation and loss: The zero gradient time and zero loss time are particularly remarkable. These values suggest that the proposed method has redefined or optimized the typical gradient and loss computation processes, potentially integrating them into other stages of the pipeline or using advanced techniques that reduce their traditional computational overhead. Epochs: consistent with other studies, the proposed method uses 50 epochs, which provides a fair basis for comparison in terms of training duration. Conclusion: The data in Table 5 provide a clear illustration of the proposed method's efficiency and effectiveness in small object detection. The reduced pre-processing, inference, and postprocessing times compared to other studies underline its suitability for real-time applications. Moreover, the competitive GFLOPs value shows that this efficiency is achieved without compromising the computational complexity. This combination of low computational overhead and effective processing makes the proposed method highly advantageous for real-time and resource-constrained environments.

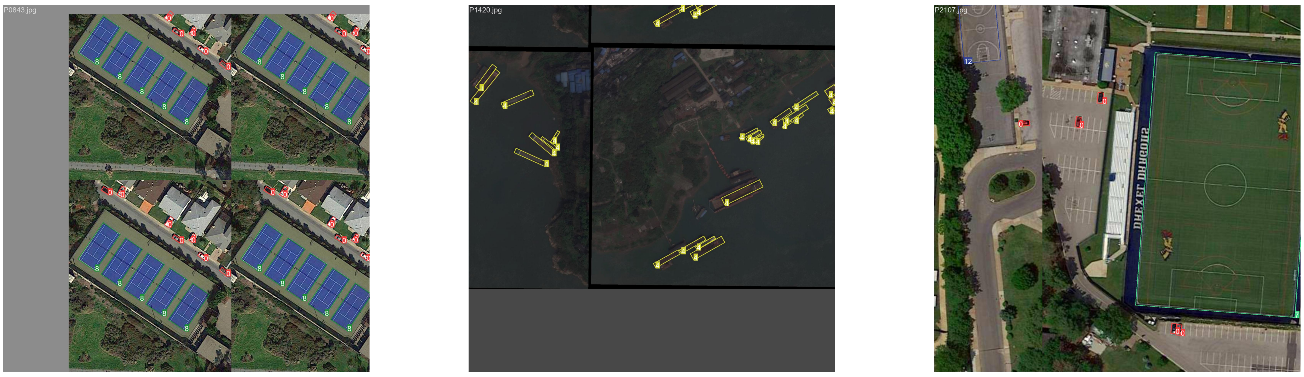
The proposed method's confusion matrices are presented in Figure 4. Figure 4a depicts the general confusion matrix, while Figure 4b illustrates the normalized version. Clearly, the normalized confusion matrix offers a more refined representation of the data compared to the general one. All training batches are illustrated in Figure 5, and the true and predicted validation images are displayed in Figure 6. It is evident that the proposed pre-processing approach with YOLOv8 has delivered exceptional results in terms of correctly identifying the true labels during predictions. Refer to Figure 7 for the confidence curves (the P curve, R curve, F1 curve, and PR curve) of the presented pre-processing approach using YOLOv8. The graph in Figure 7a demonstrates the balance between precision and confidence, two vital metrics in object detection. The precision of a model quantifies the proportion of accurate detections, which is determined by dividing the number of true positives by the total number of true positives and false positives. Confidence, on the other hand, reflects how certain the model is about the correctness of its detection, usually represented as a probability score between 0 and 1. The graph depicts how precision shifts as the confidence threshold is adjusted. The confidence threshold is the minimum value that the model's confidence score must meet for a detection to be considered valid. Raising the confidence threshold increases the model's accuracy, though it also reduces the total number of detections made. The various lines on the graph correspond to different object categories. For instance, the line marked "small-vehicle" illustrates the precision–confidence curve, indicating the model's performance in identifying small vehicles. The optimal area on a precision–confidence curve is the top-right side, where the approach illustrates both high confidence and high accuracy. Figure 7b provides a graphical depiction of the trade-off between two essential object detection metrics: recall and confidence. Recall measures the model's ability to locate all relevant instances of objects, reflecting the proportion of correctly identified objects in the images. In contrast, confidence shows how certain the model is about its predictions. The lines in the graph represent different object classes. For example, the "small-vehicle" line illustrates the model's recall at diverse confidence levels for tiny vehicles. The "all classes" curve represents the average recall across all categories. The figure's bottom-left value, "0.69 at 0.000," represents the model's recall when the confidence threshold is set to zero, indicating that the model correctly identifies approximately 69% of objects even with no confidence. Overall, the graph illustrates the model's detection performance across varying confidence levels.



