remote sensing

MDPI

Article

GANsformer: A Detection Network for Aerial Images with High Performance Combining Convolutional Network and Transformer

Yan Zhang [1,†], Xi Liu [2,†], Shiyun Wa [1], Shuyu Chen [3] and Qin Ma [1,*]

[1] College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China; 2019308250102@cau.edu.cn (Y.Z.); 2019308250126@cau.edu.cn (S.W.)
[2] College of Humanities and Development, China Agricultural University, Beijing 100083, China; 2019312100112@cau.edu.cn
[3] College of Engineering, China Agricultural University, Beijing 100083, China; shuyu.chen@cau.edu.cn
* Correspondence: maq782003@cau.edu.cn
† These authors contributed equally to this work.

Abstract: There has been substantial progress in small object detection in aerial images in recent years, due to the extensive applications and improved performances of convolutional neural networks (CNNs). Typically, traditional machine learning algorithms tend to prioritize inference speed over accuracy. Insufficient samples can cause problems for convolutional neural networks, such as instability, non-convergence, and overfitting. Additionally, detecting aerial images has inherent challenges, such as varying altitudes and illuminance situations, and blurred and dense objects, resulting in low detection accuracy. As a result, this paper adds a transformer backbone attention mechanism as a branch network, using the region-wide feature information. This paper also employs a generative model to expand the input aerial images ahead of the backbone. The respective advantages of the generative model and transformer network are incorporated. On the dataset presented in this study, the model achieves 96.77% precision, 98.83% recall, and 97.91% *mAP* by adding the Multi-GANs module to the one-stage detection network. These three indices are enhanced by 13.9%, 20.54%, and 10.27%, respectively, when compared to the other detection networks. Furthermore, this study provides an auto-pruning technique that may achieve 32.2 FPS inference speed with a minor performance loss while responding to the real-time detection task's usage environment. This research also develops a macOS application for the proposed algorithm using Swift development technology.

Keywords: object detection; transformer; deep learning; aerial image; generative model; GANsformer detection network

Citation: Zhang, Y.; Liu, X.; Wa, S.; Chen, S.; Ma, Q. GANsformer: A Detection Network for Aerial Images with High Performance Combining Convolutional Network and Transformer. *Remote Sens.* **2022**, *14*, 923. https://doi.org/10.3390/rs14040923

Academic Editors: Xiaoli Li, Zhenghua Chen, Min Wu and Jianfei Yang

Received: 18 January 2022
Accepted: 9 February 2022
Published: 14 February 2022

1. Introduction

In recent years, there has been significant progress in the development of aerial image detection [1–3]. Traditional target detection typically employs images acquired on the ground, and its dataset has limitations that constrain the scope of target detection research. Moreover, there are objective difficulties in acquiring particular images, such as capturing images in extreme geographical locations or images with large objects. However, aerial image detection—an improved and prevalent one for object detection studies—optimizes these issues, broadens the research scope of object detection, and makes access to images more flexible and convenient. Thanks to advancements in associated image detection algorithms and approaches in capturing aerial images, strong support for improved aerial image object detection has been provided. As a result, aerial image object detection technologies will become increasingly important in the future.

To categorize objects in an image [4], traditional machine learning approaches generally capture handmade features, such as corners and edges. Other classic methods, such as frame difference, optical flow, and histogram of oriented gradients [5], have great

real-time inference speed due to their comparatively simple computations. However, because they have been trained on specific qualities, they often perform poorly (in accuracy) [6]. In particular, regarding the subject topic of this paper, if previously unseen datasets [6]—comprised of various objects and collected under diverse situations—emerge, these techniques function defectively and with inferior accuracy.

Deep learning methods have recently experienced significant advances in computer vision. Numerous detection approaches and algorithms have proven to be superior in image detecting performance. Specifically, Wei Zhang et al. [7] introduced a CNN-CAPSNet—an efficient remote-sensing image classification architecture, which takes advantage of the strengths of CNN and CAPSNet models. As a first feature maps extractor, a CNN without completely connected layers was utilized. Moreover, they chose a pre-trained deep CNN model, which was utterly trained on the ImageNet dataset as a feature extractor. Statistical results showed that the presented method could achieve competitive classification performance compared to state-of-the-art methods. Mingjie Liu et al. [6] created a unique approach for detecting small objects in an unmanned aerial vehicle (UAV) application. They originally optimized the ResBlock in darknet using YOLOv3 by concatenating two ResNet units with the same width and height. Subsequently, the overall architecture of the darknet was enhanced by improving convolution operation to enrich spatial information. In addition, they gathered a UAV-viewed dataset for small object detection. Minh-Tan Pham et al. [8] presented YOLO-fine—an improved one-stage deep learning-based detection model—on the basis of YOLOv3's structure. The detector was designed to detect small items with great precision and speed, permitting real-time applications in practical situations. They further investigated its robustness to the emergence of new backdrops on the validation set, addressing the essential obstacle of domain adaptation in remote sensing. Experiments on aerial and satellite benchmark datasets reveal that YOLO-fine performs significantly better than other state-of-the-art object detectors. Nevertheless, even if Mask R-CNN [9] and Faster R-CNN [10] have high accuracy, their detection speeds are insufficient for satisfactory performance in practical applications. In the meantime, most previous scholars trained and tested these algorithms on huge natural photos, as opposed to aerial images. PASCAL VOC [11] is one of the most notable instances.

Although object detection in aerial images is a significant research field, it contains uncertainties and challenges during image capturing and processing.

1. Foremost, with variable altitude conditions and smaller object sizes, aerial images contain a considerable variation. Altitude causes diverse clarity and sizes of objects with varying resolutions in captured images. If the images were captured in an excessively high-altitude circumstance, images of the same objects would be viewed as different shapes and blurred, complicating the differentiating.
2. Afterward, under different altitude circumstances, corresponding mutable illuminance appears. Assuming images were captured at a relatively higher altitude, the lighting condition would worsen, and objects could reach less favorable sharpness [12].
3. Moreover, aerial images are prone to have multiple tiny objects in an individual image. Some of them probably distribute densely [13], which causes mutual shielding. Thus, using the pre-trained state-of-the-art models would generate lower accuracy.
4. In addition, when capturing aerial images, it would be affected by the weather. Moreover, there are clutters in the high altitude that interfere with the target detection object, such as flying birds, flying insects, and leaves.

Fortunately, some scholars have proposed various methods to tackle the problem of small object detection in aerial images mentioned above. For example, Chuanyang Liu et al. [14] developed an MTI-YOLO network for detecting insulators in complicated aerial pictures. They gathered composite insulator photos from various scenes and created a CCIN detection dataset. The provided MTI-YOLO network, as well as other comparative networks, were then trained and validated on the established dataset. The network's *AP* is 17% and 9% higher than YOLO-tiny and YOLO-v2, respectively. The suggested

network has a somewhat longer operational time than YOLO-tiny and YOLO-v2. Furthermore, the suggested network uses 25.6% and 38.9% less memory than YOLO-v2 and YOLO-v3, respectively. Luc Courtrai et al. [15] proved that the cycle Wasserstein GAN with an auxiliary YOLOv3 network performs properly in high-quality images with a shallow spatial resolution (up to 1m/pixel). Jakaria Rabbi et al. [16] suggested an end-to-end architecture that concurrently performs object detection and super-resolution to gain discernible boundary information of small-object images and alleviate the massive expense of HR photography for broad areas. To improve feature extraction ability, Danqing Xu et al. [17] deployed DenseNet. Furthermore, based on the original YOLO-V3, the detection scales were extended to four. In terms of accuracy, experiments on the RSOD and UCS-AOD datasets revealed that our approach outperformed Faster-RCNN, SSD, YOLO-V3, and YOLO-V3 small datasets. In the RSOD dataset, their approach's mAP increased from 77.10% to 88.73% compared to the original YOLO-V3. The mAP of detecting targets such as aircraft rose by 12.12% in particular. Danilo Avola et al. [18] suggested a viable and efficacious multi-stream architecture, applying diverse kernel sizes to manufacture multi-scale image analyses. They manipulated this architecture as a backbone and then developed a multi-stream-Faster R-CNN object detector, which is satisfactorily capable of achieving real-time tracking on UAV images. The performance of the proposed pipeline attained the state-of-the-art, demonstrating that the suggested multi-stream technique can accurately simulate the robust multi-scale image processing paradigm. Ren Jin et al. [19] presented an efficient algorithm in aerial images object detection. They neatly ameliorated the restraint of inputting object samples in the multi-scale training procedure. Specifically, these researchers adopted metric learning to attain the scale representation boundary of each object category. Subsequently, tiny indiscernible objects were merged into small object regions. Consequently, they established a relationship between multi-scale training and multi-scale inference. Following the aforementioned processes, the suggested technique was validated on three renowned aerial image datasets: VisDrone, DOTA, and UAVDT. Experimentally, this approach was shown to enhance detection accuracy while declining the number of processing pixels.

However, the small-object detection issues outlined above, as well as the inferior detection accuracy in this area, have not been effectively and meaningfully solved. CNNs have difficulty capturing global features, such as long-distance relationships between visual elements, often critical for advanced computer vision tasks. An intuitive solution is to expand the receptive field. Nonetheless, this solution may require more intensive but disruptive pooling operations. Recently, the transformer architecture has been introduced for vision tasks. The visual transformer (ViT) reflects the complex spatial transformations and long-range feature dependencies that make up the global features. Unfortunately, the observed visual transformer ignores local feature details, which reduces the discriminability between background and foreground. Improved visual transformers propose a tokenization module or use CNN feature maps as input tokens to capture feature adjacency information. However, questions remain about how to embed local features and global features precisely. Moreover, this architecture can outperform CNNs only when based on immense datasets.

