Object Detection in Drone Imagery via Sample Balance Strategies and Local Feature Enhancement

Xiaoyu Hou 1,2, Kunlin Zhang 1,2, Jihui Xu 1,2, Wei Huang 1,2, Xinmiao Yu 1,2 and Huaiyu Xu 1,*

1 Shanghai Advanced Research Institute, Chinese Academy of Sciences, Shanghai 201210, China; houxiaoyu@sari.ac.cn (X.H.); zhangkl@sari.ac.cn (K.Z.); xujh@sari.ac.cn (J.X.); huangw@sari.ac.cn (W.H.); yuxinmiao2018@sari.ac.cn (X.Y.)
2 University of Chinese Academy of Sciences, Beijing 100049, China
* Correspondence: xuhuaiyu@sari.ac.cn

Abstract: With the advent of drones, new potential applications have emerged for the unconstrained analysis of images and videos from aerial view cameras. Despite the tremendous success of the generic object detection methods developed using ground-based photos, a considerable performance drop is observed when these same methods are directly applied to images captured by Unmanned Aerial Vehicles (UAVs). Usually, most of the work goes into improving the performance of the detector in aspects such as design loss, training sample selection, feature enhancement, and so forth. This paper proposes a detection framework based on an anchor-free detector with several modules, including a sample balance strategies module and super-resolved generated feature module, to improve performance. We proposed the sample balance strategies module to optimize the imbalance among training samples, especially the imbalance between positive and negative, and easy and hard samples. Due to the high frequencies and noisy representation of the small objects in images captured by drones, the detection task is extraordinarily challenging. However, when compared with other algorithms of this kind, our method achieves better results. We also propose a super-resolved generated GAN (Generative Adversarial Network) module with center-ness weights to effectively enhance the local feature map. Finally, we demonstrate our method’s effectiveness with the proposed modules by carrying out a state-of-the-art performance on Visdrone2020 benchmarks.

Keywords: object detection; drones; deep learning; sample imbalance; super-resolve GAN

1. Introduction

Object detection has been widely studied for decades [1]. The most famous detectors, such as those used for surveillance, mainly focus on the object of interest in images captured by ground-based cameras [2]. However, with the advantages of low cost, high flexibility, simple operation, and a small size, camera-equipped drones have been rapidly developed and deployed to replace satellites and cameras for a wide range of applications, such as in agriculture, aerial photography, delivery, surveillance, as well as in other fields [3]. Object detection is therefore one of the key technologies that will improve the perception capability of drones, and in addition, it is the basis for other intelligent algorithms, such as segmentation [4], object tracking [5], crowd estimation [6], etc. Despite the high demand for this technology, the drone-based detection algorithm still poses more challenges than the traditional ground-based detection algorithm. Progress has been slow in the research on object detection for drones, and this has gradually become one of the bottlenecks restricting the development of drones. The level of accuracy and real-time object detection will determine whether the drones’ mission will end with the destruction of the aircraft or its safekeeping. Limited by electric power, range, and environment, the drone-based object detection algorithm brings certain challenges:

(1) The instability of fast-moving UAVs means that aerial images are often blurred and noisy [7]. In addition, less feature information is extract-able from these moving
targets, the drone may repeatedly detect the same object, and it may falsely detect a target;

(2) The objects in need of detection are generally small in the images [8]. This means that when the UAV takes photos from high up, small targets are easily missed;

(3) The UAV’s continuous movement and the changes in the external environment (such as light, clouds, fog, rain, etc.) lead to drastic changes in the target’s features within the image, and thus increase the difficulty of subsequent feature extraction [9];

(4) The drone-based object detection algorithm needs to quickly and accurately detect moving targets [10], so the algorithm must meet real-time computing requirements.

Since the target usually appears small in drone images, the object’s features are often unclear and can easily be confused with the features of other objects. In addition, having excessive background in the image can lead to having too many negative samples in the training process, which affects detection accuracy. Motivated by these observations, this paper aims to improve the efficiency and accuracy of a drone object detection system based on the challenges mentioned above. We offer to study an object detection model based on the idea of an anchor-free framework that can reduce the amount of computation of IoU (Intersection over Union) [11]. In order to adapt the positive and negative samples, we propose new sample selection strategies. In addition, the weight-Generative Adversarial Network (GAN) sub-network is proposed to enhance the features locally. Following this, experiments carried out on Visdrone datasets [12] are used to demonstrate our method’s advantage over state-of-the-art detection methods.

