applied sciences

MDPI

Article

Exploiting the Potential of Overlapping Cropping for Real-World Pedestrian and Vehicle Detection with Gigapixel-Level Images

Chunlei Wang [1], Wenquan Feng [1], Binghao Liu [1], Xinyang Ling [1] and Yifan Yang [2,*]

[1] Department of Electrics and Information Engineering, Beihang University, Beijing 100191, China
[2] Institute of Artificial Intelligence, Beihang University, Beijing 100191, China
* Correspondence: stephenyoung@buaa.edu.cn

Abstract: Pedestrian and vehicle detection is widely used in intelligent assisted driving, pedestrian counting, drone aerial photography, and other applications. Recently, with the development of gigacameras, gigapixel-level images have emerged. The large field of view and high resolution provide global and local information, which enables object detection in real-world scenarios. Although existing pedestrian and vehicle detection algorithms have achieved remarkable success for standard images, their methods are not suitable for ultra-high-resolution images. In order to improve the performance of existing pedestrian and vehicle detectors in real-world scenarios, we used a sliding window to crop the original images to solve this problem. When fusing the sub-images, we proposed a midline method to reduce the cropped objects that NMS could not eliminate. At the same time, we used varifocal loss to solve the imbalance between positive and negative samples caused by the high resolution. We also found that pedestrians and vehicles were separable in size and comprised more than one target type. As a result, we improved the detector performance with single-class object detection for pedestrians and vehicles, respectively. At the same time, we provided many useful strategies to improve the detector. The experimental results demonstrated that our method could improve the performance of real-world pedestrian and vehicle detection.

Keywords: gigapixel-level image; super-high resolution; pedestrian detection; vehicle detection

check for updates

Citation: Wang, C.; Feng, W.; Liu, B.; Ling, X.; Yang, Y. Exploiting the Potential of Overlapping Cropping for Real-World Pedestrian and Vehicle Detection with Gigapixel-Level Images. *Appl. Sci.* **2023**, *13*, 3637. https://doi.org/10.3390/app13063637

Academic Editor: Andrea Prati

Received: 23 February 2023
Revised: 6 March 2023
Accepted: 9 March 2023
Published: 13 March 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the rapid development of object detection, its application has penetrated various fields of our daily lives. The primary purpose of object detection is to distinguish the categories of objects and locate their positions. Pedestrian and vehicle detection is an essential branch of object detection. In real life, pedestrians and vehicles introduce problems such as severe occlusion, large-scale variation, and multiple poses. These problems lead to poor performances by detectors.

In recent years, object detection algorithms based on deep learning have achieved high accuracy using classic datasets such as MS COCO [1] and Pascal VOC [2]. However, in these two datasets, the resolution of pictures is 640 px, and each picture contains only a few objects, which affects the real-world application of target detection. Although there are some high-resolution remote-sensing image datasets, such as VisDrone [3] and UAVDT [4], the resolution of pictures in these datasets is about 2000 px, which does not match real-world scenarios. Recently, the gigapixel dataset PANDA [5] (25,000 × 14,000 pixels) proposed by Tsing University has gradually become the key to studying real-world object detection. Thanks to the development of array camera technology, PANDA contains gigapixel-level images, and the field of vision covers a reasonably wide range. Gigapixel-level surveillance videos recently became necessary materials to analyze group behavior in public space [6].

In recent years, object detection has gradually changed from relying on manual feature extraction methods to deep learning networks. Whether using the traditional datasets MS

COCO and Pascal VOC or the remote-sensing datasets VisDrone and UAVDT, the performance of object detectors is improving daily. However, gigapixel-level object detection still needs further research. The following sub-sections discuss the existing algorithms for traditional and gigapixel-level object detection.

1.1. Object Detection

As we all know, object detection is based on image classification. After AlexNet [7] proposed a deep convolutional neural network for image classification to achieve a breakthrough in performance, many CNN-based methods of object detection began to emerge. SPPNet [8] proposed spatial pyramid pooling, so that the size of the image no longer limited the input of the CNN, which also improved the recognition performance for multi-scale targets. Due to the long training time and large memory consumption of R-CNNs, a fast R-CNN [9] was proposed. Faster R-CNN [10] incorporated region proposal networks(RPNs) by sharing convolutional layers with SOTA object detection networks and improved the training speed of the network. However, the CNN-based deep model could not identify the negative samples that the AI model had difficulty distinguishing at that time. In order to solve this problem, online hard example mining (OHEM) [11] and focal loss [12] were proposed. By sorting input RoIs, OHEM could select hard examples from low-loss values. Focal loss solved the problem of sample imbalance by tilting the loss function to make the model focus on hard samples, which are difficult to classify. The previous object detection network was based on a single high-level feature, which made it challenging to solve multi-scale variant problems, especially with small targets. The use of image pyramids led to a surge in calculations. Through a unique feature pyramid network, [13] avoided a high number of calculations and solved multi-scale problems. In order to further enhance the feature extraction ability of the model, Zhu [14] improved the FCN by fusing local features with global features and proposed CoupleNet.

The previous object detection algorithms obtained the final results by implementing NMS through Intersection over Union (IOU). Soft NMS [15] receives the IOU of two candidates through a Gaussian function, enacts different degrees of punishment, and modifies the target's confidence in combination with the strength of the punishment. Cascade R-CNN [16] was an essential improvement in object detection. Through multiple cascaded detectors and different IOU thresholds, Cascade R-CNN improved the mAP by two to four points in almost all R-CNN-based networks. Wu [17] explored the role of a fully connected head and convolution head in object detection in detail and proposed the double-head method, which involved a fully connected head focusing on classification and a convolution head for bounding box regression. EfficientDet [18] proposed a weighted bi-directional feature pyramid network (BiFPN) and a compound scaling method to enhance the feature extraction capability of models. OTA [19] proposed a label allocation method based on an optimization strategy, which allowed the network to choose the number of anchors corresponding to each ground truth, thereby avoiding the anchor allocation problem caused by setting in advance. Li [20] proposed a new dual weighting paradigm, which could achieve a better loss by giving different weights to positive and negative samples.

Since Joseph Redmon proposed YOLOv1 [21], single-stage object detection models have gradually become a research hotspot. The main difference between single-stage and two-stage detectors is that the single-stage model only needs to input images into the network to obtain the end-to-end result. However, the two-stage model needs to propose a candidate region and then perform object detection, which greatly affects the running speed of the detector. In the beginning, the detection accuracy of the single-stage model was low, but with the deepening of research, a large number of excellent works have emerged: for example, the YOLO series [21–24], SSD [25], and RefineDet [26] . With the development of object detection, the accuracy of the single-stage model has become not inferior to that of the two-stage model while ensuring a high speed, especially YOLOv5 [27], which has been widely used due to its high detection speed and accuracy.

Compared with YOLO algorithms, SDD directly uses CNN for detection. There are two significant changes in the SDD algorithm. First, it uses convolutional feature maps of different scales for detection. Second, it uses anchor box priors of different scales and aspect ratios. SDD solves the problems of the YOLO algorithms, i.e., the complexity of determining small targets and inaccurate positioning. RefineDet introduced the idea of regression from coarse to fine in a two-stage object detector to improve the regression accuracy of SSD. At the same time, it introduced feature fusion to further improve the detection ability for small targets.