(**a**) General confusion matrix

(**b**) Normalized confusion matrix

Figure 4. Confusion matrix of the proposed approach.

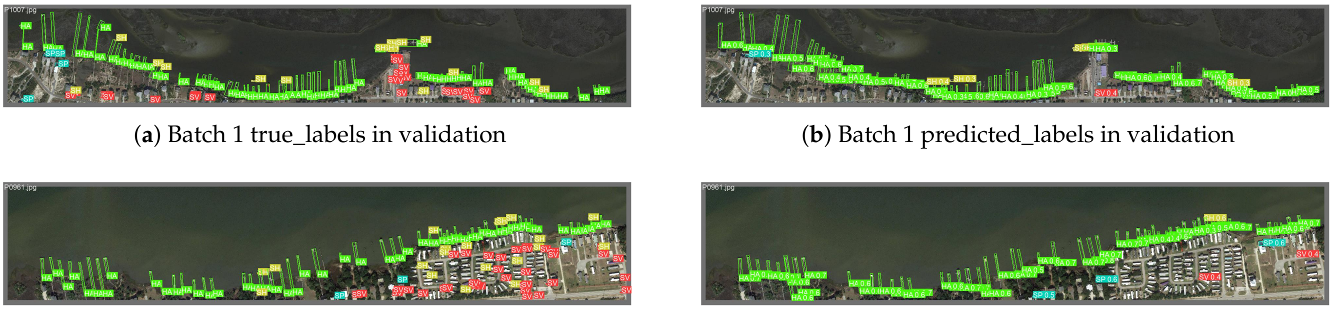


(**a**) Batch 1 in training

(**b**) Batch 2 in training

(**c**) Batch 3 in training

Figure 5. Images for three different batches.



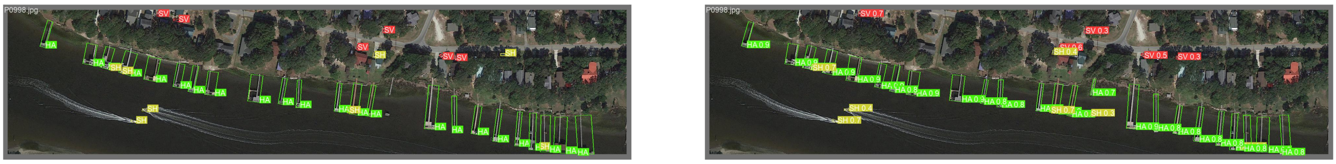
(**a**) Batch 1 true_labels in validation