2. Related Work

UAV (Unmanned Aerial Vehicle) is a technology that has emerged in recent years and offers more spatial resolution than standard remote acquisition systems such as satellite or airborne cameras. Despite recent developments offering some promising results, they still primarily rely on manual feature representation. These representations will limit the performance of the recognition system as they work well under limited conditions. The increase in spatial resolution poses new challenges for automatic classification because objects belonging to the same class will look very different to each other [13]. Besides, drone images are greatly affected by illumination, rotation, and scale changes, thereby further increasing the complexity of identifying the robust visual artifacts used to represent image content.

2.1. Object Detection

In the object detection task, we can classify the detector according to whether the algorithm uses anchors to generate candidate target boxes, which can usually be divided into two types: Anchor-based and anchor-free. The anchor-based detector can be divided into one-stage and two-stage according to the algorithm flow. Moreover, the anchor-free detector mainly contains keypoint-based and segmentation-based methods.

Anchor-based detector: An anchor (also known as an anchor box) is comprised of a set of rectangular boxes that are clustered on the training set. The boxes use k-means before training, which represents the aspect scale of the main distribution of the targets in the dataset. During the inference, the n-candidate rectangular boxes are extracted from these anchors on the feature map, and are generated before undergoing further classification and regression. There are two kinds of detector: Two-stage method and one-stage method.

The two-stage method first uses the algorithm to generate a series of candidate boxes, and then these are classified and regressed using the convolutional neural network. We can take Faster R-CNN [14] as an example, as it consists of a separate region proposal network and a region-wise prediction network to detect objects. Lots of algorithms are proposed based on the idea of Faster R-CNN. The authors of [15,16] propose to improve the training strategy and reform the loss function. The authors of [17,18] propose to redesign the architecture of the detection method. The authors of [19,20] propose the method on
feature fusion and enhancement. The authors of [21,22] propose improving the balance and proposal aspects during training.

The one-stage method is the end-to-end algorithm which directly regresses the class and position. It is faster than the two-stage method and only a little less accurate. The one-stage anchor-based detector receives lots of attention when working with the proposed SSD [23]. The authors of [24] introduce new loss function to improve the accuracy. The authors of [25,26] propose to enrich the feature and align the different domains. At present, the performances of the methods based on the one-stage anchor-based detector and the two-stage anchor-based detector are very close.

Anchor-free detector: As the name implies, anchor-free does not require setting the anchor’s aspect ratio in advance, including the keypoint-based method and segmentation-based method. Due to this simple network structure, it is probably more suitable for industrial applications.

The keypoint-based method first locates a few key-points generated by the pre-learned procedures to create bounding boxes for objects. CornerNet [27], defining two key-points (top-left corner and bottom-right corner) to represent the bounding box of objects, is the most representative of keypoint-based methods. At the same time, the CornerNet-Lite [28] method is the lite version improving its pace. CenterNet [29] extends CornetNet as a triplet (top-left corner, bottom-right corner and center) to improve performance. ExtremeNet [30] introduces five key points (top, left, bottom, right, and center) to generate objects’ bounding boxes. In pose estimation, 3D object location, and orientation identification, the key-point-based detector is the basic module, regressing other properties by center points in [31]. In RepPoint [32], it studies the deformable convolution and avoids the problem of no matching, despite key-points based detector, not like CornerNet or ExtremeNet.

The segmentation-based method is similar to the instance-segmentation algorithm. It tends to find any positive sample pixel in the detection frame and directly predict the bounding box’s four regression values (top, bottom, left, and right) by the full convolution branch. YOLO [33] is an early method that divides the picture into $n \times n$ grids. Each grid cell contains the center point of a target and then detects the objects. In DenseBox [34], the positive samples are defined as located in the center of the object. By regressing the four value distances from the center to the bound, the location and bounding boxes of objects are predicted. FSAF [35] uses the RetinaNet with an anchor-free branch to extract features and then predict four distances to bounds with the proposed branch, which define the central region of the target. FoveaBox [36] is inspired by the human eye’s fovea structure, which is divided into two parts: Central vision (foveal) and peripheral vision (peripheral). FoveaBox jointly predicts the possible location of the central area of the target for candidate box prediction. FCOS [37] is the fully convolutional one-stage object detection, which predicts the four distances with center-ness scores based on the positive samples.