Given the transformer's huge and irreconcilable shortcoming, this research utilizes the core idea, attention, of the transformer. It then proposes a high-performance network for detecting aerial images that incorporates a convolutional network and the transformer, optimizing the mainstream detection network. The following are the study's key contributions:

1. We modified the transformer to reduce the number of parameters, improve the training speed, and act as a branch network to improve CNN's ability to capture global features. Because GANsformer inherits and combines the structural and global feature extraction advantages of CNN and visual transformers, its performance is significantly better than CNN and ViT with comparable parameter complexity. GANsformer has demonstrated its remarkable potential capability in aerial images detection tasks. Eventually, on the validation set, the suggested technique achieves 96.77%, 98.86%,

and 97.91% on *Precision*, *Recall*, and *mAP*, respectively. This experimental result indicates that the suggested model outperforms all other comparison models.

2. In Section 6, we evaluated the performance of various combinations of generative models to verify the efficacy of Multi-GANs implementations. Experimentally, the SPA-GAN model performs best in the attention extraction module, whereas the WGAN model works best in image augmentation.
3. Moreover, the detection task's loss function is optimized by substituting the *IoU* loss with a more appropriate *CIoU* loss.
4. Additionally, this paper established a detection application based on the macOS. The optimized model based on the proposed method, when being integrated in the practical application device, could also effectively and satisfactorily detect objects in aerial images.

Despite the above outstanding achievements, there are a few limitations in this research. In the beginning, even though this paper attempted numerous strategies to solve the imbalance problem of small sample datasets, the detection accuracy remained the lowest across all classes in terms of detection outcomes. Furthermore, the suggested detection network is based on the one-stage network with intrinsic flaws. Lastly, the loss function's design has the possibility of being enhanced. These are challenges that the authors of this paper will need to overcome in the future.

The remainder of this paper is arranged as follows: Section 3 introduces the dataset and describes the Multi-GANs detection network. Section 4 explains the experimental setup and evaluation indicators. Section 5 discusses validation and detection results and analyzes the experimental results. Section 6 conducts numerous ablation experiments to verify the efficacy of the optimized method. Section 7 is a summary of the entire paper.

2. Related Work

Object detection is a critical research topic in the "science" of computer vision. With the growth of deep learning [20,21] and machine learning [22–24] in image detection applications, various sophisticated computer vision systems used for evaluating object detections have been presented. Up until 2012, the classic machine learning approach was commonly used to detect objects. Afterward, CNN-based models for detecting objects could be classified into two types: one-stage and two-stage models. The single shot multi-box detector (SSD) [25–27], You Only Look Once (YOLO) [28–30], and EfficientDet [31] series are all one-stage versions. Meanwhile, the one-stage approach collects features from the network directly to forecast the object type and position. Faster R-CNN [10] and Mask R-CNN [9] are two-stage models. The two-stage approach must generate proposals—a predetermined box containing probable objects to be detected—before performing fine-grained object detection. As a result, the two-stage approach has a comparatively slow speed as it must repeat the classification and detection procedure numerous time and again. The alternative one-stage object detection technique, on the other hand, feeds all of the bounding boxes into a network merely once and then forecasts them, making it fast and appropriate for mobile. Thereby, we chose the one-stage detection network in this research.

Numerous novel CNN approaches are being created, based on investigators merging new modules and advancements in linked disciplines, such as industry, agriculture, and medicine. For example, Yan Zhang et al. [32] suggested a CNN augmented by a MAF module in the agricultural field. This work used image preprocessing to broaden and augment the illness samples, warming up methods, and transfer learning to accelerate training. The proposed system could detect three types of maize illnesses efficiently and correctly, achieving 97.41% accuracy (in the validation set), outperforming traditional AI methods. A CNN-based detection network with a pruning inference and generative module was also previously suggested [33]. The pruning inference provided here dynamically disabled a portion of the network structure in various conditions, reduced parameter amounts and processes, and accelerated the network. When detecting apple blooms, this model achieved 90.01% (precision), 98.79% (recall), and 97.43% (*mAP*), respectively. The inference speed

exceeded 29 frames per second. In the case of pear flaws detection, scholars developed an upgraded CNN model; more specifically, a deep convolutional adversarial generation network was used to extend the diseased pictures [34]. According to the experimental data, the detection accuracy of the proposed technique was 97.35% (in the validation). Furthermore, this model performed well on two unseen types of pears, demonstrating its generalization potential and robustness.

CNNs are remarkable for advanced computer vision tasks, such as image classification, target detection, and instance segmentation. The impressive performances of CNNs in these tasks are mainly attributed to convolutional operations. That is, hierarchically collecting local features as powerful image representations. Despite the advantages in local feature extraction, capturing the global features remains a challenge for CNNs, but it is often critical for advanced computer vision tasks. There are two feasible solutions to alleviate this limitation:

1. One viable solution is to define larger receptive fields by introducing deeper architectures or more pooling operations. Dilated convolution methods [35,36] increase the sampling step, while deformable convolution learns the sampling position. SENet [37] and GENet propose using global AvgPooling to aggregate the global context and then rethinking the feature channels. In contrast, CBAM [38,39] uses global max pooling and AvgPooling to independently refine features in the spatial and channel dimensions, respectively.
2. Another possible solution is the global attention mechanism [40–42], which has significant advantages in capturing long-range dependencies in natural language processing. Inspired by the non-local means approach, non-local operations are introduced into CNNs via a self-attention mode. Thus, the response at each location is a weighted sum of global location features. The attention convolutional network [43] connects the convolutional feature map with the self-attention feature map to enhance the convolutional operation and, thus, capture remote interactions.

Recently, transformer architectures have been introduced for vision tasks. As a pioneering work, visual transformer (ViT) [44–46] validates the feasibility of pure transformer architectures for computer vision tasks. To exploit long-range dependencies, transformer blocks act as standalone architectures or are introduced into CNNs for image classification, target detection, semantic segmentation, image enhancement, and image generation. However, the self-attentive mechanism in visual transformers often ignores local feature details. Moreover, transformers usually outperform CNNs' overall performance only on massive datasets. Given this vast and irreconcilable drawback, this study refers to the transformer's prominent attention. Then we incorporate it with a convolutional network to propose the network in this paper.

3. Materials and Methods

We used the aerial images dataset, which was released by Northwestern Polytechnical University in 2016, a publicly available level 10 geospatial object detection remote sensing dataset. This dataset contains a total of 800 images, including 650 images of the object and 150 images of the background, with 10 categories of objects: aircraft, ships, oil tanks, baseball fields, basketball courts, athletic fields, ports, bridges, tennis fields, and vehicles [47], as shown in Figure 1.

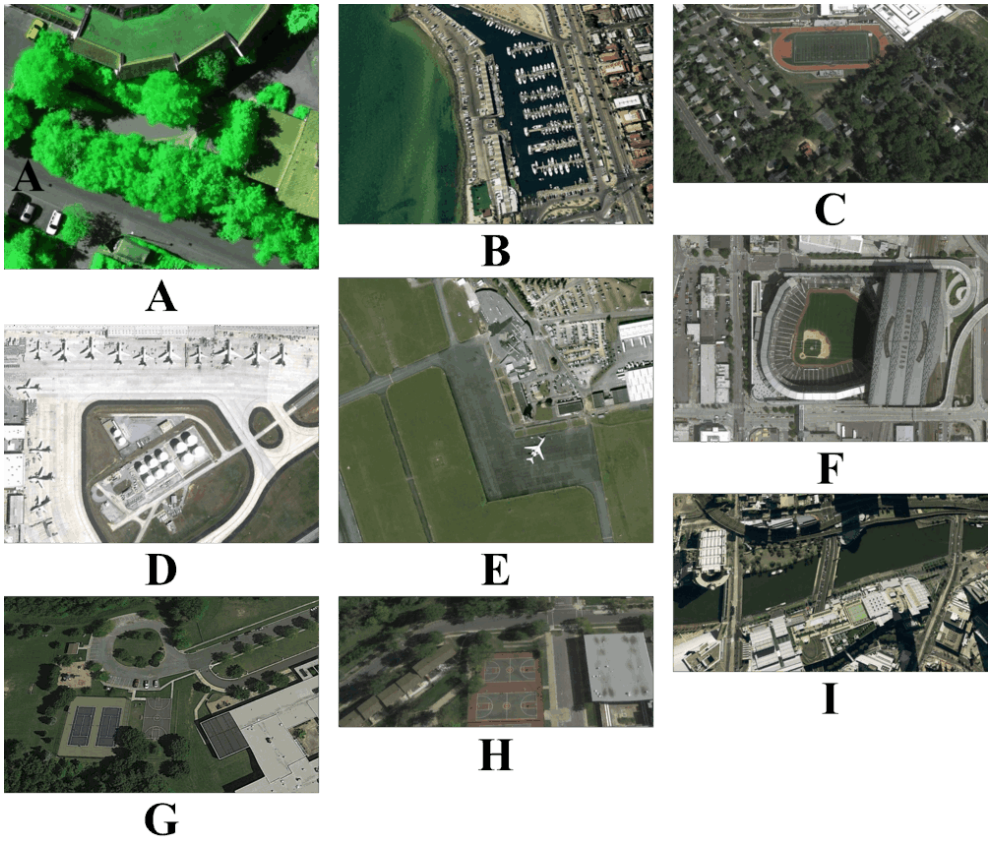

Figure 1. Dataset visualization. (**A**–**I**) shows multiple targets and multiple scales of the dataset: (**A**) is a green channel-enhanced image containing cars; (**B**) contains a vast number of dense ships; (**C**) is an athletic field, with a much larger scale than the ship in (**B**); (**D**) contains both airplane and oil tank objects to be detected; (**E**) contains an airplane; (**F**) contains a baseball field; (**G**) contains tennis and basketball courts; (**H**) contains two basketball courts; (**I**) contains a bridge, with the most huge scale of all targets.

3.1. Data Augmentation

As analyzed above, the dataset contains 10 kinds of objects to be detected, although the total number of images is only 800. However, deep learning, especially the CNN model, needs a large amount of data to drive the training process. In this paper, the augmentation operation was performed on the dataset first.