1.2. High-Resolution Image Object Detection

As the resolution of cameras increases, traditional object detection datasets such as MS COCO (640 × 640 pixels) no longer meet the needs of researchers. Several works already exist regarding high-resolution aerial and remote-sensing images, such as the VisDrone [3] and UAVDT [4] datasets. Ref. [28] cut high-resolution images into small cropped images by uniform cropping and random cropping for object detection to improve small object detection accuracy. Ref. [29] referred to the crowd counting network density map to generate the density distribution of the high-resolution image and then cropped the image according to the density distribution. Compared with uniform cropping, this method was beneficial in reducing the influence of the background. Ref. [30] proposed a learnable downsampling module to enable the network to focus more on the target's surroundings when downsampling the input images, thereby achieving the segmentation of high-resolution images.

Since the creation of PANDA, large-scale, long-term, multi-object visual analysis tasks have gradually become a research hotspot. PANDA provides enriched and hierarchical ground truth annotations, including 15,974.6 k bounding boxes, 111.8k fine-grained attribute labels, 12.7 k trajectories, 2.2 k groups, and 2.9 k interactions [5].

The image-cropping method is used chiefly in high-resolution images. However, for gigapixel-level images, this method suffers from a considerable time cost, and the cropped image contains a large number of background and cropped objects. Ref. [31] proposed a novel patch arrangement framework. First, the authors obtained the image's cropping area according to the target's distribution and placed the cropped images on a uniform canvas. Finally, object detection was performed on these canvases.

Ref. [32] proposed a region NMS algorithm, a fusion strategy involving the mapping of a small image onto a large image to eliminate the cropped image, resulting in redundant fragmentation boxes. The authors used a sliding window to crop all original images and obtained pre-detection results. Then, they used the labels of the images to crop the objects. However, this method inevitably fragmented the objects, resulting in a decrease in target detection performance.

Chen [33] proposed GigaDet, a novel gigapixel-level framework. Based on the spatial sparsity of objects in the PANDA dataset, GigaDet proposed a patch generation network to globally locate areas that may contain objects and determine the appropriate adjustment ratio for each patch. The collected multi-scale plaques were then input into the detector in parallel for accurate and fast detection in a local manner. This idea was similar to image cropping but contained more objects.

In order to overcome the significant differences in target pose, scale, and occlusion in the PANDA dataset, Mo [34] proposed PVNet. Firstly, the authors designed a deformable deep residual network as the backbone to enhance the effective perceptual field. Secondly, they adopted a path aggregation feature pyramid network to process the multi-scale features. Finally, the DyHead module was introduced to enhance the scale, space, and task awareness of the detection head and further optimize the classification and positioning of pedestrians and vehicles.

Object scale differences depend on spatial location. Furthermore, the problem of harrowing feature extraction is caused by overlapping targets. SARNet [35] used spatial

attention to enhance the representation ability and dilated the feature pyramid network to improve the detection accuracy for different scales.

This paper used a two-stage object detection method for the PANDA dataset. First, we cropped the original gigapixel-level images with a sliding window. Second, we input the cropped image into the YOLOv5 detector. The cropping method inevitably generated cropped targets that non-maximum suppression (NMS) could not remove. We proposed a midline method to alleviate the performance decrease caused by this situation. The experiments proved that our method could improve the mAP and F1 score.

The main research content of this paper is as follows: In the second section, we introduce the related works on object detection and gigapixel-level object detection algorithms. In the third section, we describe our method and techniques in detail. In the fourth section, we design rigorous experiments to prove the effectiveness of the proposed method and ablation study to verify the impact of each technique on the performance. The fifth section presents a summary.

The main contributions are summarized as follows:

- We used different cropping sizes to train pedestrian and vehicle models separately, which could improve the precision and recall of within-class objects.
- We proposed an effective post-processing method that could eliminate the impact of cropped objects and improve the ability of the detector.
- We implemented effective strategies and extensive experiments demonstrating the effectiveness of our method for the PANDA dataset.

2. Materials and Methods

This study used YOLOv5 as the baseline for pedestrian and vehicle detection in the gigapixel-level dataset PANDA. We mainly used a cropping-based method to process gigapixel-level images. This section introduces the scientific problems related to gigapixel-level images and our solutions, including the network, the proposed midline post-processing method, and the collection of techniques.

2.1. Scientific Problems for Gigapixel Images

This paper is based on the gigadetection track of the gigavision competition held by Tsinghua University in 2022. The gigadetection track included object detection for pedestrians and vehicles. The training set contained 390 pictures in 13 scenes, and the competition test set contained 165 pictures in 5 scenes; each picture had a resolution of gigapixels and contained a large number of multi-scale objects. Pedestrians and vehicles had different representations and categories. Pedestrians had different postures ('standing', 'walking', 'sitting', or 'riding') and different ages ('adult' or 'child'). Vehicles had different sizes ('small car', 'midsize car', or 'large car') and categories ('bicycle', 'motorcycle', 'tricycle', 'electric car', or 'baby carriage'). The main challenges of this competition were that: (1) The size of the images was much larger than any previous dataset. For the PANDA dataset, the mAP of our method was 61.4 %. In the 31 December 2022 https://www.gigavision.cn/ competition, our method was placed 10th. As shown in Figure 1a, the width and height of the images in the VisDrone dataset are only 1920 pixels and 1080 pixels, respectively, which is one-tenth the size of the PANDA dataset images. This means that PANDA has an image area hundreds of times larger than VisDrone. Existing detectors cannot directly perform object detection on such images. (2) There was a large background area in the image. As shown in Figure 1b, almost two-thirds of the area had no targets. This meant that an unbalanced number of positive and negative samples would lead to performance damage when we performed object detection on this image. (3) Figure 1c depicts the large difference in the target sizes of pedestrians and vehicles, which meant that if the cropping size was too small, the vehicles would be easily cut into pieces, and if it was too large, the pedestrian targets would be too small to detect. Due to these challenges, we made a series of improvements, and the experiments and competition results proved that our improvements were effective.

One of the main difficulties in PANDA is that the resolution is too high to use current state-of-the-art object detectors directly. The competition sponsor provided three baselines to detect gigapixel-level images directly. RetinaNet achieved a 0.225 mAP and a 0.272 F1 score, Faster R-CNN achieved a 0.284 mAP and a 0.331 F1 score, and Cascade R-CNN achieved a 0.309 mAP and a 0.352 F1 score. Currently, several works have been published on object detection in gigapixel-level images. Due to the limitations of GPU memory, the current methods cannot use the whole image as the input of the deep learning model. They usually employ a sliding window strategy to crop and resize the image. After obtaining the detection results of each sub-image, they combine them to obtain the final detection result. The authors of [5] used Faster R-CNN with the above cropping scheme and obtained a 75.5% AP.50 for the visible part of the large object. However, in the same setting, the AP.50 for small objects was only 20.1%, which was unsatisfactory. It is also a challenge to select an appropriate cropping window size. A large cropping window leads to the inaccurate detection of small objects. Similarly, a small cropping window causes large objects to be cropped, affecting performance.