2.2. Sample Imbalance

One of the problems in object detector training is the sample imbalance, especially the imbalance in the ratio of positive to negative samples. First, when training an object detector, regardless of whether it is anchor-based or anchor-free, we need to design the sample balance strategies, which can roughly be divided into three aspects: Positive and negative sample definition, sampling, and design loss. In terms of solving this problem how-ever, there are two main solutions. One is the hard sampling method, such as OHEM [38], in which a certain amount of positive and negative samples are selected from the whole sample base, and then only the selected samples need their loss calculating. The other is the soft sampling method, such as Focal Loss [39], which calculates the loss of all the selected samples but assigns different weights to different samples. In addition, the ATSS [40] method is critical as it indicates how to select positive and negative sample during object detection training, and is thus the essential difference between one-stage anchor-based and center-based anchor-free detectors. The hyper-parameter $k$ is designed in ATSS to select the positive candidate samples from each pyramid level.
2.3. Generative Adversarial Networks

The Generative Adversarial Network (GAN) [41] is a generator and discriminator framework. The discriminant network parameters are optimized to maximize the probability of correctly distinguishing real data from fake data. The purpose of generating the network is to maximize the likelihood that the discriminant network cannot identify its forged samples. GANs have been proven to be an excellent image generation model, and its performance in the fields of super-resolution [42], style transfer, and feature enhancement is continually improving. In [43,44], GANs are used to learn the map between two manifolds for style transfer. In [45], GANs are applied for image super-resolution. While Perception GANs [46] aims to generate super-resolved representations for small objects on the object detection task.

2.4. Datasets for Drone Imagery Object Detection

While ground-based datasets, such as MSCOCO [47], PASCAL VOC [48], and ImageNet [49] have achieved great success, when these datasets are used for object detection in drone images, there is a massive performance degradation. To date, there are not many datasets that can be applied to object detection from drones because it requires a significant amount of data annotation. COWC [50] is an aerial-based dataset which consist of 32.7 annotated vehicles and 5.8 useful negative samples, (i.e., boats, trailers, bushes, and A/C units). The quality, appearance, or rotation of annotated targets are all uncontrollable however. Meanwhile, the size of a vehicle in this dataset is between 24 to 48 pixels. CARPK [51] is a drone-based dataset that mainly focuses on car counting and includes 1448 images that were captured by drones in parking lots. DOTA [52] is an aerial-based dataset that contains 2806 aerial photos that are in 15 categories and 188,282 instances. Visdrone [12] is a drone-based dataset and a large-scale benchmark that facilitates object detection from drone imagery. The Visdrone datasets contain ten object categories, including pedestrian, person, car, van, bus, truck, motor, bicycle, awning-tricycle, and tricycle. In daily life, vehicles and people are the highest frequencies detected. In this paper, we mainly apply the Visdrone datasets to train and detect objects in the cars and people categories.

3. Methods

Object detection from drone imagery is becoming increasingly useful in many industry scenarios. However, there are many small targets in the detection task. In addition, the variations in altitude, the object’s scale, view angle, weather and illumination bring about more significant challenges than when using traditional object detection, such as when using ground-based cameras. In general, there are two technical routes, anchor-based and anchor-free. The anchor-based method can generate anchors to help achieve a high AP performance, such as Faster-RCNN [14], SSD [23], and other algorithms. However, with this method the aspect ratio needs to be artificially set, which means designing it in advance for any new datasets that have prior experience. In contrast, the anchor-free method does not require the design of anchor boxes, meaning that it has higher spatial freedom and is more suitable for target detection in UAV-based scenarios. Usually, there are two routes for anchor-free detection: Key-point based methods (e.g., Cornet [28] and CenterNet [29]) and segmentation based methods (e.g., FCOS [37], FSAF [35], Fovea [36], etc.). The keypoint-based methods enlarge the original image in order to improve accuracy based on the idea of crucial point detection, but at the same time, it increases the computing cost. On the other hand, the segmentation based method is mainly based on dense prediction to produce many false-positive samples, which thereby causes a higher recall and lower precision.