3.1.1. Basic Augmentation

In this paper, we referred to the method proposed by Alex et al. [48]. We used image flipping, translation, and scaling for simple data augmentation. Image flipping and image translation mainly improve the model's accuracy by increasing the amount of data. The primary purpose of image scaling is to improve the robustness of the model to different scales of targets. As mentioned in Section 4.1, the scales of different targets are not the same, so it is necessary to improve the model's ability to detect the same target at different scales. The specific implementation of image scaling is image affine transformation.

The target image's width and height are anticipated to be w_{target} and h_{target}, whereas those of the original image are w_{origin} and h_{origin}. Equation (1) illustrates that when images are enlarged and shrunk, the Ω, which represents the scaling factor, is first defined. At that moment, we split the width and height of the original image through Ω. Afterward, after the target frame's center point intersects with that of the processed image, we take a fragment inside the target frame.

$$\Omega = min\{\frac{h_{target}}{h_{origin}}, \frac{w_{target}}{w_{origin}}\} \quad (1)$$

In addition to the above spatial and scale data enhancements, this paper also uses fundamental color channel transformations, such as HSV channel color change, to enhance the recognition performance of the model for different lighting conditions.

3.1.2. Advanced Augmentation

In addition to the primary data enhancement methods mentioned above, we also used the following five advanced augmentations to tackle the problem above of uneven amounts of data for different kinds of aerial images, as shown in Figure 2.

In order to solve the enormous memory loss and the network's unsatisfactory sensitivity to adversarial examples, we referred to the method depicted in the Mixup [49], which is intended to address the network's massive memory loss and inadequate sensitivity to adversarial samples. Since the model used in this paper includes the Multi-GANs module, enhancing the sensitivity of the adversarial samples can improve the module's precision. The method is shown in the Equations (2)–(4).

$$\lambda = Beta(\alpha, \beta) \quad (2)$$

$$mixed\_batch_x = \lambda \times batch_{x1} + (1 - \lambda) \times batch_{x2} \quad (3)$$

$$mixed\_batch_y = \lambda \times batch_{y1} + (1 - \lambda) \times batch_{y2} \quad (4)$$

The Cutout [50] method arbitrarily chops out a portion of the sample and fills it with a specific pixel, while the categorization result stays unaffected. Cutout is accomplished by masking the image with a defined size rectangle and setting all values within the rectangle to zero or other solid color values. Cutout allows the convolutional neural network to use global information from the entire image rather than local information from a few minor features.

The CutMix [51] method is adopted to remove a portion of the region. The other data in the training set's area pixel values are stochastically filled rather than filling zero pixels. CutMix allows the model to recognize two targets from a local view of an image, increasing the training efficiency. It also allows the model to focus on the areas where the target is hard to differentiate. However, there is a lack of knowledge in several areas, which will impact training efficiency.

The SnapMix [52] method randomly cuts out some areas in the sample. It fills them with a particular patch from other images stochastically, and the classification label remains unchanged.

The Mosaic [29] method can utilize multiple images at once. The most crucial benefit of Mosaic is that it can improve the background of identified objects. Multiple images' data will be counted in the BatchNorm calculation, significantly enhancing the model's generalization.

These specific effects are shown in Figure 2.

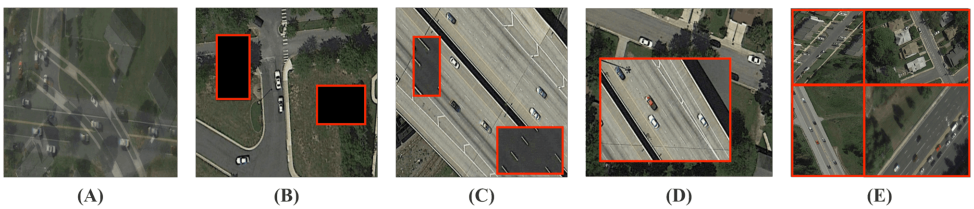

Figure 2. Illustration of five augmentation methods. (**A**) Mixup; (**B**) Cutout; (**C**) CutMix; (**D**) SnapMix; (**E**) Mosaic.

3.2. GANsformer Detection Network

Mainstream one-stage object detection models—YOLO [28,29,53,54] and SSD [25,26] are frequently utilized in target detection and have demonstrated outstanding performance on MS COCO [55] and Pascal VOC [56] datasets. However, the characteristics of YOLO series are not suitable for detecting aerial images.

As mentioned in the analysis of the dataset characteristics, there are high-density small object detection scenarios in practical applications. The general approaches to solve the small object detection problem include: increasing the resolution of the input image, which increases the computational complexity, and multi-scale feature representation, which makes the results uncontrollable. At present, the mainstream detection network incorporates the feature pyramid network (FPN) [57]. After the backbone extracts the features, the FPN contains the neck network with the fusion of deep feature maps and shallow feature maps. This structure improves the network's detection ability for different scales of objects. Nevertheless, it also complicates the network and has the possibility of overfitting. Therefore, this paper proposes a Multi-GANs structure, aiming to improve the above problem and enhance the model performance of the detection network. The main idea is to add a generative network model in front of the backbone of the detection network to augment the dataset. Subsequently, add a feature extractor based on the generative network model in the backbone of the detection network, improving CNN's feature extraction capability. The subsequent neck network and head network function will work more satisfactorily and efficiently when enough features are extracted.

Compared with the mainstream object detection models, including one-stage and two-stage, the main innovation of the GANsformer detection network is:

1. Two generative network models are added to the network to address the inadequate training of CNNs due to small datasets and improve the ability of deep CNNs to extract image features.
2. We modified the transformer, by reducing the number of parameters, improving the training speed, to improve the CNN's ability to capture global features as a branch network. Because it inherits and combines the structural and global feature extraction advantages of CNN and visual transformers, The performance of GANsformer is significantly better than CNN and vision transformer with comparable parameter complexity, showing the great potential capability in aerial images detection tasks.
3. Use Mixup, Cutout, CutMix, SnapMix, and Mosaic data augmentation methods to reduce overfitting and render the detection network to identify smaller-scale objects better.
4. Using label smoothing techniques and optimizing the loss function to improve the performance of GANsformer detection network.
5. Improve the NMS algorithm in the detection network by adding weight coefficients to fuse the bounding boxes.

Figure 3 illustrates the structure of the GANsformer detection network.

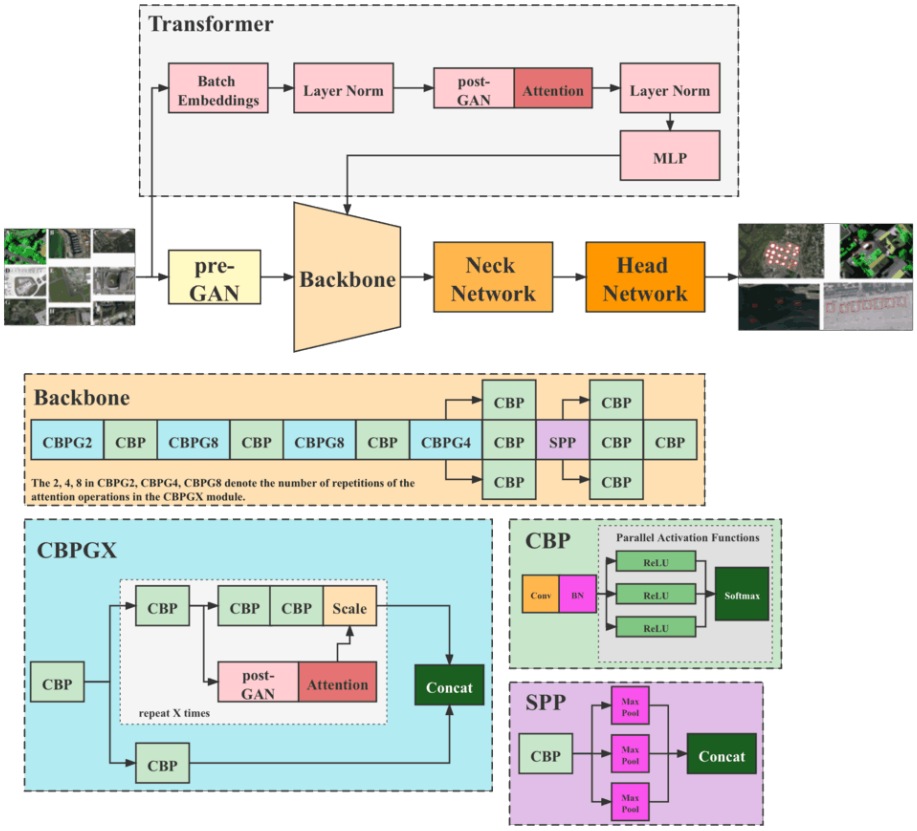

Figure 3. Structure of the GANsformer detection network.

3.2.1. Multi-GANs

Multi-GANs comprise two GAN modules: the pre-GAN and the post-GAN, as shown in Figure 3. Pre-GAN is placed before the backbone of the one-stage detection network to expand aerial images. The GAN module here can be implemented using various algorithms. Algorithm 1 shows the process of implementing pre-GAN using WGAN in the form of pseudo-code.