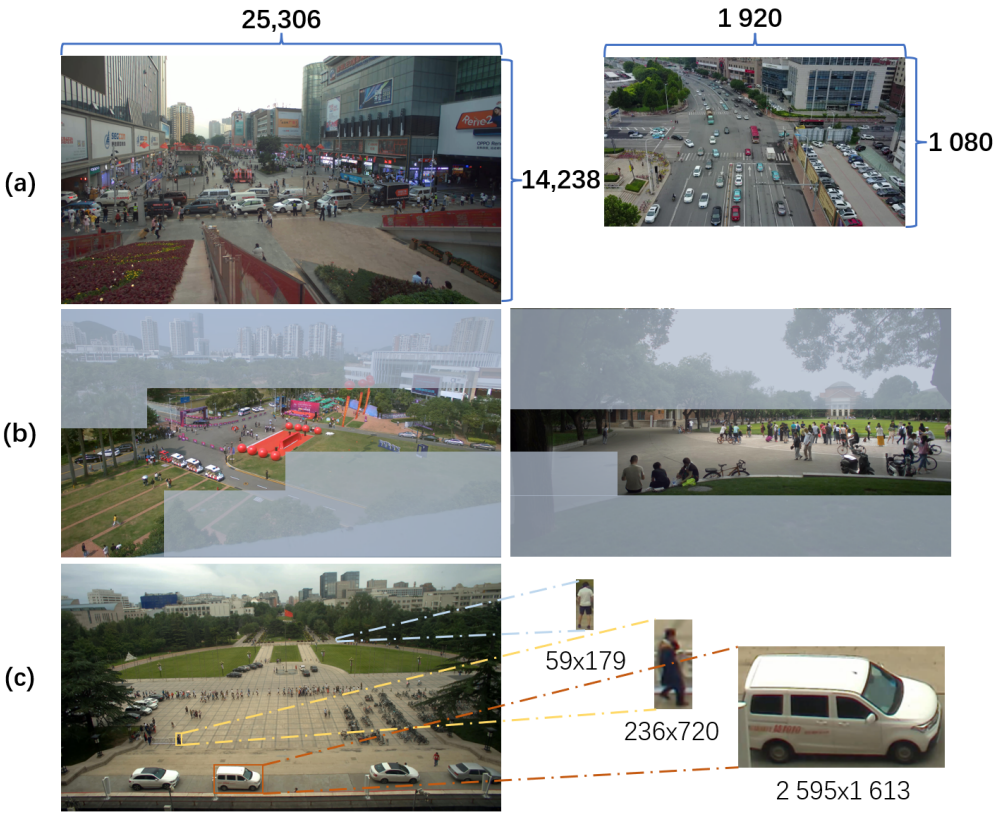

Figure 1. Challenges in object detection using the PANDA dataset. (**a**) The image size of the PANDA dataset (left) is much larger than that of VisDrone (right). (**b**) Almost two-thirds of the image contains no targets (gray area). (**c**) Pedestrians and vehicles are of different sizes.

This study explored the performance of existing object detection methods in a gigapixel-level dataset. We used the single-stage detector YOLOv5 as a baseline. Due to the higher accuracy for small targets, TPH-YOLOv5 [36] performed better than YOLOv5 for the remote-sensing dataset UAVDT. However, compared to YOLOv5, TPH-YOLOv5 adds additional computational overhead. Although there are already better YOLO algorithms, such as YOLOv7 [37], considering the calculation resources and accuracy trade-off, we still used YOLOv5 as a baseline. We proposed a midline method to reduce the performance penalties for cropped targets. Based on the width and height distribution of objects in the PANDA dataset, we explored the effect of different crop sizes on the results and selected the optimal sliding window size and overlap.

After determining the baseline, we tried several SOTA methods and improvement strategies in classic object detection datasets for scientific problems in gigadetection. We used varifocal loss to solve the problem of sample imbalance. When the number of negative samples is higher than that of the positive samples, the model's training is biased towards the negative samples due to the loss function, which leads to poor performance. The objects are relatively concentrated in the PANDA dataset, and the large background area leads to a higher number of negative samples.

In this study, we used the same experimental setup as [32], with 300 images of the first ten scenarios as the training set and 90 images of the last three scenarios as the test set. The number of images was much smaller than traditional object detection datasets, so we had to perform data augmentation to expand the training samples. Mosaic [24] combines multiple images to generate new samples for data enhancement. Copy-Paste [38] achieves SOTA results using the COCO dataset by randomly pasting objects from one image to arbitrary positions in another image. MixUp [39] superimposes two images by adjusting the transparency to improve the robustness of the detector. Test-time augmentation (TTA) [40] creates multiple augmented copies of each image in the test set. After predicting all images, the model fuses the copy results with the original image. Whether performing classification or segmentation, TTA has been proven to be an effective means of improvement [41]. In this paper, we experimentally demonstrated the effectiveness of the above data augmentation schemes.

In this competition, we also found inaccuracies in the labeling of the original dataset, with some objects heavily occluded by buildings and some objects being unlabeled, which would affect the performance of the detector. Our experiments proved that data cleaning is an effective means of improving scores.

We proposed a novel image cropping strategy, the midline method. At the same time, for the gigadetection track in the gigavision competition, we introduced a large number of useful techniques, including image cropping strategies, the separate training and fusion of pedestrians and vehicles, data cleaning, and model modification.

2.2. YOLOv5

This study used YOLOv5x6 as the baseline for gigapixel-level object detection. YOLOv5x6 uses CSPDarknet53 with an SPPF module as the backbone, which has four detection heads corresponding to different-scale objects. It is suitable for 1280 × 1280 pixel images. The specific model structure is shown in Figure 2.

YOLOv5x6.2 replaced the Focus module with a 6 × 6 convolution layer, which is theoretically equivalent. However, the 6x6 convolution layer is more efficient than Focus for some existing GPU devices. As shown in Figure 2, the backbone of YOLOv5x6 has five C3 layers. The input image is defined as $\mathbf{x}$, and the output of the 6x6 convolution layer is f_0. Then, the output of each C3 layer from top to bottom in the backbone can be obtained as f_i, where $i = 1, ..., 5$. These four features can be obtained as follows:

$$f_i = B_i(f_{i-1}), i = 1, ..., 5 \tag{1}$$

where $B_i(\cdot)$ denotes different blocks in the backbone.

The two C3 layers from bottom to top in the yellow part can be denoted as:

$$f_1{}' = Concat(f_4, Upsampling(SPPF(f_5))) \tag{2}$$

$$f_2{}' = Concat(f_3, Upsampling(f_4)) \tag{3}$$

where $Concat(\cdot)$ and $Upsampling(\cdot)$ are the concatenate and upsampling operations, respectively. $SPPF(\cdot)$ refers to the SPPF layer. The four C3 layers from bottom to top in the blue part can be denoted as:

$$f_i'' = \begin{cases} Concat(f_{i+1}'', f_{i+4}), i = 1 \\ Concat(f_{i+1}'', f_{i-1}'), i = 2, 3 \\ Concat(f_{i-2}, f_{i-2}'), i = 4 \end{cases} \quad (4)$$

The outputs of the last four prediction layers can be denoted as:

$$p_i = Conv(f_i''), i = 1, ..., 4 \quad (5)$$

where $Conv(\cdot)$ is the convolution operations.

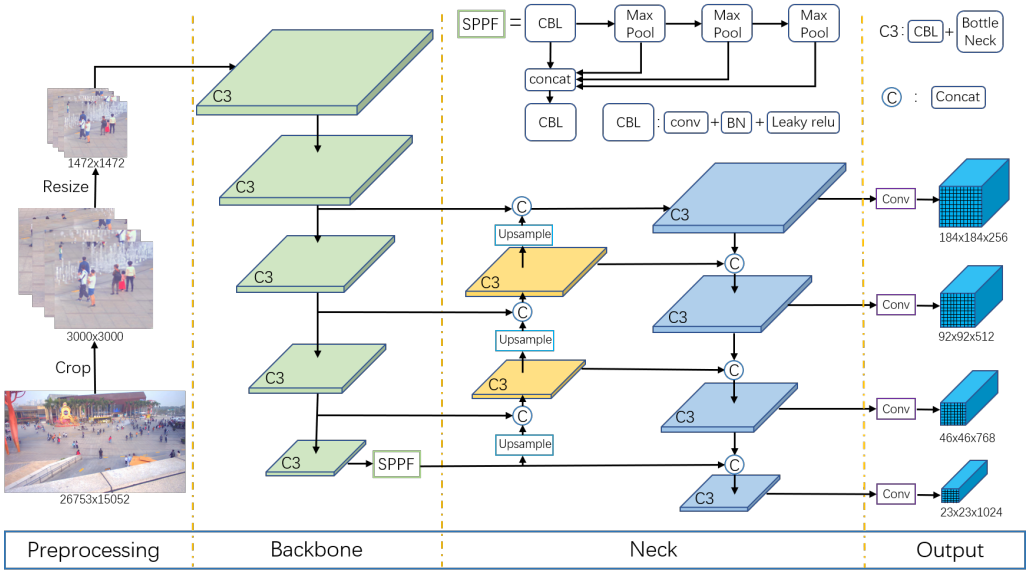

Figure 2. YOLOv5x6 framework. The preprocessing part crops the original image and inputs the cropped image into the network. The backbone uses CSPDarknet53 with a SPPF module. The neck is used for feature extraction. YOLOv5x6 can output four scales of feature map, which can perform target detection on larger images.