(**b**) Batch 1 predicted_labels in validation

(**c**) Batch 2 true_labels in validation

(**d**) Batch 2 predicted_labels in validation

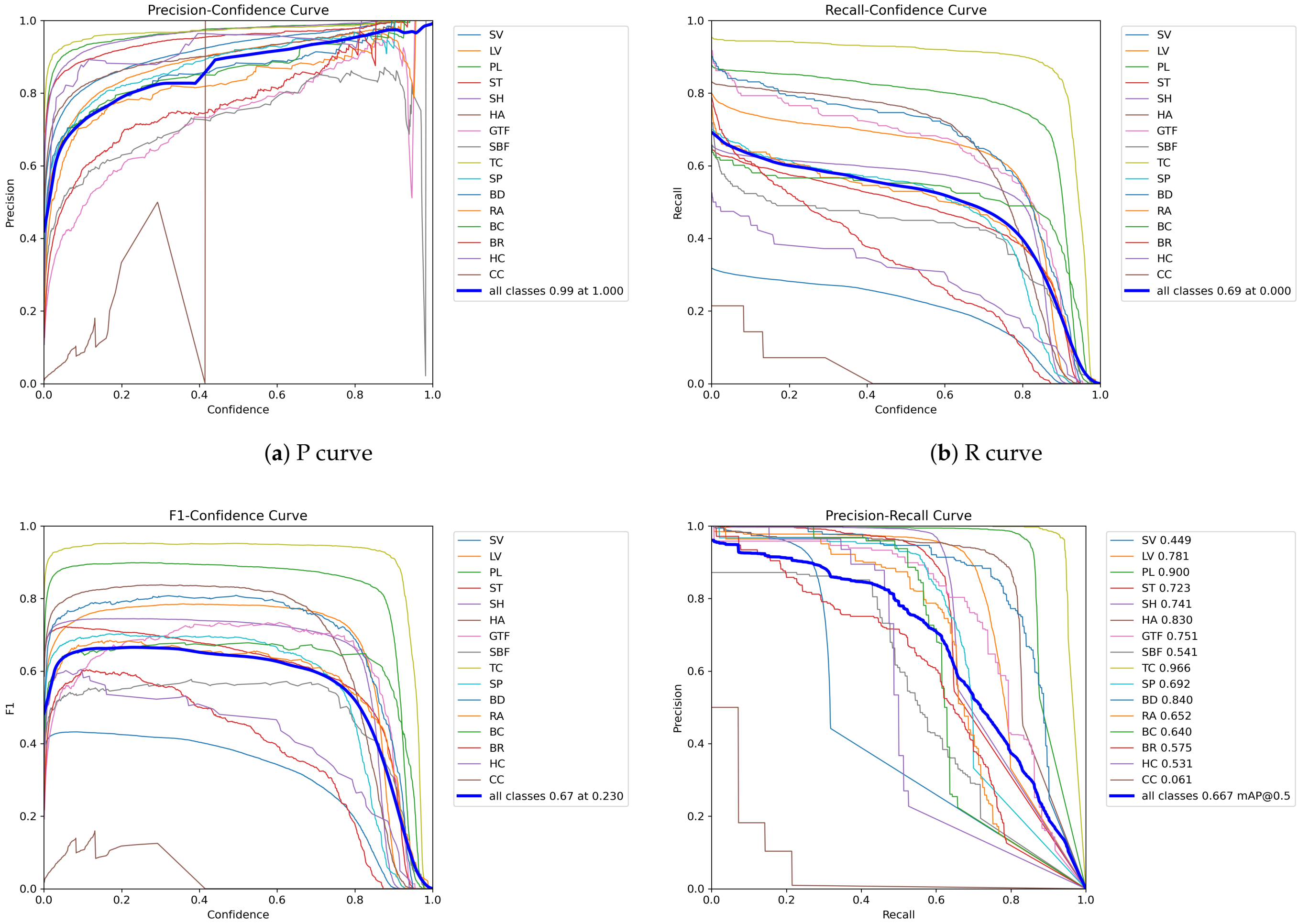
Figure 6. *Cont.*



(**e**) Batch 3 true_labels in validation

(**f**) Batch 3 predicted_labels in validation

Figure 6. Validation set true and predicted labels.



(**a**) P curve

(**b**) R curve

(**c**) F1 curve

(**d**) PR curve

Figure 7. Confidence curves.

Discussion

Research in the domain of detecting tiny objects in aerial imagery has consistently highlighted the challenges posed by the small size and low resolution of these objects. Previous studies, such as those by Yang et al. [17] and Wang et al. [18], primarily focused on enhancing detection accuracy through innovative architectures and loss functions, often neglecting critical aspects like data pre-processing and comprehensive evaluation metrics. Many existing works typically report the mean average precision (mAP) without considering the impacts of noise reduction and data normalization, which are essential for improving object visibility and discriminative features.