Algorithm 1 WGAN. $\alpha = 0.00005, c = 0.01, m = 64, n_{critic} = 5$

Input: dataset D
Output: dataset D'
Require: α, the learning rate. c, the clipping parameter. m, the batch size. n_{critic}, the number of iterations of the critic per generator iteration.
Require: ω_0, initial critic parameters. θ_0, initial generator's parameters.
while θ has not converged **do**
  for $t = 0, \cdots, n_{critic}$ **do**
    Sample $\{x^{(i)}\}_{i=1}^{m} \sim \mathbb{P}_r$ a batch from the real data.
    Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a batch of prior samples.
    $g_w \leftarrow \nabla_\omega \{\frac{1}{m}\sum_{i=1}^{m} f_\omega(x^{(i)}) - \frac{1}{m}\sum_{i=1}^{m} f_\omega(g_\theta(z^{(i)}))\}$
    $\omega \leftarrow \omega + \alpha \cdot RMSProp(\omega, g_\omega)$
    $\omega \leftarrow clip(\omega, -c, c)$
  end for
  Sample $\{z^{(i)}{}_{i=1}^{m} \sim p(z)\}$ a batch of prior samples.
  $g_\theta \leftarrow \nabla_\omega \frac{1}{m} f_\omega(g_\theta(z^{(i)}))$
  $\theta \leftarrow \theta - \alpha \cdot RMSProp(\theta, g_\theta)$
end while

The Wasserstein generative adversarial network (WGAN) [58,59] differs from the original GAN in that it uses the Wasserstein distance instead of the Jensen–Shannon difference to evaluate the difference in distribution between the actual and generated samples. This change makes the training faster and the training process more stable. The lack of stability in the training process is one of the drawbacks of the original GAN. To address the issues of the original GAN, WGAN gives four targeted improvement points.

1. The last layer of the discriminator removes the sigmoid.
2. The values of the generator and discriminator are not calculated logarithmically.
3. Truncate the absolute value after each discriminator parameter update, limiting it to a fixed constant c and not greater than it.
4. Use the RMSProp and SGD algorithms instead of momentum optimization algorithms, including momentum and Adam.

Another probable implementation of pre-GAN is the balancing GAN [60], introduced by IBM, which is a specific improvement on ACGAN [61], specifically designed to solve the problem of small sample size in unbalanced datasets. Figure 4 reflects the network structure of BAGAN.

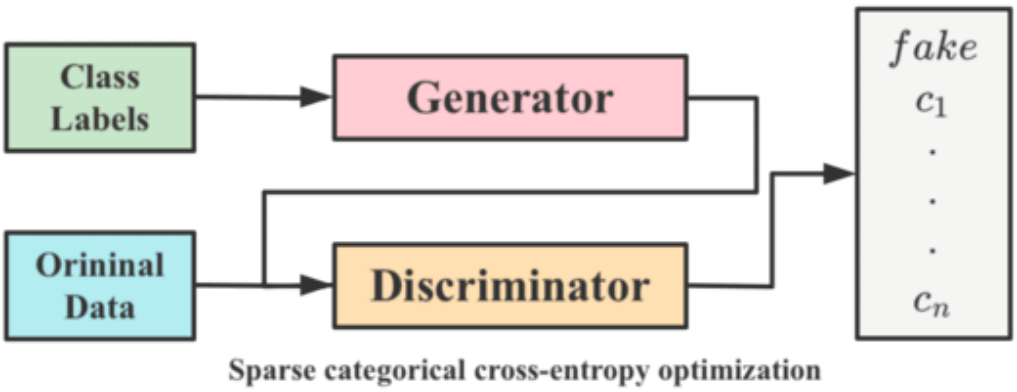

Figure 4. Network structure of BAGAN.

The post-GAN locates in the attention mechanism module, as depicted in Figure 3. Its principal function is to add a noise mask to the feature maps taken from the backbone to enhance the model's robustness. The following Result Section 5 section displays that introducing noise can considerably enhance model performance. The post-GAN module can be built in a variety of ways. SAGAN, for example, is used as indicated in Figure 5.

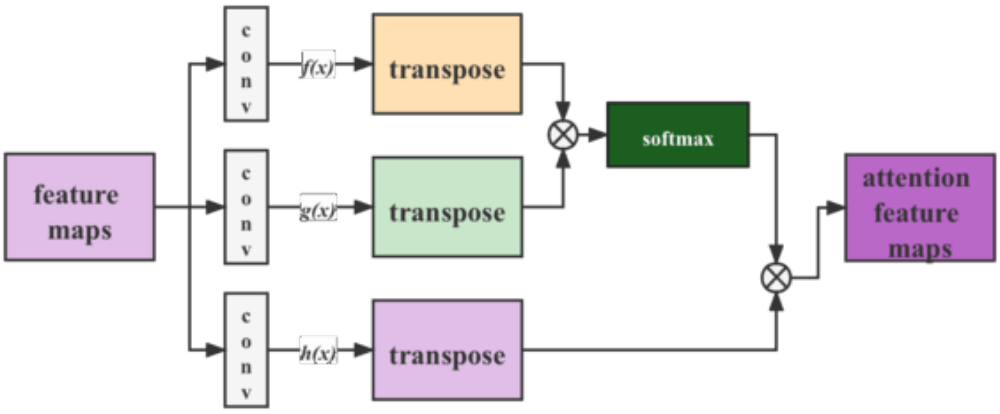

Figure 5. Flow chart of SAGAN.

Figure 6 shows the visualization of the feature maps and attention feature maps optimized by post-GAN.

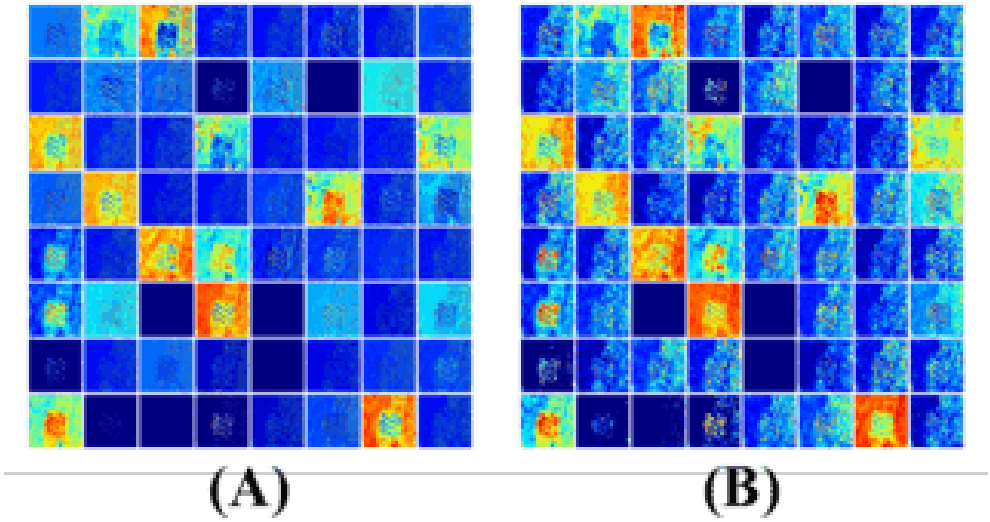

Figure 6. Comparison of feature maps with attention feature maps. (**A**) Feature maps; (**B**) attention feature maps optimized by post-GAN.

This Multi-GANs module enhances the detection network's robustness by adding asymmetric Multi-GANs detection network branches to regularize the results. In order to prevent the gradient from disappearing in the convolutional layer for inputs data more minor than zero, we changed the activation function for each block in the original model from ReLU to LeakyReLU. Meanwhile, since instance normalization layer works more satisfactorily on generative tasks than batch normalization layer, the batch normalization layers of Multi-GANs module was replaced with instance normalization layers.

3.2.2. Transformer

In 2020, the transformer achieved extraordinary classification, detection, and segmentation results. However, the drawbacks are apparent:

1. Its training time is exceedingly long;
2. It is not conducive to deployment acceleration;
3. It requires a vast dataset;

Therefore, in this paper, we refer to the idea of the transformer and design it as a branch network, which exploits its ability to extract global features and relies on the CNN backbone, avoiding its training time from being too prolonged.

As Figure 3 depicts, CNN backbones still utilize the feature pyramid structure. Moreover, the feature map's resolution decreases as the depth of the network increases while the number of channels increases. The transformer branch network is responsible for providing global features to the backbone. First, the input image is divided into patches, and then mainly undertake transformation for each patch as a flattening operation. For instance, assuming the input image size is 256×256, if it is divided into 64 patches, each is 32×32 in size. The original transformer encoder is composed of alternating multi-heads self-attention and multi-layer perceptron. Nevertheless, in this paper, to reduce the number of parameters and training time, this part is transformed into the identical mechanism as the attention module in the backbone, i.e., attention based on post-GAN optimization. After being processed through the layer norm layer, all the features are pooled and sent to the CNN backbone.

3.2.3. Loss Function

The Multi-GANs detection network's loss function is composed of three portions: regression box loss, $CIoU$ loss, and classification loss. The calculation process is shown in Equations (5)–(8). Box coordinate error (x_i, y_i) denotes the predicted box's center position coordinate, and (w_i, h_i) is its width and height. $(\hat{x}_i, \hat{y}_i)$ and $(\hat{w}_i, \hat{h}_i)$ denote coordinates and size of the labeled ground truth box, respectively. Furthermore, λ_{coord} and λ_{noobj} are constants. $K \times K$ represents the grids' amount. M expounds the predicted boxes' overall amount. Besides, I_{ij}^{obj} is one when the ith grid detects a target and zero otherwise.

$$Loss = Loss_{bounding\_box} + Loss_{ciou} + Loss_{classification} \tag{5}$$

$$\begin{aligned} Loss_{bounding\_box} =& \lambda_{coord} \sum_{i=0}^{K\times K} \sum_{j=0}^{M} I_{ij}^{obj} (2 - w_i \times h_i)[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \\ & \lambda_{coord} \sum_{i=0}^{K\times K} \sum_{j=0}^{M} I_{ij}^{obj} (2 - w_i \times h_i)[(w_i - \hat{w}_i)^2 + (h_i - \hat{h}_i)^2] \end{aligned} \tag{6}$$

$$\begin{aligned} Loss_{ciou} =& \sum_{i=0}^{K\times K} \sum_{j=0}^{M} I_{ij}^{obj} [\hat{C}_i log(C_i) + (1 - \hat{C}_i log(1 - C_i))] + \\ & \lambda_{noobj} \sum_{i=0}^{K\times K} \sum_{j=0}^{M} I_{ij}^{noobj} [\hat{C}_i log(C_i) + (1 - \hat{C}_i log(1 - C_i))] \end{aligned} \tag{7}$$

$$Loss_{classification} = \sum_{i=0}^{K\times K} I_{ij}^{obj} \sum_{c\in classes} [\hat{p}_i(c) log(p_i(c)) + (1 - \hat{p}_i(c) log(1 - p_i(c))] \tag{8}$$

In the training process, pred_bbox was divided into positive and negative examples. For any ground truth, calculate IoU with all pred_bboxes, and the largest IoU was the positive example. One pred_bbox could only be assigned to one ground truth. For instance, the first ground truth matched the pred_bbox of a positive example, then the following ground truth needed to find the largest IoU among the remaining pred_bboxes as a positive example. Except for the positive examples, if the IoU with all ground truth was less than the threshold, it was negative. Prediction boxes that were neither positive nor negative were discarded.