Since the number of images in the PANDA dataset is relatively small, we needed to use data augmentation techniques such as MixUp, Mosaic, and Flip. Mosaic was proposed by YOLOv5; it subjects four images to random scaling, random cropping, and random arrangement for splicing, which can improve the detection ability of the detector for small targets. In YOLOv3 and YOLOv4, one must define the value of the initial anchor in advance, which often leads to poor adaptability between the anchor boxes and ground truth. However, in YOLOv5, the authors used a clustering method to calculate the optimal anchor frame value adaptively.

In practical applications, the usual method is to resize the original image to a standard size. However, the size of the image often varies, leading to the redundant filling of black borders and affecting the inference speed of the detector. As a result, YOLOv5 adaptively adds minor padding to the original image, reducing the number of calculations and improving the inference speed of object detection.

2.3. Pipeline

In this section, we introduce the pipeline of the cropping-based object detection algorithm and the implementation of the proposed method. Firstly, we analyzed the target distribution of the PANDA dataset and chose a suitable cropping size. Secondly, to reduce the impact of the cropped target, we removed the labels below the ratio of the cropped

target to the original target in the sub-image below 0.35. Thirdly, we resized the sub-images into 1472 pixels, passed them through the YOLOv5 variant models, and obtained trained models. Next, we cropped the test images and detected them. Finally, we fused the results. The pipeline of the whole algorithm is shown in Algorithm 1.

Algorithm 1: Gigapixel-level Object Detection

Input: input gigapixel images Q
Output: final results
i represents the i-th image of dataset Q, $i = 0, ..., a - 1$;
for *i from 0 to* $a - 1$ **do**
    Crop image i with with size of 3000×3000 according to overlap = 0.5;
    Remove the labels below the ratio of cropping target to original target in sub-image below 0.35;
    Resize sub-image into 1472×1472;
    Obtain sub-image collection K
end
Preset the anchor box according to the size of target in sub-images;
Pass K to the YOLOv5 variant models and obtain trained models Ω;
Crop the test image and record the coordinates of the upper left corner of sub-image T;
Input T into Ω and obtain prediction result X ;
Stitch X and use midline method to remove the cropped labels;
Return final results;

2.4. Midline

In order to prevent the target from being ignored during the image-cropping process, it is often necessary to overlap the cropped images. The cropping process results in a large number of cropped objects, which significantly reduce the performance of the detector. Therefore, we proposed a midline post-processing method to remove the cropped targets. When the detection results of the sub-images were mapped back to the original image, adding the midline method improved the performance with almost no increase in calculations.

It can be seen from Figure 3 that the original image cropping scheme produced a cropped target, contained in the left sub-image in orange. When the two sub-images were finally fused, the cropped target frame that NMS could not remove led to performance degradation. As shown at the bottom of the figure, we made a decision at the midline of the overlapping part of the two pictures. If the target was on the left side of the midline, the right sub-picture discarded the target. Using the midline cropping scheme stopped this from happening.

In the PANDA dataset, we used overlap to crop the images, as shown in Figure 4. After predicting the cropped sub-images, we needed to fuse them into the original image and remove the non-maximum value frame by NMS. However, as shown in Figure 4b, some of the cropped objects could not be removed by NMS, and the confidence of these boxes was usually high, which would seriously affect the performance. When fusing cropped sub-images into the original image, we used the midline strategy to significantly improve the performance, which is shown in Figure 4c. This was because we removed a large number of small cropped objects that could not be removed. This method could generate improvements of 2%, as shown in Table 1.Both experiments and visualizations demonstrated that this approach improved the performance of the object detectors.

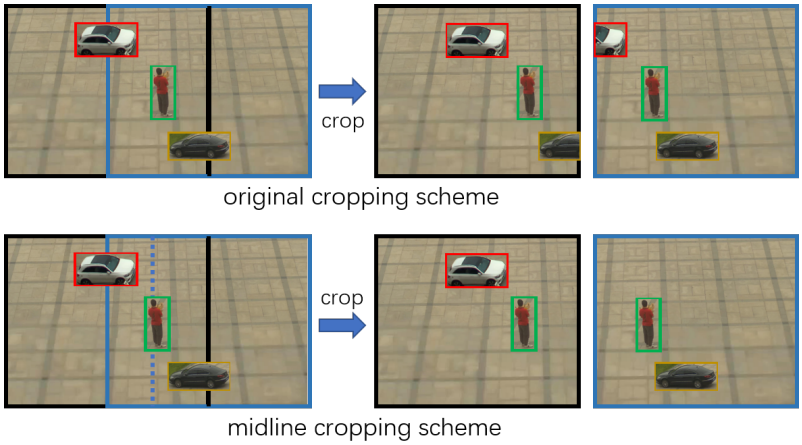

Figure 3. Midline image-cropping scheme. The top figure shows the original cropping scheme. The orange target in the black sub-image and the red target in the blue sub-image were cropped. The bottom figure shows the midline cropping scheme, which removed some cropping targets that NMS could not eliminate.



Figure 4. Visualization of midline in PANDA dataset. (**a**) Schematic of overlapping cuts. (**b**) After fusion and NMS. (**c**) After fusion, midline, and NMS.

Table 1. The experimental results for pedestrian and vehicle detection using YOLOv5x6.

Data Clean	Data Augment	VF Loss	TTA	Muti-Scale	Single-Class	Mid Line	AP50	AP75	mAP	AR500	FPS	mAP Test	AR500	Score
×	×	×	×	×	×	×	0.663	0.497	0.460	0.582	~0.17	0.43	0.5418	0.476
√	×	×	×	×	×	×	0.669	0.508	0.477	0.594	~0.17	0.45	0.6047	0.515
√	√	×	×	×	×	×	0.707	0.550	0.515	0.638	~0.17	0.50	0.6216	0.556
√	√	√	×	×	×	×	0.710	0.556	0.520	0.644	~0.17	0.52	0.6438	0.578
√	√	√	√	×	×	×	0.736	0.579	0.542	0.652	~0.07	0.55	0.6558	0.599
√	√	√	√	√	×	×	0.755	0.599	0.563	0.675	~0.07	0.57	0.6802	0.620
√	√	√	√	√	√	×	0.799	0.630	0.584	0.685	~0.04	0.59	0.6995	0.641
√	√	√	√	√	√	√	**0.819**	**0.673**	**0.614**	**0.698**	~0.04	**0.61**	**0.7117**	**0.659**

2.5. Single-Class Model

For gigapixel-level object detection, a standard solution is to crop the original image, then detect the cropped images and fuse the results. Due to the significant size difference between vehicles and pedestrians and the unbalanced number of samples, we conducted

single-object detection for vehicles and pedestrians to build expert models and finally fused them. The experiments proved that using different cropping sizes for different classes improved the results.