In contrast, our study introduces a robust data pre-processing technique for YOLOv8, which includes noise reduction, data restructuring, and normalization. These strategies significantly enhance the clarity of small object boundaries and improve detection accuracy across all classes in the DOTA-v1.5 dataset. By utilizing 50 epochs for training and encompassing all relevant categories for small object detection, our approach not only achieves a higher mAP but also establishes a new standard for processing speed and efficiency. Furthermore, our comprehensive evaluation—encompassing confusion matrices and performance metrics—addresses gaps left by prior studies [19,20,26,27], underscoring the importance of rigorous evaluation and thorough analysis in advancing the field.

Future research directions should focus on enhancing the model's capabilities for detecting small objects. This could involve integrating multi-scale detection techniques to identify small objects at various resolutions or exploring advanced architectures, such as YOLOv10, that may offer improved feature-extraction capabilities. Additionally, experimenting with hybrid approaches that combine traditional object detection methods with deep learning techniques could yield beneficial results. Although our benchmarking analysis provides valuable insights, there remains significant potential for improving model performance in detecting small objects, particularly by addressing the limitations of current pre-processing techniques.

Moreover, future research should explore the implementation of more robust data-augmentation strategies that simulate diverse real-world scenarios, enhancing model robustness against varying conditions. While our evaluation metrics offer a clearer understanding of model effectiveness, ongoing development and refinement are necessary to advance small object detection in aerial imagery. By acknowledging these limitations and pursuing these research directions, we aim to contribute to the ongoing advancement of accurate and efficient techniques for small object detection, ultimately improving the applicability and reliability of such models in practical applications.

5. Conclusions

Our study provides a benchmarking of one-stage tiny object recognition algorithms on the DOTA dataset with YOLOv8. We also propose a novel reprocessing approach that significantly improves tiny object representation by managing noise, formatting data, and normalizing it, resulting in an increased mean average precision and setting new performance and efficiency benchmarks for real-time applications. Our study emphasizes the importance of evaluating parameters such as speed, parameters, GLPFS, epochs, and gradients, often overlooked in prior research. The confusion matrices and training/validation batches underscore the model's effectiveness, with all the experimental results collected under consistent conditions. Furthermore, the suggested approach enhances comprehension of the current state of the art by evaluating not just the mean average precision but also the speed and computational efficiency. This thorough assessment facilitates the selection of the most suitable method for practical deployment, taking into account all pertinent factors.

This study's limitation lies in the exclusive use of YOLOv8 for our proposed pre-processing methods. Additionally, it does not address the potential to enhance the training speed and reduce the training time of YOLOv8. Future research should explore novel pre-processing methods for object detection that could substantially improve the training speed and reduce training time across various models.

Author Contributions: Conceptualization was carried out by M.A.M.A.-H.; methodology by M.A.M.A.-H.; software development by M.A.M.A.-H., A.H., and F.T.; validation by M.A.M.A.-H.; formal analysis by M.A.M.A.-H.; investigation by M.A.M.A.-H.; resources were provided by M.A.M.A.-H.; data curation by M.A.M.A.-H.; writing—original draft preparation by M.A.M.A.-H.; writing—review and editing by Y.L.; visualization by M.A.M.A.-H.; supervision by Y.L.; project administration by M.A.M.A.-H.; and funding acquisition by Y.L. All authors have reviewed and approved the final version of the manuscript for publication.

Funding: This research was partially funded by the National Natural Science Foundation of China under grant #62176247. Additional support was provided by the Fundamental Research Funds for the Central Universities.

Data Availability Statement: The DOTA dataset is available at https://captain-whu.github.io/DOTA/dataset.html (accessed on 9 October 2024).