In this way, the loss function could reduce the weight of easy-to-classify samples so that the model could focus more on difficult-to-classify samples during training. Through this improvement, the network's accuracy could be promoted while the inference speed of the network was maintained.

It could be inferred from Equation (9),

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{9}$$

if the two prediction boxes do not intersect, their IoU value was 0. Then this value could not reflect the distance between the two, which was, the degree of coincidence. At the same time, the corresponding loss was 0, and the gradient of backpropagation was 0, and learning and training could not be performed. In the CVPR2019 paper [62], GIoU is proposed, and its calculation is shown in Equation (10),

$$GIoU = IoU - \frac{|A_c - U|}{|A_c|} \tag{10}$$

where A_c represents the smallest rectangular area that contains both the prediction frame and ground truth. In the above formula, it could be inferred that when the prediction frame completely covers the ground truth, $GIoU$ could not well reflect the coincidence of the two. In order to consider the distance and overlap rate at the same time, $DIoU$ [63] is proposed, and its calculation process is shown in Equation (11),

$$DIoU = IoU - \frac{\rho^2(\boldsymbol{b}, \boldsymbol{b}^{gt})}{c^2} + \alpha\nu \tag{11}$$

where $\boldsymbol{b}$ and $\boldsymbol{b}^{gt}$ represent the center points of the prediction frame and ground truth, respectively, and ρ represents the Euclidean distance between these two center points, and c represents the diagonal distance of the smallest rectangle that could simultaneously contain the prediction frame and ground truth. However, because the expression method does not consider the aspect ratio of the outer frame, based on $DIoU$, $CIoU$ is proposed [63], which is the measurement method used in the loss function in this paper, and the penalty term is shown in Equation (12).

$$\mathcal{R}_{CIoU} = \frac{\rho^2(\boldsymbol{b}, \boldsymbol{b}^{gt})}{c^2} + \alpha\nu \tag{12}$$

where α is the weight function, and ν, defined as $\nu = \frac{4}{\pi^2}(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h})^2$, is used to measure the similarity of the aspect ratio. The gradient of $CIoU$ loss is similar to $DIoU$. Moreover, when the length and width are in $[0, 1]$, the value of $w^2 + h^2$ is usually tiny, which leads to the explosion of the gradient. So when it comes to the implementation, $\frac{1}{w^2+h^2}$ is replaced with 1. The loss function of $CIoU$ is defined as shown in Equation (13).

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(\boldsymbol{b}, \boldsymbol{b}^{gt})}{c^2} + \alpha\nu \tag{13}$$

3.2.4. Fusion Method for Bounding Boxes

We proposed a new fusion algorithm for bounding boxes that gave up the NMS solution of removing bounding boxes with low confidence and adopted the method of fusing different bounding boxes of the same object. In fact, the weight coefficients s was introduced in the fusion process, and the weight coefficients of each bounding box are calculated as shown in Equation (14).

$$C_s = \alpha \times A_s + (1 - \alpha) \times B_s \tag{14}$$

The $alpha$ represents the sub-network weights of the generative sub-network, and by adjusting the size of α, we can control the degree of influence that the generative sub-network on the main detection network; $(x_1, y_1), (x_2, y_2)$ represent the top left and bottom right coordinates of a box, respectively; c represents the confidence level of a bounding box. So, the higher the c box is, the larger s is, and it contributes more in the process of generating a new box. The shape and position of the new box is closer towards boxes with larger weights.

$$C_{x_1} = \frac{A_{x_1} \times A_s + B_{x_1} \times B_s}{A_s + B_s} \tag{15}$$

$$C_{y_1} = \frac{A_{y_1} \times A_s + B_{y_1} \times B_s}{A_s + B_s} \tag{16}$$

$$C_{x_2} = \frac{A_{x_2} \times A_s + B_{x_2} \times B_s}{A_s + B_s} \tag{17}$$

$$C_{y_2} = \frac{A_{y_2} \times A_s + B_{y_2} \times B_s}{A_s + B_s} \tag{18}$$

Equations (15)–(18) show how to get the fused bounding box C by using two bounding box A and B.

4. Experiment

4.1. Dataset Analysis

As illustrated in Figure.1, the aerial image dataset used in this paper has the following characteristics:

1. The dataset contains a relatively large number of detection targets. Most of the images possess more than one detection target, and some of them, such as cars, represent a tiny proportion of the overall image.
2. The samples in the dataset are distributed unevenly. To be more specific, the number of bridge samples is 4.5 times higher than that of baseball field samples.
3. The overall data volume is small, making deep learning training rather tricky.

4.2. Evaluation Metrics

To validate the model's performance, four metrics are used for the evaluation in this paper, namely, *mAP*, Precision (*P*), Recall (*R*), and FPS. The Jaccard index, commonly known as the intersection over union (*IoU*), is specified as the intersection of predicted segmentation, which also divides the label. The value of this indicator ranges from 0 to 1: 0 indicates no overlap, and 1 represents complete overlap. It is a true situation when the $IoU \geq$ is 0.5; otherwise, it is a false positive situation. The binary classification calculation formula is:

$$IoU = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{TP + FP + FN} \tag{19}$$

where A denotes ground truth and B is predicted segmentation.

Pixel accuracy (*PA*) is the percentage of an image's accurately classified pixels, as known as, the proportion of correctly classified pixels to entire pixels. The formula is as follows:

$$PA = \frac{\sum_{i=0}^{n} p_{ii}}{\sum_{i=0}^{n} \sum_{j=0}^{n} p_{ij}} = \frac{TP + TN}{TP + TN + FP + FN} \tag{20}$$

n indicates the total amount of categories, $n + 1$ represents the category amount, containing backdrops. p_{ii} indicates the overall real pixels' amount, in which the label is i and predicted to be class i. That is, the entire amount of matched pixels for real pixels (class i). p_{ij} expounds the overall amount of real pixels (label i) that are predicted to be class j, which can be regarded as pixels' amount (label i) that are classified into class j incorrectly. Moreover, TP symbolizes true positives' amount (positive in both labels and predicted value). TN expounds the amount of true negatives (negative in both labels and predicted value). FP is the amount of false positives (negative in label and positive in predicted value). FN describes the amount of false negatives (positive in label and negative in predicted value). In addition, $TP + TN + FP + FN$ specifies the overall pixels' amount, $TP + TN$ specifies pixels' amount correctly classified.

Mean pixel accuracy (*mPA*) is a straightforward improvement on *PA*. *mPA* computes the percentage of pixels precisely recognized in every class and average the outcomes, as indicated in Equation (21).

$$mAP = \frac{\sum_{i=1}^{k} (AP_i)}{k} \tag{21}$$

Precision (*P*) is the percentage of samples categorized as positive samples in the accurately classified samples.

$$P = \frac{TP}{TP + FP} \tag{22}$$

Recall (R) demonstrates the percentage of correctly categorized positive samples in overall positive samples.

$$R = \frac{TP}{TP + FN} \tag{23}$$

4.3. Experiment Setting

A personal computer (CPU: Intel(R) i9-10900KF; GPU: NVIDIA RTX 3080 10 GB; Memory: 16 GB; OS: Ubuntu 18.04, 64 bits) was used to carry out the entire model training and validation process. We chose the Adam optimizer with an initial learning rate, $a_0 = 1e^{-4}$. The learning rate increment was adjusted using the method specified in Section 4.5 and the training speed is optimized.

4.4. Label Smoothing

Usually, there are a small number of mislabels in machine learning samples, which can affect the prediction effect, especially when the sample size is small. Therefore, in this paper, we adopted the label smoothing technique to improve the situation, which is based on the following solution: to avoid 'over-trusting' the labels of training samples by assuming that some of the labels may be incorrect at the time of training.

At each iteration, instead of inputting (x_i, y_i) directly into the training set, an error rate ϵ is set, and (x_i, y_i) is substituted into the training with probability $1 - \epsilon$, and $(x_i, 1 - y_i)$ is substituted into the training with probability ϵ. In this way, the model is trained with both correct and incorrect label inputs, and it is conceivable that a model so trained will not match every label 'to the fullest extent', but only to a certain extent. This way, the model will be less impacted if there are indeed incorrect labels.

When we use cross-entropy to describe the loss function, for each sample i, the loss function is:

$$Loss_i = -y_i \times P(\hat{y}_i = 1|x_i) - (1 - y_i) \times P(\hat{y}_i = 0|x_i) \tag{24}$$

After randomization, the new labels have the same probability of $1 - \epsilon$ as y_i and a different probability of ϵ, i.e. $1 - y_i$. Therefore, when the randomized labels are used as training data, the loss function has the same probability of $1 - \epsilon$ as the above equation, and the probability of ϵ as:

$$Loss_i = -(1 - y_i) \times P(\hat{y}_i = 1|x_i) - y_i \times P(\hat{y}_i = 0|x_i) \tag{25}$$

After weighted averaging Equations (24) and (25) by probability, having $y'_i = \epsilon \times (1 - y_i) + (1 - \epsilon) \times y_i$, we can obtain:

$$Loss_i = -(1 - y'_i) \times P(\hat{y}_i = 1|x_i) - y'_i \times P(\hat{y}_i = 0|x_i) \tag{26}$$

Compared with the original cross-entropy expression, only y_i is replaced with y'_i, while everything else remains the same. This is actually equivalent to replacing each label y_i with y'_i and then performing the regular training process. Therefore, in this paper, randomization was not conducted before training except replacing each label accordingly.