In order to choose an appropriate image cropping scheme, we analyzed the target size of the training set, and the results are shown in Figure 5. When we cropped the image with a width and height of 6000 × 4000, most of the targets were guaranteed to not be cropped. Since the object detector resized the long side of the image to 1472, some small targets became hard to recognize at this scale. If we used a size of 3000 × 3000, many large objects would be cropped. According to the visual analysis in Figure 5, we cropped pedestrians and vehicles to their appropriate sizes and trained the models separately.

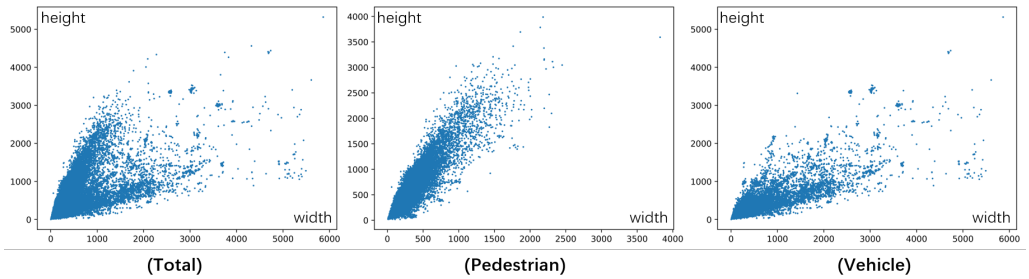

Figure 5. Cropping window size for digadetection. (Total) all objects in both pedestrian and vehicle categories.(Pedestrian) width and height distribution of pedestrians. (Vehicle) width and height distribution of vehicles.

However, some images of vehicles and pedestrians almost overlapped. The model may have extracted similar features, since single-object detection models cannot distinguish such a difference, and this may have affected the performance. Take a baby carriage and child as an example. Some babies sit in baby carriages, which may cause the ground truth of the two to overlap. The single-object detector cannot further compare the differences in this feature. The simple fusion of the results may lead to a decrease in performance.

Figure 6b,c show the different detection results for the baby carriage. The left-hand sides of (b) and (c) are the results of the two-class target detection models. The detector labeled the baby carriage as both a pedestrian and a vehicle, regardless of whether a baby was in the carriage. We used single-class object detectors for pedestrians and vehicles with different cropping sizes and fused the results. The right-hand side of (b) and (c) show that the detector could correctly identify the baby and baby carriage.

2.6. Data Cleaning

We found that the background (marble ground) was falsely detected as a target. Visual analysis showed severely occluded targets in the dataset, as depicted in Figure 6a (left). The part in the red box was visually similar to the marble floor in the feature map, which may have led to false positives. We also found that the targets in some images were not fully labeled (Figure 6a, right). These unlabeled targets would be treated as background, resulting in the target detection confidence not being as high as expected.

First of all, we marked some unlabeled objects, such as some obvious ublabeled pedestrians and vehicles. Second, since the images of the same scene were taken from the same angle at different times, the labeling of several long-term vehicles on the images of the same scene was not the same. Because the characteristics extracted from the same area were identical, the label was sometimes a car and sometimes foreground, which caused confusion in the detector. Finally, for some severely occluded objects, the ground truth was almost a billboard or tree, so we removed these bounding boxes.

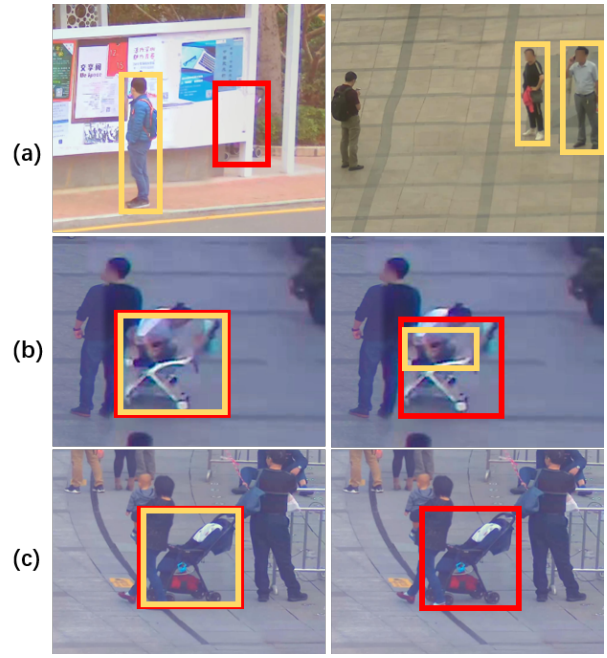

Figure 6. Visualization of some problems. (**a**) The baby carriage (red box) is severely occluded by the billboard, and the person on the left of the right image is unlabeled. (**b**) Detection of baby carriage with a baby. (**c**) Detection of baby carriage without a baby.

2.7. Loss Function

The loss function of YOLOv5 consists of three parts: class loss, objectness loss, and location loss. The loss function of YOLOv5x6 can be denoted as:

$$L(all) = L(obj) + L(x, y, w, h) + L(class) \tag{6}$$

The total loss can be calculated as the sum of the three parts or their weighted sum. The authors used the cross-entropy function to calculate classes and location loss. YOLOv5 also determines the CIoU to calculate objectness loss, which can be denoted as:

$$CIoU = 1 - IoU + \frac{\rho^2(\mathbf{b}, \mathbf{b}_{\mathrm{gt}})}{c^2} + \alpha v \tag{7}$$

where IoU refers to the intersection and union ratio of A and B; C represents the smallest box surrounding A and B; $\mathbf{b}$ and $\mathbf{b}_{\mathrm{gt}}$ are the center point of the prediction frame and the ground truth frame, respectively; $\rho^2(\cdot)$ denotes the Euclidean distance; and c represents the diagonal length of the smallest rectangle containing these two frames.

In order to solve the problem of unbalanced positive and negative samples, focal loss is implemented to further improve cross-entropy loss:

$$FL(p_t) = -\alpha_t (1 - p_t)^{\gamma} \log(p_t) \tag{8}$$

where p_t is the predicted probability, and α_t and γ are the focusing parameters, which reduce the weight of the samples that are easy to classify so that the model can pay more attention to the samples that are difficult to classify. In YOLOv5, the focal loss is integrated into the code.

In order to further solve the imbalance between foreground and background in dense object detection, [42] proposed varifocal loss (VFL), which obtains SOTA results using COCO dataset. The formula is as follows:

$$VFL(p_t, q) = \begin{cases} -q(q \log(p_t)) + (1 - q) \log(1 - p_t)) & q > 0 \\ -\alpha_t {p_t}^{\gamma} \log(1 - p_t) & q = 0 \end{cases} \tag{9}$$

where q is the label, and when it is a negative sample, q is 0. VarifocalNet only reduces the loss contribution of negative samples and preserves the learning information of positive samples. In this paper, we embedded VFL into YOLOv5 and conducted experiments on the PANDA dataset. The experimental results showed that VFL worked well in gigapixel-level object detection.

2.8. Attention Block

The attention mechanism can focus on important information with a high weight and ignore irrelevant information with a low weight, so that the network can pay more attention to key areas in the image and realize the efficient allocation of information processing resources. In recent years, attention mechanisms have been increasingly studied in computer vision research.

Hu [43] proposed the squeeze-and-excitation network (SENet), which considers the relationship between feature channels and adds an attention mechanism to feature channels. SENet automatically obtains the importance of each feature channel and utilizes the obtained feature importance ranking to make the network pay more attention to important features of the current task.