Conflicts of Interest: Author Ahsan Habib was employed by the company Technovative Solutions LTD. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Han, W.; Chen, J.; Wang, L.; Feng, R.; Li, F.; Wu, L.; Yan, J. Methods for small, weak object detection in optical high-resolution remote sensing images: A survey of advances and challenges. *IEEE Geosci. Remote Sens. Mag*. **2021**, *9*, 8–34. [CrossRef]
2. Shivappriya, S.N.; Priyadarsini, M.J.P.; Stateczny, A.; Puttamadappa, C.; Parameshachari, B.D. Cascade object detection and remote sensing object detection method based on trainable activation function. *Remote Sens.* **2021**, *13*, 200. [CrossRef]
3. Zhang, M.; Xu, S.; Song, W.; He, Q.; Wei, Q. Lightweight underwater object detection based on yolo v4 and multi-scale attentional feature fusion. *Remote Sens.* **2021**, *13*, 4706. [CrossRef]
4. Chen, L.; Liu, C.; Chang, F.; Li, S.; Nie, Z. Adaptive multi-level feature fusion and attention-based network for arbitrary-oriented object detection in remote sensing imagery. *Neurocomputing* **2021**, *451*, 67–80. [CrossRef]
5. Rabbi, J.; Ray, N.; Schubert, M.; Chowdhury, S.; Chao, D. Small-object detection in remote sensing images with end-to-end edge-enhanced GAN and object detector network. *Remote Sens.* **2020**, *12*, 1432. [CrossRef]
6. Mishra, B.; Garg, D.; Narang, P.; Mishra, V. Drone-surveillance for search and rescue in natural disaster. *Comput. Commun.* **2020**, *156*, 1–10. [CrossRef]
7. Adarsh, P.; Rathi, P.; Kumar, M. YOLO v3-Tiny: Object Detection and Recognition using one stage improved model. In Proceedings of the 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 6–7 March 2020; pp. 687–694.
8. Jiang, P.; Ergu, D.; Liu, F.; Cai, Y.; Ma, B. A Review of Yolo algorithm developments. *Procedia Comput. Sci.* **2022**, *199*, 1066–1073. [CrossRef]
9. Pan, H.; Jiang, J.; Chen, G. TDFSSD: Top-down feature fusion single shot MultiBox detector. *Signal Process. Image Commun.* **2020**, *89*, 115987. [CrossRef]
10. Kattenborn, T.; Leitloff, J.; Schiefer, F.; Hinz, S. Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. *ISPRS J. Photogramm. Remote Sens*. **2021**, *173*, 24–49. [CrossRef]
11. Du, L.; Zhang, R.; Wang, X. Overview of two-stage object detection algorithms. *J. Phys. Conf. Ser.* **2020**, *1544*, 012033. [CrossRef]
12. Sultana, F.; Sufian, A.; Dutta, P. A review of object detection models based on convolutional neural network. *Intell. Comput. Image Process. Based Appl.* **2020**, 1–16.
13. Yi, J.; Wu, P.; Liu, B.; Huang, Q.; Qu, H.; Metaxas, D. Oriented object detection in aerial images with box boundary-aware vectors. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Virtual, 5–9 January 2021; pp. 2150–2159.
14. Xie, X.; Cheng, G.; Wang, J.; Yao, X.; Han, J. Oriented R-CNN for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 3520–3529.
15. Carranza-García, M.; Torres-Mateo, J.; Lara-Benítez, P.; García-Gutiérrez, J. On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data. *Remote Sens.* **2020**, *13*, 89. [CrossRef]
16. Cai, Y.; Luan, T.; Gao, H.; Wang, H.; Chen, L.; Li, Y.; Li, Z. YOLOv4-5D: An effective and efficient object detector for autonomous driving. *IEEE Trans. Instrum. Meas*. **2021**, *70*, 1–13. [CrossRef]
17. Yang, X.; Yan, J.; Liao, W.; Yang, X.; Tang, J.; He, T. Scrdet++: Detecting small, cluttered and rotated objects via instance-level feature denoising and rotation loss smoothing. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 2384–2399. [CrossRef]
18. Wang, P.; Sun, X.; Diao, W.; Fu, K. FMSSD: Feature-merged single-shot detection for multiscale objects in large-scale remote sensing imagery. *IEEE Trans. Geosci. Remote Sens.* 2019, *58*, 3377–3390. [CrossRef]
19. Qian, W.; Yang, X.; Peng, S.; Zhang, X.; Yan, J. RSDet++: Point-based modulated loss for more accurate rotated object detection. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 7869–7879. [CrossRef]
20. Zakaria, Y.; Mokhtar, S.A.; Baraka, H.; Hadhoud, M. Improving Small and Cluttered Object Detection by Incorporating Instance Level Denoising Into Single-Shot Alignment Network for Remote Sensing Imagery. *IEEE Access* **2022**, *10*, 51176–51190. [CrossRef]
21. Li, Y.; Pei, X.; Huang, Q.; Jiao, L.; Shang, R.; Marturi, N. Anchor-free single stage detector in remote sensing images based on multiscale dense path aggregation feature pyramid network. *IEEE Access* **2020**, *8*, 63121–63133. [CrossRef]
22. Qian, X.; Wu, B.; Cheng, G.; Yao, X.; Wang, W.; Han, J. Building a bridge of bounding box regression between oriented and horizontal object detection in remote sensing images. *IEEE Trans. Geosci. Remote Sens*. **2023**, *61*, 1–9.
23. Hou, L.; Lu, K.; Yang, X.; Li, Y.; Xue, J. G-rep: Gaussian representation for arbitrary-oriented object detection. *Remote Sens.* **2023**, *15*, 757. [CrossRef]
24. Lin, P.; Wu, X.; Wang, B. Oriented Object Detection Based on Foreground Feature Enhancement in Remote Sensing Images. *Remote Sens.* **2022**, *14*, 6226. [CrossRef]
25. Cao, D.; Zhu, C.; Hu, X.; Zhou, R. Semantic-Edge-Supervised Single-Stage Detector for Oriented Object Detection in Remote Sensing Imagery. *Remote Sens.* **2022**, *14*, 3637. [CrossRef]
26. Jiang, Q.; Dai, J.; Rui, T.; Shao, F.; Lu, G.; Wang, J. Context-Based Oriented Object Detector for Small Objects in Remote Sensing Imagery. *IEEE Access* **2022**, *10*, 100526–100539. [CrossRef]