4.5. Training Strategy

Warm-up [64] is a training strategy. The exp warm-up method is examined in this article, which involves linearly accelerating the learning from a minuscule value to the predefined learning speed and then fading in terms of the exp function law. This paper also tried cos warm-up. According to the cos function law, the learning rate increased linearly from a minimal value to a preset value and then decayed. The principle of cosine decay is shown in Equation (27).

$$\eta_t = \eta^i_{min} + \frac{1}{2}(\eta^i_{max} - \eta^i_{min})(1 + cos(\frac{T_{cur}}{T_i}\pi)) \tag{27}$$

Among it, i represents the number of iterations, η^i_{max} and η^i_{min} represent the maximum and minimum values of the learning rate, respectively, T_{cur} represents the number of epochs currently executed. In contrast, T_i represents the overall number of epochs in the number i step.

5. Result

In this section, the model introduced in Section 3.2 was implemented for object detection in aerial images. We trained the datasets with three input sizes, 300 × 300, 512 × 512, and 608 × 608.

5.1. Validation Results

Table 1 illustrates the statistical results. The best results of the index are marked red. In Table 1, YOLO v5 [30] has the best speed, with *FPS* reaching 60.3. The *P*, *R*, and *mAP* of Faster-RCNN (when the input is 300 × 300) are 82.87%, 78.32%, 90.13%, which are the worst performances of all models. These *P*, *R*, and *mAP* of YOLO v5 are the most excellent among all the YOLO series. Meanwhile, they are superior to those of the Faster-RCNN and SSD series (0.9446, 0.9718, 0.9674, respectively). The most similar model to YOLO v5 is RefineDet, with *P*, *R*, and *mAP* of 94.91%, 98.49%, and 96.97%, respectively (when the input is 512 × 512). Nevertheless, its inference speed is significantly lower than YOLO v5, reaching only 42% of the latter. That is probably due to the stronger performance of the attention extraction module in YOLO v5. Overall, the YOLO series are the best models among the comparisons. We split the input into two transmission GANsformers of 300 × 300 and 512 × 512 for testing, and the results show that the former has better performance, with the three parameters reaching 96.77%, 98.83%, 97.91%, respectively. Moreover, its *precision* and *mAP* are superior to all the other comparison models. However, our model is not superior in inference speed, only 53% of YOLO v5. The complexity of the Multi-GANs module causes it. As depicted above, the GANsformer detection network reflects the best detection performance on the validation set, according to the results.

Table 1. Comparisons of different detection networks' performance (in %).

Model	Input Size	Precision	Recall	mAP	FPS
SSD	300 × 300	83.96	80.23	87.64	33.7
	512 × 512	86.43	86.26	91.27	32.3
FSSD	300 × 300	89.76	94.37	94.85	32.9
	512 × 512	93.75	96.89	96.31	32.2
RefineDet	300 × 300	94.34	98.28	96.81	27.8
	512 × 512	94.91	98.49	96.97	25.3
EfficientDet L2	300 × 300	92.10	95.33	94.98	20.8
	512 × 512	93.24	95.98	95.14	20.2
Faster RCNN	300 × 300	82.87	78.32	90.13	25.0
	512 × 512	85.29	76.91	92.20	46.7
YOLO v3	608 × 608	94.92	98.43	96.93	52.1
YOLO v4	608 × 608	94.38	98.51	97.42	57.5
YOLO v5	608 × 608	95.98	98.57	97.51	60.3
ours	300 × 300	96.77	98.83	97.91	32.2
	512 × 512	96.45	98.86	97.50	30.4

5.2. Detection Results

For further comparison, we extracted four images from the aerial image series of test set. These images show as many detection scenarios as possible in the dataset, such as scenarios with multiple detected objects, scenarios where the image color channel is heavily distorted, and scenarios where the detected objects are too small and sparse. Figures 7–16 show the detection results. Figure 7 denotes ground truth; the red boxes in the rest of the images denote the predicted bounding box. It can be witnessed that Faster-RCNN performs very poorly in these four images, while EfficientDet, SSD series,

and YOLO series perform relatively well and detect lesions accurately. However, when the detected objects are too tiny, all model performances decrease, and part of the models even have some unlabeled detected objects. This situation is probably related to the attention extraction module in these networks.

Our model outperforms the previous models by highly accurate object detection, even when detecting moderately dense objects. Although there is still room for improvement, it has outperformed other models. On the one hand, we augment the image with the WGAN model before it is fed into the backbone. On the other hand, we add the SAGAN model to the attention extraction module of the model, which can significantly improve the model's robustness.

Figure 7. The ground truth in the dataset.

Figure 8. The detection results of YOLO v3 in the dataset.

Figure 9. The detection results of YOLO v4 in the dataset.

Figure 10. The detection results of YOLO v5 in the dataset.

Figure 11. The detection results of SSD in the dataset.

Figure 12. The detection results of FSSD in the dataset.

Figure 13. The detection results of RefineDet in the dataset.

Figure 14. The detection results of EfficientDet L2 in the dataset.

Figure 15. The detection results of Faster RCNN in the dataset.

Figure 16. The detection results of GANsformer detection network in the dataset.

As can be seen from Figure 16, our model performs very well when detecting an aircraft and car, especially when shadows are included. It indicates that the GANsformer detection network can effectively improve the model robustness.

5.3. Results Analysis

As Section 5.2 depicted, the detection of objects at different scales and densities has different outcomes. Table 2 displays the performance of the GANsformer when detecting the below objects in the dataset.

Table 2. The detection capability of the GANsformer detection network in each object category (in %).

Object	Precision	Recall	mAP
Bridge	97.35	99.01	98.59
Baseball Field	96.21	98.19	97.32
Basketball Court	96.13	98.60	97.03
Airplane	96.27	98.59	97.28
Track and Field	97.29	98.04	98.55
Oil tank	96.18	98.83	97.89
Tennis Field	97.21	98.89	98.37
Port	96.99	98.89	97.87
Ship	97.23	98.81	97.56
Car	96.91	98.95	97.90

As illustrated in Table 2, our model performs well when recognizing comparatively large scale objects, such as bridges, tracks, and fields, which are often not dense and have discerned and clear boundaries in the image. For small-scale objects, such as aircraft and cars, the detection error is relatively high, as shown in Figure 16.

6. Discussion

6.1. Ablation Experiment of Multi-GANs

This paper uses pre-GAN in backbone and attention extraction modules, while GAN models have many branches and focus. The primary purpose of the pre-GAN module in front of the backbone is to enhance the model input. In contrast, the post-GAN module in the attention extraction module generates an attention mask to enhance the model's robustness. Therefore, for the two GAN modules with different purposes, different GAN models are implemented in this paper, including WGAN, BAGAN, SAGAN, and SPA-GAN. Several ablation experiments are conducted, and the experimental results are shown in Table 3.

Table 3. Results of different implements of Multi-GANs (in %).

Method	Precision	Recall	mAP	FPS
no GAN (baseline)	94.17	95.22	94.39	47.9
WGAN + SAGAN	96.06	97.69	97.13	34.3
BAGAN + SAGAN	95.18	97.19	96.98	34.1
WGAN + SPA-GAN	96.77	98.83	97.91	32.2
BAGAN + SPA-GAN	96.38	98.55	97.20	32.2

Table 3 reflects that using WGAN and SPA-GAN to implement pre-GAN and post-GAN, respectively, can optimize the model performance, with the three primary metrics reaching 96.77%, 98.83%, and 97.91%. As a comparison, WGAN is better than BAGAN in the choice of pre-GAN. Regardless of the implementation of the post-GAN, this is probably because BAGAN uses a different formula from WGAN in computing the difference between the generated data and the original data, failing to maximize the generator's and discriminator's performances. By comparing the baseline model, it is apparent that the Multi-GANs module, regardless of the implementation approaches, can significantly improve the model's performance by 3.52% in terms of mAP parameters. Furthermore, we tried to visualize the mask of noise generated by two post-GANs, as shown in Figure 17.

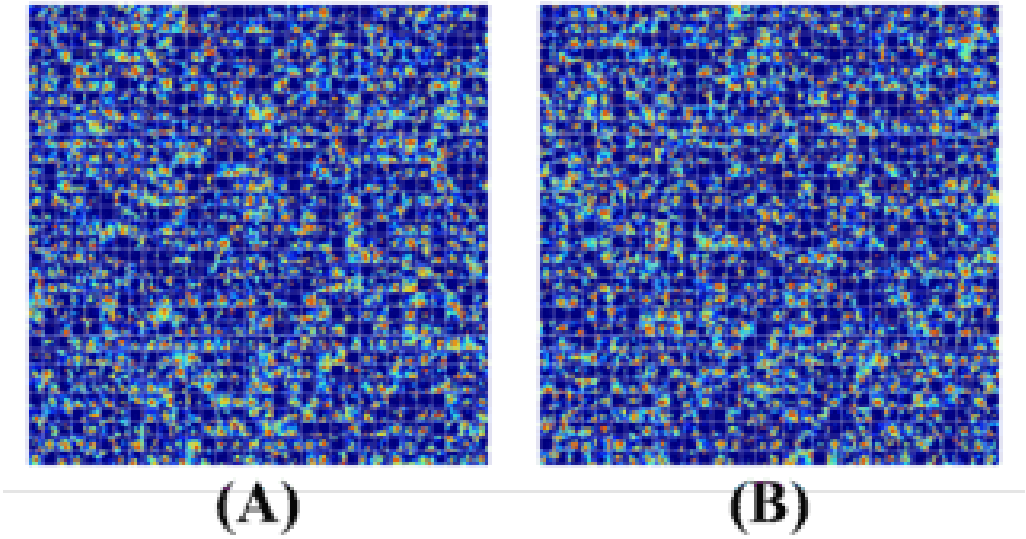

Figure 17. Illustration of noise mask by two post-GANs (**A**,**B**).