The convolutional block attention module (CBAM) [44] combines the two-dimensional attention mechanism of the feature channel and feature space. The method of extracting feature channel attention is similar to SENet. Spatial attention involves performing maximum pooling and average pooling in units of channels and concatenating the results.

The original lightweight network only considers the channel attention and ignores the location information. Coordinate attention (CA) [45] realizes the dual attention of position and channel by encoding horizontal and vertical location information into the channel attention.

This paper integrated the above three attention mechanisms into the YOLOv5 system-level code and tested its performance on the PANDA dataset. The experimental results showed that the attention mechanism still had ample room for improvement for gigapixel-level images.

2.9. Optimal Number of Anchors

Prior knowledge plays a vital role in guiding machine learning. An anchor is a kind of prior knowledge provided in advance, which can point out the possible size of the target box to speed up the learning ability of the model. Since YOLOv5 is an anchor-based model, the setting of the initial anchor significantly affects the convergence speed and effectiveness of the model.

According to the input bounding box size, YOLOv5 uses the K-means algorithm to pre-calculate K optimal anchors for each feature layer so that the deviation from the object pre-position is small. However, due to the view field and super-high resolution, the target scale in the image changes drastically. Therefore, YOLOv5's preset three-scale anchors for each feature layer are unsuitable for gigapixel-level images. Hence, each feature layer needs to preset more anchors to adapt to the scale change of the target.

3. Experiment

3.1. Dataset and Setting

3.1.1. Dataset

PANDA-Image is the world's first gigapixel-level video dataset. It consists of 18 scenes, and each scene has 30 images of approximately 25 k $\times$ 14 k resolution with a field of view coverage of up to 1 km^2. Furthermore, it allows thousands of targets to be observed simultaneously over nearly one hundred scale variations. Since the first 13 scenes are labeled, we first show the results from the competition implementing different improvements. Then, we used the first ten scenes as the training set and the last three as the test set for further verification. The dataset contains two categories: pedestrians and vehicles. At the same

time, due to the large number of targets in a single image, scale variation, and occlusion, it is quite challenging to perform object detection on PANDA-Image.

3.1.2. Evaluation Metrics

In this paper, we used standard the MS COCO [1] evaluation method: AP, AP_{50}, AR, and $AR_{\text{max=500}}$. The first AP refers to $AP_{0.50:0.05:0.95}$, which is the average AP with continuous IoU thresholds (from 0.50 to 0.95 in steps of 0.05). AP, AR, and $Score$ are calculated as follows:

$$AP = \frac{TP}{TP + FP} \tag{10}$$

$$AR = \frac{TP}{TP + FN} \tag{11}$$

$$Score = \frac{2 \times AP \times AR}{AP + AR} \tag{12}$$

where TP is the number of detected boxes with an IoU greater than the threshold, and FP refers to the lower number. FN indicates the number of undetected ground truths. AP refers to the average precision, which measures the degree of false detection. AR denotes the average recall, which is the number of true positives divided by the sum of true positives and false negatives. $Score$ represents the F1 score. The F1 score considers AP and AR to be equally important, and it can distinguish the pros and cons of the method very well.

3.1.3. Experimental Setup

We implemented our methods on Ubuntu 18.04, and all experiments were trained on a single NVIDIA RTX3090 GPU with a 24 GB memory. The methods were based on Pytorch 1.12.0, conda 4.12.0, and CUDA 11.7 and were pre-trained on the COCO dataset. We trained our methods on PANDA for 50 epochs, and the first 2 epochs were used for warm-up. We used the SGD optimizer with an initial learning rate of 0.00625, a momentum of 0.9, and a weight decay of 0.0001. We resized the input image to 1472 $\times$ 1472 and set the batch size to 5. The follow-up experiments explain further details of the experimental setup.

3.2. Cropping and Fusion

3.2.1. Cropping

We performed image cropping according to a sliding window of 3000 $\times$ 3000 and cropped the images of the first 13 scenes into 28,957 sub-images.In the sub-images, if the ratio of the clipped label frame of the target to the original target label box was less than 0.35, the label was discarded. In the process of sliding window cropping, we could not guarantee that the size of each original image was an integer multiple of 3000. The sub-images on the right-hand side and bottom were expanded to the upper left until their size was equal to 3000. Then, we resized the sub-images to 1472 $\times$ 1472. Secondly, we directly resized the original images to 1472. Finally, we took the images of the first ten scenes as the training set and the images of the last three scenes as the testing set.

3.2.2. Fusion

Our baseline model was based on the YOLOv5 framework. After detection, we obtained the results of the sub-image, whose name comprised two parts: the name of the original image and the coordinates of the upper-left corner. For example, in a sub-image called "IMG_01_03_6000_9000", "IMG_01_03" refers to the third picture of the first scene, and "6000_9000" refers to the coordinate of the upper-left corner in this original image. In the fusion process, we used the midline method to remove some of the cropped boxes to improve the system's performance. Since it was challenging to use the midline method to remove the cropped boxes of larger objects, we assumed that the boxes with a width and height greater than 800 pixels in the sub-image could be detected in the original image, so

we removed these cropped boxes in the sub-image. The experiments proved that this was an effective method. The inhibition coefficient of the region NMS algorithm was 0.6.

3.3. Evaluation Results

We used the public open-source YOLOv5-6.2 as a base model and improved it by implementing the above methods. The experimental results in Table 1 show how the useful techniques improved the detection results. TTA is the abbreviation of test time augmentation.

We experimented with all the methods under the same settings and gradually added method improvements to the network. The performance was poor when we directly input the cropped images into the network. During the competition, we found several problems in labeling the dataset, such as severely occluded and unlabeled objects, so we cleaned the data and achieved functionality. Then, we used Mosaic, with a 50% probability of image left–right flipping and a 10% probability of image mixup, and obtained 51.5% mAP and 50% mAP_{test}. After that, for challenge two (shown in Figure 1b), we used varifocal loss to reduce the impact of the positive and negative sample imbalance on the results by increasing the weight of difficult-to-classify samples. Previous research [20] has proved that TTA is an effective means of improvement, and our experiment showed the same conclusion. We adopted multi-scale training to improve the robustness of the detection model in relation to object size by training images of different scales. In the following experiment, we used 3000×3000 and 6000×4000 crop sizes to train pedestrians and vehicles and obtain their respective single-target detection models, before finally fusing them. The experiments showed that different cropping sizes could avoid the missed detection of small pedestrians and vehicle cropping. At the same time, this method could effectively learn the intra-class features of pedestrians and vehicles. We demonstrated that our proposed midline method suppressed cropped objects, improving the mAP and mAP_{test} by 3% and 2%, respectively.

3.4. Visualization Experiment

In this section, we provide the final visualization experiment results for the entire model with all of the techniques and methods employed.

Figure 7 shows the final prediction results and heat map of the gigadetection. The detection results showed that even if the targets in the dataset were scattered and there were a large number of background areas, the method using varifocal loss could still accurately predict the background and targets. The heatmap also accurately located the targets. The visualization experiment proved that the pipeline and method we adopted were reasonable and reliable.