Therefore, as Figure 1 shows, the framework of object detection from images captured by drones, which mainly consists of four parts: Feature extracting module, adaptive selection for positive and negative, and easy and hard samples, weight GANs, and classifications and regression. In this paper, we combine the idea of a keypoint-based method with the FCOS-based algorithm. Using weight-GANs, we are able to achieve a local enhancement of the feature map. Simultaneously, we design the sample selection strategies for positive
and negative samples and hard and easy samples to improve AP performance. Finally, we apply two branches to get the classification and regression results.

![Figure 1. Framework of object detection from images captured by drones.](image1.png)

### 3.1. Feature Extracting Module and Offset-Head

We apply the Resnet101 [53] as the backbone and Feature Pyramid Networks (FPNs) [20] as the neck part section for the detector. The module of the feature maps extraction module is the basic part of the framework. Usually, FPN uses a top-down architecture, connected horizontally, to build intra-network feature pyramids from a single scale of input. The FPNs contain two parts: A down-top module and a top-down module. Extracted from the original image, the feature maps which have the same scale are called a stage. The last feature maps are stored in the stage of the bottom-top module. For objects detection, FPN is task-independent and each level of the pyramid in the FPN is used to detect objects at a specific scale. In total, There are four feature maps of different sizes totally are generated. After adding the up-sampled feature map with the feature map after that underwent $1 \times 1$ convolution, each feature map then goes through a $3 \times 3$ convolution layer, respectively, to eliminate the any negative impact that may have been caused by direct summation. Assume that these five feature maps levels, sorted from largest to smallest in size, are P3, P4, P5, P6 and P7. Then all the feature maps are assembled as and work as the features fed into the next module, just as is shown in Figure 2.

![Figure 2. The framework of the feature extracting module.](image2.png)

In this paper, the framework we proposed is designed based on the fully convolutional one-stage object detector, and it lets $F_i^k \in \mathbb{R}^{H \times W \times C}$ be the feature maps at pyramid level $i$. On each feature pyramid, the anchor-based approach places anchor points uniformly on each $H \times W$ spatial location, and the training target is determined by calculating the IoU overlap between all the anchor points and the ground truth box. Finally, the objectives are optimized using pyramid features. Besides, we also proposed a regression branch as the head part section of the anchor-free detector based on the idea of pixel-wise prediction (e.g.,...
The idea of pixel-wise prediction is similar with the algorithms of semantic segmentation, as the core is the method of dense prediction. The detector based on pixel-wise predictions can avoid complex calculations related to the anchor box, such as overlapping issues when training the model. In addition, it, and avoids introducing hyper-parameters related to the anchor box, which are often very sensitive to the final detection results performance. On each FPN layer $F_i$, the anchor points are uniformly distributed uniform. The Figure 3a shows the detailed structure of Head (regression branch), and (b) defines shows the definition of the four variants. The white points in (b) are the anchor points in some FPN layers, and the blue point is the center point of the ground truth object. The offset between anchor point and center point is defined as $(\Delta x, \Delta y)$, and the width and height of the predicted box are defined as $(\Delta w, \Delta h)$. The distance between the anchor point and center points can be calculated as: 

$$\text{Distance} = \sqrt{\Delta w^2 + \Delta y^2}.$$ 

By Through the regression branch, we can directly obtain the distance directly without needing any additional calculations. Thus the predicted boxes are regressed, a. Moreover, for each ground truth object box, each anchor regresses the corresponding offset, width and height. When there are multiple ground-truth object boxes mapped with one anchor, the anchor with the lowest distance from the center point of the object box will be kept. Specifically, each center point of in the ground-truth object box is defined as $(x_c, y_c)$, and the width and height are $w_c$ and $h_c$ respectively. Any anchor which falls into a ground-truth object box, it will be defined as a 4D vector $t^* = (\Delta x, \Delta y, \Delta w, \Delta h)$ which is regressed by the regression branch.

Figure 3. The details of detector head. (a) The detailed structure of Offset-Head (regression branch), and (b) The definition of four variants.