27. Cheng, G.; Wang, J.; Li, K.; Xie, X.; Lang, C.; Yao, Y.; Han, J. Anchor-free oriented proposal generator for object detection. *IEEE Trans. Geosci. Remote Sens*. **2022**, *60*, 1–11. [CrossRef]
28. Chen, W.; Miao, S.; Wang, G.; Cheng, G. Recalibrating Features and Regression for Oriented Object Detection. *Remote Sens*. **2023**, *15*, 2134. [CrossRef]
29. Gao, L.; Gao, H.; Wang, Y.; Liu, D.; Momanyi, B.M. Center-Ness and Repulsion: Constraints to Improve Remote Sensing Object Detection via RepPoints. *Remote Sens.* **2023**, *15*, 1479. [CrossRef]
30. Wei, C.; Ni, W.; Qin, Y.; Wu, J.; Zhang, H.; Liu, Q.; Bian, H. RiDOP: A Rotation-Invariant Detector with Simple Oriented Proposals in Remote Sensing Images. *Remote Sens.* **2023**, *15*, 594. [CrossRef]
31. Zheng, H.; Ji, M.; Wang, H.; Liu, Y.; Fang, L. Crossnet: An end-to-end reference-based super resolution network using cross-scale warping. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 88–104.
32. Law, H.; Deng, J. Cornernet: Detecting objects as paired keypoints. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 734–750.
33. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. Centernet: Keypoint triplets for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6569–6578.
34. DOTA-v1.5. DOTA Dataset. Available online: https://captain-whu.github.io/DOTA/ (accessed on 5 January 2023).
35. Hussain, M. YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection. *Machines* **2023**, *11*, 677. [CrossRef]

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.