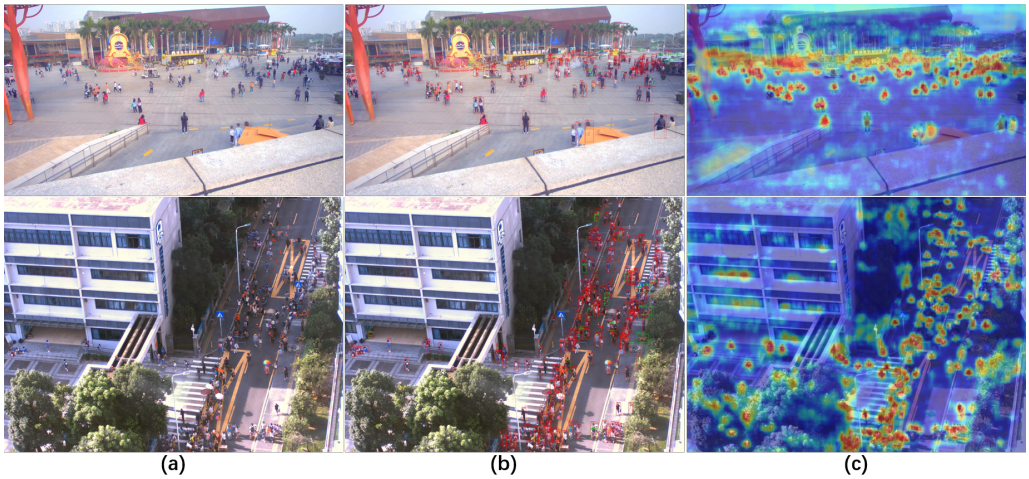

Figure 7. Visualization results of gigadetection. (**a**) Original image. (**b**) Prediction results. (**c**) Heat map.

3.5. Ablation Study

3.5.1. Comparison with State-of-the-Art Methods on PANDA

In this subsection, we compare our method with a series of state-of-the-art detectors on the PANDA dataset, and the results are shown in Table 2. Compared to the SOTA method PVDet, we achieved a 0.3 % total *AP* boost and 2.9 % pedestrain *AP* boost.

Table 2. Comparison with SOTA results.

Classes	**Pedestrian**	**Vehicle**	**Total**	
Method	**AP**	**AP**	**AP50**	**AP**
Region-NMS	46	-	-	-
SARNet	49.1	60.2	-	51.9
PVDet	59.2	**63.3**	**82.2**	61.1
Ours	**62.1**	59.9	81.9	**61.4**

3.5.2. Midline

We added the midline method when integrating the target detected by sub-images into the original image. Table 3 shows that this post-processing method could significantly improve the results of gigadetection based on cropped images. At the same time, it brought almost no additional calculation overhead. Regardless of the object classes, our proposed method dramatically improved the reliability and validity of the results.

Table 3. Midline method.

Cropped	**Pedestrian**			**Vehicle**			**Total**		
Method	**AP50**	**AP75**	**mAP**	**AP50**	**AP75**	**mAP**	**AP50**	**AP75**	**mAP**
Without	0.810	0.644	0.596	0.785	0.616	0.580	0.799	0.630	0.584
Midline	**0.833**	**0.682**	**0.621**	**0.809**	**0.644**	**0.599**	**0.819**	**0.673**	**0.614**

3.5.3. Slipping Window Size

Table 4 shows that if we directly detected the original image rather than using the cropping strategy, due to the severe downsampling, some objects were too small to detect. Next, we tested the effect of different cropping sizes. We used a cropping size of 6000 × 4000 and 3000 × 3000 to detect vehicles and pedestrians, and the experiments showed that the cropping size of 3000 × 3000 worked best for vehicle and pedestrian object detectors. Single-class fusion meant that we used cropping sizes of 3000 × 3000 for pedestrians and 6000 × 4000 for vehicles and fused the results. The experiments proved that single-class fusion had the best performance.

Table 4. Cropping window size.

Window Size	**AP50**	**AP75**	**mAP**	**AR500**	**mAP**	**AR500**	**Score**
Original	0.508	0.353	0.332	0.4012	0.34	0.4228	0.378
6000 × 4000	0.683	0.545	0.50	0.6075	0.45	0.6181	0.519
3000 × 3000	0.755	0.599	0.563	0.675	0.57	0.6802	0.620
Single-class fusion	**0.799**	**0.630**	**0.584**	**0.685**	**0.59**	**0.6995**	**0.641**

3.5.4. Loss Function

In this experiment, we verified the commonly used CIoU loss in object detection. CIOU loss considers the bounding box regression's overlapping area, center point distance, and aspect ratio. However, the difference in aspect ratio varies compared to the real difference between the width and height and their confidence, so it may hinder the model from effectively optimizing the similarity. Focal loss prevents a large number of simple negative samples from overwhelming the detector during training and focuses the detector on a sparse set of difficult examples. Therefore, the performance was better than CIoU loss. Varifocal loss proposes an awareness of the IoU based on focal loss to sort dense targets. At the same time, the weight of positive and negative samples was different. It was more

in line with the characteristics of gigapixel-level images where the target is concentrated in a specific image area. Table 5 demonstrates the superior performance of varifocal loss for this dataset.

Table 5. Different loss functions.

Loss Function	AP50	AP75	mAP	AR500	mAP	AR500	Score
CIoU loss	0.707	0.550	0.515	0.638	0.50	0.6216	0.556
Focal loss	**0.718**	**0.559**	0.519	0.638	0.51	0.6401	0.568
Varifocal loss	0.710	0.556	**0.520**	**0.644**	**0.52**	**0.6438**	**0.578**

3.5.5. Attention Block

In order to improve the performance of the detector, we applied various attention blocks to the results. Unfortunately, the experiments proved that attention module degraded the final performance. This may have been because we did not use the original image for training and testing as in other datasets. When we used a cropping scheme, some sub-images may not have contained any targets. At the same time, due to the large size of the images, it was difficult for these non-gigapixel-level attention blocks to adapt to the images.

In the experiment, we added an attention block after each C3 module in the head rather than the backbone. Because no attention block was added during pre-training, the effect of adding one to the backbone was not significant. Table 6 illustrates that traditional attention modules performed poorly in this problem. This was because the attention block generally works well on large datasets (e.g., COCO), while the gigadetection dataset had a small number of images.

Table 6. Different attention blocks.

Attention Block	AP50	AP75	mAP	AR500	mAP	AR500	Score
Original	**0.755**	**0.599**	**0.563**	**0.675**	**0.57**	**0.6802**	**0.620**
SE	0.754	0.593	0.556	0.666	0.56	0.6712	0.610
CBAM	0.750	0.590	0.558	0.669	0.56	0.6693	0.611
CA	0.755	0.598	0.558	0.667	0.56	0.6681	0.609

3.5.6. Number of Anchors

YOLOv5x6 has a total of four feature layers. In this section, we set each feature layer's initial number of anchors to three, four, and five, respectively. The K-means clustering algorithm obtained the anchor size. Since the anchors of each feature layer needed to contain targets of all scales in the layer, the anchor size varied with the initial anchor number. The experiments in Table 7 proved that the detection performance improved as the number of anchors increased by using the image cropping method for object detection on the gigadetection dataset.

Table 7. Number of anchors.

Number	AP50	AP75	mAP	AR500	mAP	AR500	Score
3	0.710	0.556	0.520	0.644	0.52	0.6438	0.578
4	0.715	0.559	0.523	0.643	0.53	0.6460	0.580
5	**0.715**	**0.563**	**0.524**	**0.647**	**0.53**	**0.6469**	**0.583**

3.5.7. Fusion Method

Non-maximum suppression (NMS) and weighted boxes fusion (WBF) are commonly used post-processing methods in multi-object detection. NMS first sorts according to the confidence score of the detection box and only takes the box with the highest confidence. WBF weights the adjacent boxes. Soft-NMS scores were scaled as an overlapping linear or Gaussian function instead of roughly deleting all boxes. However, the remaining detection boxes contained many false boxes caused by chopping. Table 8 proves that the performance

of NMS was better than WBF and Soft-NMS because NMS could remove some of the cropped boxes.