As can be seen from Figure 17, the feature maps generated by the two post-GAN implementations differ significantly, but the generated highlights are roughly the same. Because noise mask acts on the feature maps, it is not straightforward to be understood in the conventional way humans think and read. However, it can effectively improve the model's performance, and the noise generation area is roughly the same. We believe that the post-GAN module can add noise to the object area and thereby enhance the model's robustness.

6.2. Ablation Experiment of Data Augmentation Methods

To verify the effectiveness of the various pre-processing methods proposed in Section 3.1.2, the ablation experiments were performed on the GANsformer detection network, selected from the above experiments with the best performance. The experimental results are shown in Table 4.

Table 4. Ablation experiment result of different pre-processing methods on the GANsformer detection network (in %).

MixUp	CutOut	CutMix	SnapMix	Mosaic	Precision	Recall	mAP
✓	✓	✓	✓	✓	96.77	98.86	97.91
✓	✓	✓		✓	96.21	98.51	97.82
✓	✓		✓		96.53	98.82	97.69
✓		✓		✓	96.76	98.89	97.93
	✓	✓	✓	✓	96.23	98.56	97.85

Through the analysis of experimental results, we can find those augmentation methods are of great assistance in improving the performance of the GANsformer detection network. The principles of SnapMix and Mixup are similar. It could be seen that the model performs best when Mixup, CutMix, and Mosaic methods are used simultaneously.

6.3. Validation on Wheat Head Dataset

In this section, we validate the generalization capability of the proposed model GANsformer. We undertake it on the wheat dataset [65]. Figure 18 illustrates the concerning experimental results. The experimental results show that the proposed method still has good performance on the wheat dataset and ascertains that the suggested method can indeed enhance the network's ability to extract image features.

Figure 18. Detection results on wheat dataset.

6.4. Detection Application on macOS

In order to make the proposed method feasible and viable in a macOS platform, this paper wraps the model based on Swift language and develops a macOS application. The main functions include: (1) importing the images to be annotated and completing the detection task at one time; (2) selecting two pruning strategies; (3) saving the recognition results to a CSV file. The primary functions of this application are shown in Figure 19.

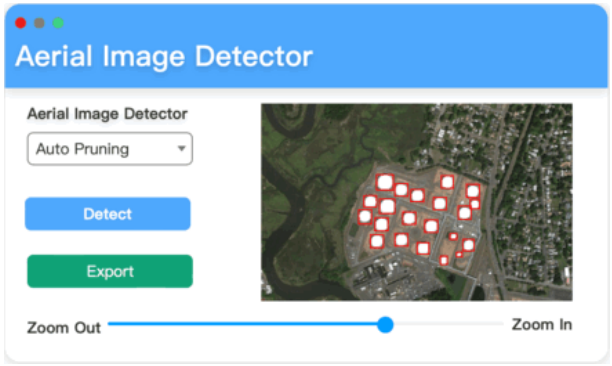

Figure 19. Screenshot of application based on GANsformer detection network.

6.5. Limitation

With the rapid proliferation of UAVs and the increasing pixel quality of filming devices, the small object detection accuracy of the aerial image is still promising. Even though our algorithm delivers the best segmentation effect, it is still far from the ideal accuracy. As shown in Figure 16, our model's errors are pronounced in the task of small-scale object detection, so we analyzed the causes of these errors and their improvement methods.

1. Scarcity of datasets. The dataset is inadequate, and the substantial difference in the sample size of different categories in the dataset are fundamental reasons for the unsatisfactory performance of the model. The dataset used in this paper contains 800 images with ten categories of targets, of which 150 are pure background images with no target. The bridge category accounts for 26% of the ten categories of objects. Although this paper uses various data augmentation methods and Multi-GANs to expand the data, the imbalance among samples in the dataset is still not completely solved. One viable solution to the above problem is to set a parameter α in the loss function construction, whose value is inversely proportional to the percentage of each category in the total dataset. Specifically, the smaller the percentage of the category in the loss function, the more critical it is to balance the sample size gap between categories.
2. Drawbacks of the one-stage structure. Considering the fusion trend of one-stage and two-stage models and advantages of a one-stage network in terms of the model's inference speed and accuracy, this paper constructs a one-stage-based detection network. However, the one-stage network model has its intrinsic shortcomings, and the feature extraction ability of the backbone still needs to be improved. Although the backbone's ability to process feature maps has been improved by applying the attention extraction

module of Multi-GANs, the distance between the shallowest and deepest networks is getting more considerable, and more information is lost as the number of layers of the network increases. The effectiveness of feature maps fusion is also decreasing gradually.

3. The definition of the loss function is still deficient. The first point apparently indicates that the definition of the loss function can still be improved to include more information and balance the imbalance between the samples in the dataset.

7. Conclusions

Object detection in aerial images is a classic topic in computer vision research. Nonetheless, impediments in detecting small objects in aerial images should not be neglected, i.e., (1) aerial images with variable altitudes and illuminance conditions in which small objects are dense, increasing detection challenges and causing low detection accuracy. (2) The actual weather will affect capturing of aerial images. Sometimes there will be interference factors, such as flying birds, insects, or leaves. Additionally, conventional approaches in this research area are not outstanding enough: (1) traditional machine learning methods typically focus on inference speed but low accuracy. (2) When using CNNs, there are demerits in the case of insufficient samples, such as instability, non-convergence, and overfitting. (3) Due to the intrinsic drawbacks of conventional CNN models and transformer structures, such as the lack of global features, extremely long training time, and massive dataset required, these two techniques are not excellent enough to be solely applied in this study's research field.

Therefore, this paper proposes a GANsformer detection network based on a one-stage detection network, aiming to address these above-mentioned problems. In this paper, we used a dataset, containing 800 aerial images, with ten types of targets to be detected: aircraft, ships, oil tanks, baseball fields, basketball courts, athletic fields, ports, bridges, tennis fields, and vehicles, as shown in Figure 1.

The following demonstrates primary innovations of the GANsformer detection network proposed in this paper:

1. Multi-GANs model: first and foremost, a generative model is added in front of the backbone to expand the input aerial images, which aims to alleviate the general problem of small sample size datasets. Second, GAN models are added to the attention extraction module to generate attention masks. Figure 6 shows the effect of adding GAN models on feature maps, and the results of the experimental part also illustrate that this approach can effectively improve the robustness of the model. Ultimately, on the validation set, the proposed method reaches 96.77%, 98.86% and 97.91% on *precision*, *recall*, and mAP, respectively. This experimental result demonstrates that the proposed model outperforms all the comparison models.
2. We modified the transformer, by reducing the number of parameters, improving the training speed, to improve CNN's ability to capture global features as a branch network. The performance of the GANsformer—because it inherits and combines the structural and global feature extraction advantages of CNN and visual transformers—is significantly better than CNN and vision transformer, with comparable parameter complexity, showing great potential capability in aerial image detection tasks.
3. In order to verify the effectiveness of various implementations of Multi-GANs, in Section 6, we tested the performance of different combinations of generative models. Experimental results expound that the SPA-GAN model performs best in the attention extraction module, while in image augmentation, WGAN performs best.
4. This paper encapsulated the model and developed a corresponding application under the macOS platform, making the model applicable.

Although the proposed model has surpassed the comparison model, limitations still exist. Although this paper has used various methods to improve the imbalance problem of small sample datasets, the detection accuracy is still the worst among all classes in terms

of detection results. Moreover, the proposed detection network is based on one-stage, but the one-stage network has its inherent defects. Eventually, the design of the loss function can still be potentially improved.

Based on the shortcomings proposed in Section 6.5, the authors of this paper will work on redesigning the model's loss function in the future to address the imbalance of the dataset and to further optimize the model from the perspective of a loss function design.

Ultimately, this paper encapsulates the model and develops a corresponding application under the macOS platform, making the model applicable.

Author Contributions: Conceptualization, Y.Z.; methodology, Y.Z.; validation, Y.Z.; writing—original draft preparation, Y.Z.; writing—review and editing, Y.Z., S.W., S.C. and Q.M.; visualization, Y.Z.; supervision, Y.Z. and Q.M.; project administration, Y.Z. and X.L.; funding acquisition, Q.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Provincial Natural Science Foundation Project grant number ZR2021MC099. This work was supported by the Major S&T project (Innovation 2030) of China under grant 2021ZD0113701.

Acknowledgments: We are grateful to the Edison Coding Club of CIEE at China Agricultural University for their strong support during our thesis writing.