Table 8. Different post-processing methods.

Fusion Method	AP50	AP75	mAP	AR500	mAP	AR500	Score
NMS	**0.819**	0.673	**0.614**	0.698	**0.61**	0.7119	**0.659**
Soft-NMS	0.810	**0.675**	0.606	**0.699**	0.60	**0.7123**	0.654
WBF	0.812	0.661	0.609	0.684	0.61	0.6988	0.649

4. Conclusions

In order to achieve real-world pedestrian and vehicle detection, we proposed the midline method to solve the problem of a large number of cropped targets in image cropping. Compared with other studies, our method could improve the detection performance while rarely introducing calculations.. We used varifocal loss to solve the imbalance between positive and negative samples. By data cleaning, we solved the unreliability of some annotations in the PANDA dataset. We applied many techniques to gigapixel-level images to improve the accuracy of the detector. We achieved a 61.4% mAP for pedestrian and vehicle detection in the PANDA dataset. However, our proposed midline method had certain limitations, in that it performed poorly for large chopped objects and increased computation. Therefore, in future research, we plan to further improve the detection ability of the model through lightweight and improved methods. We hope this paper can help developers and researchers become more experienced in analyzing and processing ultra-high-resolution images.

Author Contributions: Conceptualization, C.W. and W.F.; methodology, C.W., W.F. and B.L.; validation, C.W. and W.F.; formal analysis, W.F. and C.W.; writing—original draft preparation, W.F. and C.W.; writing—review and editing, B.L. and Y.Y.; visualization, W.F. and C.W.; supervision, X.L.; funding acquisition, Y.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under grants 62072021 and 62002005.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We sincerely thank the authors of YOLOv5 and TPH-YOLOv5 for providing their algorithm codes to facilitate the comparative experiments.

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

References

1. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
2. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
3. Zhu, P.; Wen, L.; Bian, X.; Ling, H.; Hu, Q. Vision meets drones: A challenge. *arXiv* **2018**, arXiv:1804.07437.
4. Du, D.; Qi, Y.; Yu, H.; Yang, Y.; Duan, K.; Li, G.; Zhang, W.; Huang, Q.; Tian, Q. The unmanned aerial vehicle benchmark: Object detection and tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 370–386.
5. Wang, X.; Zhang, X.; Zhu, Y.; Guo, Y.; Yuan, X.; Xiang, L.; Wang, Z.; Ding, G.; Brady, D.; Dai, Q.; et al. Panda: A gigapixel-level human-centric video dataset. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 3268–3278.
6. Cristani, M.; Raghavendra, R.; Del Bue, A.; Murino, V. Human behavior analysis in video surveillance: A social signal processing perspective. *Neurocomputing* **2013**, *100*, 86–97. [CrossRef]
7. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]

8. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef] [PubMed]
9. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference On Computer Vision, Santiago, Chile 7–13 December 2015; pp. 1440–1448.
10. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28* , 1137–1149. [CrossRef] [PubMed]
11. Shrivastava, A.; Gupta, A.; Girshick, R. Training region-based object detectors with online hard example mining. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 761–769.
12. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
13. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
14. Zhu, Y.; Zhao, C.; Wang, J.; Zhao, X.; Wu, Y.; Lu, H. Couplenet: Coupling global structure with local parts for object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4126–4134.
15. Bodla, N.; Singh, B.; Chellappa, R.; Davis, L. Improving object detection with one line of code. *arXiv* **2017**, arXiv:1704.04503.
16. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
17. Wu, Y.; Chen, Y.; Yuan, L.; Liu, Z.; Wang, L.; Li, H.; Fu, Y. Rethinking classification and localization for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 10186–10195.
18. 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, 14–19 June 2020; pp. 10781–10790.
19. Ge, Z.; Liu, S.; Li, Z.; Yoshie, O.; Sun, J. Ota: Optimal transport assignment for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Online, 19–25 June 2021; pp. 303–312.
20. Li, S.; He, C.; Li, R.; Zhang, L. A Dual Weighting Label Assignment Scheme for Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 9387–9396.
21. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
22. 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 June 2017; pp. 7263–7271.
23. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
24. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
25. 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. Springer, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
26. Zhang, S.; Wen, L.; Bian, X.; Lei, Z.; Li, S.Z. Single-shot refinement neural network for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4203–4212.
27. Jocher, G.; Chaurasia, A.; Stoken, A.; Borovec, J.; NanoCode012.; Kwon, Y.; TaoXie.; Michael, K.; Fang, J. ultralytics/yolov5: V6.2-YOLOv5 Classification Models, Apple M1, Reproducibility, ClearML and Deci.ai integrations. *Zenodo* **2022**. Available online: https://github.com/ultralytics/yolov5/releases/tag/v6.2 (accessed on 9 October 2022).
28. Ozge Unel, F.; Ozkalayci, B.O.; Cigla, C. The power of tiling for small object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–17 June 2019.
29. Li, C.; Yang, T.; Zhu, S.; Chen, C.; Guan, S. Density map guided object detection in aerial images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 190–191.
30. Jin, C.; Tanno, R.; Mertzanidou, T.; Panagiotaki, E.; Alexander, D.C. Learning to downsample for segmentation of ultra-high resolution images. *arXiv* **2021**, arXiv:2109.11071.
31. Fan, J.; Liu, H.; Yang, W.; See, J.; Zhang, A.; Lin, W. Speed Up Object Detection on Gigapixel-Level Images With Patch Arrangement. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 4653–4661.
32. Li, L.; Guo, X.; Wang, Y.; Ma, J.; Jiao, L.; Liu, F.; Liu, X. Region NMS-based deep network for gigapixel level pedestrian detection with two-step cropping. *Neurocomputing* **2022**, *468*, 482–491. [CrossRef]
33. Chen, K.; Wang, Z.; Wang, X.; Gong, D.; Yu, L.; Guo, Y.; Ding, G. Towards real-time object detection in GigaPixel-level video. *Neurocomputing* **2022**, *477*, 14–24. [CrossRef]
34. Mo, W.; Zhang, W.; Wei, H.; Cao, R.; Ke, Y.; Luo, Y. PVDet: Towards pedestrian and vehicle detection on gigapixel-level images. *Eng. Appl. Artif. Intell.* **2023**, *118*, 105705. [CrossRef]
35. Wei, H.; Zhang, Q.; Han, J.; Fan, Y.; Qian, Y. SARNet: Spatial Attention Residual Network for pedestrian and vehicle detection in large scenes. *Appl. Intell.* **2022**, *52*, 17718–17733. [CrossRef]
36. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 2778–2788.

37. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
38. Ghiasi, G.; Cui, Y.; Srinivas, A.; Qian, R.; Lin, T.Y.; Cubuk, E.D.; Le, Q.V.; Zoph, B. Simple copy-paste is a strong data augmentation method for instance segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Online, 19–25 June 2021; pp. 2918–2928.
39. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv* **2017**, arXiv:1710.09412.
40. Shanmugam, D.; Blalock, D.; Balakrishnan, G.; Guttag, J. Better aggregation in test-time augmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 1214–1223.
41. Kim, I.; Kim, Y.; Kim, S. Learning loss for test-time augmentation. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 4163–4174.
42. Zhang, H.; Wang, Y.; Dayoub, F.; Sunderhauf, N. Varifocalnet: An iou-aware dense object detector. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 8514–8523.
43. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
44. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
45. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Online, 19–25 June 2021; pp. 13713–13722.

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.