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

References

1. Eikelboom, J.A.; Wind, J.; Van de Ven, E.; Kenana, L.M.; Schroder, B.; de Knegt, H.J.; van Langevelde, F.; Prins, H.H. Improving the precision and accuracy of animal population estimates with aerial image object detection. *Methods Ecol. Evol.* **2019**, *10*, 1875–1887. [CrossRef]
2. Xiao, Z.; Wang, K.; Wan, Q.; Tan, X.; Xu, C.; Xia, F. A2S-Det: Efficiency Anchor Matching in Aerial Image Oriented Object Detection. *Remote. Sens.* **2021**, *13*, 73. [CrossRef]
3. Chen, C.; Zhong, J.; Tan, Y. Multiple-oriented and small object detection with convolutional neural networks for aerial image. *Remote. Sens.* **2019**, *11*, 2176. [CrossRef]
4. Wang, Y.; Zorzi, S.; Bittner, K. Machine-learned 3D Building Vectorization from Satellite Imagery. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 1072–1081.
5. Abbasi, S.; Rezaeian, M. Visual object tracking using similarity transformation and adaptive optical flow. *Multimed. Tools Appl.* **2021**, *volume*, 1–19. [CrossRef]
6. Liu, M.; Wang, X.; Zhou, A.; Fu, X.; Ma, Y.; Piao, C. UAV-YOLO: Small object detection on unmanned aerial vehicle perspective. *Sensors* **2020**, *20*, 2238. [CrossRef] [PubMed]
7. Zhang, W.; Tang, P.; Zhao, L. Remote sensing image scene classification using CNN-CapsNet. *Remote. Sens.* **2019**, *11*, 494. [CrossRef]
8. Pham, M.T.; Courtrai, L.; Friguet, C.; Lefèvre, S.; Baussard, A. YOLO-Fine: One-stage detector of small objects under various backgrounds in remote sensing images. *Remote. Sens.* **2020**, *12*, 2501. [CrossRef]
9. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
10. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
11. He, J.; Deng, Z.; Zhou, L.; Wang, Y.; Qiao, Y. Adaptive pyramid context network for semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7519–7528.
12. Nguyen, N.D.; Do, T.; Ngo, T.D.; Le, D.D. An evaluation of deep learning methods for small object detection. *J. Electr. Comput. Eng.* **2020**, *2020*, 3189691. [CrossRef]
13. Hu, G.X.; Yang, Z.; Hu, L.; Huang, L.; Han, J.M. Small object detection with multiscale features. *Int. J. Digit. Multimed. Broadcast.* **2018**, *2018*, 4546896. [CrossRef]
14. Liu, C.; Wu, Y.; Liu, J.; Han, J. MTI-YOLO: A Light-Weight and Real-Time Deep Neural Network for Insulator Detection in Complex Aerial Images. *Energies* **2021**, *14*, 1426. [CrossRef]
15. Courtrai, L.; Pham, M.T.; Lefèvre, S. Small Object Detection in Remote Sensing Images Based on Super-Resolution with Auxiliary Generative Adversarial Networks. *Remote. Sens.* **2020**, *12*, 3152. [CrossRef]
16. 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]
17. Xu, D.; Wu, Y. Improved YOLO-V3 with DenseNet for multi-scale remote sensing target detection. *Sensors* **2020**, *20*, 4276. [CrossRef] [PubMed]

18. Avola, D.; Cinque, L.; Diko, A.; Fagioli, A.; Foresti, G.L.; Mecca, A.; Pannone, D.; Piciarelli, C. MS-Faster R-CNN: Multi-stream backbone for improved Faster R-CNN object detection and aerial tracking from UAV images. *Remote. Sens.* **2021**, *13*, 1670. [CrossRef]
19. Jin, R.; Lv, J.; Li, B.; Ye, J.; Lin, D. Toward efficient object detection in aerial images using extreme scale metric learning. *IEEE Access* **2021**, *9*, 56214–56227. [CrossRef]
20. Fujiyoshi, H.; Hirakawa, T.; Yamashita, T. Deep learning-based image recognition for autonomous driving. *IATSS Res.* **2019**, *43*, 244–252. [CrossRef]
21. Sim, H.S.; Kim, H.I.; Ahn, J.J. Is deep learning for image recognition applicable to stock market prediction? *Complexity* **2019**, *2019*, 4324878. [CrossRef]
22. Hatt, M.; Parmar, C.; Qi, J.; El Naqa, I. Machine (deep) learning methods for image processing and radiomics. *IEEE Trans. Radiat. Plasma Med Sci.* **2019**, *3*, 104–108. [CrossRef]
23. Ann, E.T.L.; Hao, N.S.; Wei, G.W.; Hee, K.C. Feast In: A Machine Learning Image Recognition Model of Recipe and Lifestyle Applications. *MATEC Web Conf. EDP Sci.* **2021**, *335*, 04006. [CrossRef]
24. Gu, H.; Wen, F.; Wang, B.; Lee, A.K.; Xu, D. *Machine Learning-Based Image Recognition for Visual Inspections*; SNAME Maritime Convention: Tacoma, WA, USA, 2019.
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, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
26. Li, Z.; Zhou, F. FSSD: Feature fusion single shot multibox detector. *arXiv* **2017**, arXiv:1712.00960.
27. 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.
28. 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.
29. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
30. Jocher, G. Yolov5. 2020. Available online: https://github.com/ultralytics/yolov5 (accessed on 17 January 2022).
31. 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.
32. Zhang, Y.; Wa, S.; Liu, Y.; Zhou, X.; Sun, P.; Ma, Q. High-Accuracy Detection of Maize Leaf Diseases CNN Based on Multi-Pathway Activation Function Module. *Remote. Sens.* **2021**, *13*, 4218. [CrossRef]
33. Zhang, Y.; He, S.; Wa, S.; Zong, Z.; Liu, Y. Using Generative Module and Pruning Inference for the Fast and Accurate Detection of Apple Flower in Natural Environments. *Information* **2021**, *12*, 495. [CrossRef]
34. Zhang, Y.; Wa, S.; Sun, P.; Wang, Y. Pear Defect Detection Method Based on ResNet and DCGAN. *Information* **2021**, *12*, 397. [CrossRef]
35. Wu, H.; Zhang, J.; Huang, K.; Liang, K.; Yu, Y. Fastfcn: Rethinking dilated convolution in the backbone for semantic segmentation. *arXiv* **2019**, arXiv:1903.11816.
36. Wang, B.; Lei, Y.; Tian, S.; Wang, T.; Liu, Y.; Patel, P.; Jani, A.B.; Mao, H.; Curran, W.J.; Liu, T.; et al. Deeply supervised 3D fully convolutional networks with group dilated convolution for automatic MRI prostate segmentation. *Med. Phys.* **2019**, *46*, 1707–1718. [CrossRef]
37. Li, X.; Shen, X.; Zhou, Y.; Wang, X.; Li, T.Q. Classification of breast cancer histopathological images using interleaved DenseNet with SENet (IDSNet). *PLoS ONE* **2020**, *15*, e0232127. [CrossRef]
38. Wang, S.H.; Fernandes, S.; Zhu, Z.; Zhang, Y.D. AVNC: Attention-based VGG-style network for COVID-19 diagnosis by CBAM. *IEEE Sensors J.* **2021**. [CrossRef]
39. Chen, L.; Tian, X.; Chai, G.; Zhang, X.; Chen, E. A New CBAM-P-Net Model for Few-Shot Forest Species Classification Using Airborne Hyperspectral Images. *Remote. Sens.* **2021**, *13*, 1269. [CrossRef]
40. Cai, W.; Wang, Y.; Ma, J.; Jin, Q. Can: Effective cross features by global attention mechanism and neural network for ad click prediction. *Tsinghua Sci. Technol.* **2021**, *27*, 186–195. [CrossRef]
41. Wu, T.; Ku, T.; Zhang, H. Research for image caption based on global attention mechanism. In *Proceedings of the Second Target Recognition and Artificial Intelligence Summit Forum*; International Society for Optics and Photonics: Bellingham, WA, USA, 2020; Volume 11427, p. 114272.
42. Gan, X.; Wang, L.; Chen, Q.; Ge, Y.; Duan, S. GAU-Net: U-Net Based on Global Attention Mechanism for brain tumor segmentation. *J. Physics Conf. Ser.* **2021**, *1861*, 012041. [CrossRef]
43. Bello, I.; Zoph, B.; Vaswani, A.; Shlens, J.; Le, Q.V. Attention augmented convolutional networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 3286–3295.
44. Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; et al. A survey on visual transformer. *arXiv* **2020**, arXiv:2012.12556.
45. Sajid, U.; Chen, X.; Sajid, H.; Kim, T.; Wang, G. Audio-visual transformer based crowd counting. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 2249–2259.

46. Truong, T.D.; Duong, C.N.; Pham, H.A.; Raj, B.; Le, N.; Luu, K. The Right to Talk: An Audio-Visual Transformer Approach. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 1105–1114.
47. Cheng, G.; Han, J.; Zhou, P.; Guo, L. Multi-class geospatial object detection and geographic image classification based on collection of part detectors. *ISPRS J. Photogramm. Remote. Sens.* **2014**, *98*, 119–132. [CrossRef]
48. Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet Classification with Deep Convolutional Neural Networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*. Available online: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf (accessed on 17 January 2022).
49. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv* **2017**, arXiv:1710.09412.
50. DeVries, T.; Taylor, G.W. Improved regularization of convolutional neural networks with cutout. *arXiv* **2017**, arXiv:1708.04552.
51. Yun, S.; Han, D.; Oh, S.J.; Chun, S.; Choe, J.; Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 6023–6032.
52. Huang, S.; Wang, X.; Tao, D. SnapMix: Semantically Proportional Mixing for Augmenting Fine-grained Data. *arXiv* **2020**, arXiv:2012.04846.
53. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
54. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
55. 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*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
56. Everingham, M. *The PASCAL Visual Object Classes Challenge 2007*; Springer: New York, NY, USA, 2007.
57. 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.
58. Arjovsky, M.; Bottou, L. Towards principled methods for training generative adversarial networks. *arXiv* **2017**, arXiv:1701.04862.
59. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein generative adversarial networks. In Proceedings of the International Conference on Machine Learning Sydney, Sydney, NSW, Australia, 6–11 August 2017; pp. 214–223.
60. Mariani, G.; Scheidegger, F.; Istrate, R.; Bekas, C.; Malossi, C. Bagan: Data augmentation with balancing gan. *arXiv* **2018**, arXiv:1803.09655.
61. Odena, A.; Olah, C.; Shlens, J. Conditional image synthesis with auxiliary classifier gans. In Proceedings of the International Conference on Machine Learning, Sydney, NSW, Australia, 6–11 August 2017; pp. 2642–2651.
62. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
63. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 26–28 June 2020; Volume 34, pp. 12993–13000.

64. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016.
65. Kaggle. Global Wheat Detection. 2020. Available online: https://www.kaggle.com/c/global-wheat-detection (accessed on 17 January 2022).