3.2. Adaptive Selection for Positive and Negative, and Easy and Hard Samples

In this paper, we study the relationship and trends between the hyper-parameter $k$ and the IoU threshold, as shown in Figure 4. We observe hyper-parameter $k$’s curve, and the mean and standard deviation of the top $k$ candidates, and find that when the hyper-parameter $k$ is set to no less than 9, the IoU’s lowest threshold is usually 0. Thus, with a hyper-parameter of 9, the selected positive and negative samples are more balanced, and there is a high probability that it can obtain a higher AP performance. This is consistent with the conclusions that are presented in ATSS [40] with regards to hyper-parameter $k$. Furthermore, the sample imbalance problem can be partly solved by using the balance strategies on the positive and negative samples balancing, and easy and hard samples balancing.
We propose the object detection framework based on FCOS, which utilizes IoU to divide the candidates into positive and negative samples. In the training process, sampler labels the anchor boxes with $\text{IoU} > \text{threshold}$ as positives and $\text{IoU} < \text{threshold}$ as negatives. In this paper, we use the mean of top $k$ candidates’ IoU as the threshold to make adaptive sample selection, which is sorted by distance from the center point of ground truth. The standard deviation of top $k$ candidates’ IoU can reflect the offsets of the candidate boxes. As shown in Figure 4, when the amount of candidates is increasing, the mean and standard deviation (std) of IoU tend to be constant. Since the distribution of IoU is not normalization, the anchor boxes with $\text{IoU} > \text{mean} + \text{std}$ belongs to easy-positives. The anchor boxes with $\text{IoU} < \text{mean} - \text{std}$ belongs to easy-negative. Moreover, the anchor boxes between $\text{mean} - \text{std}$ and $\text{mean} + \text{std}$ belongs to hard-positive and hard-negative. As shown in Figure 5, according to the statistics of candidates’ IoU, we counted the proportion of the amount of IoU values in different intervals. The proportion of hard-samples is higher, so the effective suppression is needed.

Positive and negative sample balance strategies: In a picture, the detection target only takes up a small part, and the remaining parts form up the background. During the training process, the boxes with an IoU over 0.5 are usually picked as positive samples, and the target boxes with an IoU below 0.5 are usually detected as negative samples. This will inevitably result in far more negative samples than positive samples, with the training process generally controlling the ratio of positive to negative samples at 1:3 [39].

Easy and hard sample balance strategies: The samples can be divided into four categories: Easy-positive samples, easy-negative samples, hard-positive samples and hard-
negative samples. The hard samples mean that they have been incorrectly classified as opposite samples. In general, hard samples contribute a significant loss but a small amount, while easy samples contribute a slight loss but a large amount. Therefore, the loss is impacted more by the easy samples and learned less from hard samples. Following this, we propose strategies on the adaptive selection for positive and negative, and easy and hard samples (ASPNEH) which are based on the idea of adaptive training sample selection (ATSS). Algorithm 1 is shown as below:

**Algorithm 1** Adaptive selection for positive and negative and easy and hard samples.

**Input:**
- $G$ is a set of ground-truth boxes in the image
- $A$ is the set of all anchor boxes generated from FPN features levels
- $k$ is a hyper-parameter with a default value of 9
- $\delta_1$ is the parameter which limits the amount of easy-samples with a default value of 1
- $\delta_2$ is the parameter which limits the amount of hard-samples with a default value of 2

**Output:**
- $EP$ is a set of easy-positive samples
- $EN$ is a set of easy-negative samples
- $HN$ is a set of hard-negative samples

1. for each ground-truth $g \in G$
2. \quad $C \leftarrow$ select the $k$ anchors from each level in $A$, and $c^k_g \in C$ that have the closest center are closest to the center of ground-truth $g$
3. \quad compute the IoU between $c^k_g$ and $g$: $d^k_g = IoU(c^k_g, g)$
4. \quad compute the mean of the top $k$ anchors: $m^k_g = \frac{1}{k} \sum d^k_g$
5. \quad compute the standard deviation of the top $k$ anchors: $s^k_g = \sqrt{\frac{1}{k} \sum (d^k_g - m^k_g)^2}$
6. \quad compute the IoU of the upper threshold: $t^k_{upper} = m^k_g + s^k_g$
7. \quad compute the IoU of the lower threshold: $t^k_{lower} = m^k_g - s^k_g$
8. end for
9. for each candidate $c_g \in C_g$
10. \quad if $IoU(c^k_g, g) \geq t^k_{upper}$ and center of $c$ in $g$ then
11. \quad \quad $EP = EP \cup c^k_g$
12. \quad end if
13. end for
14. for each candidate $c_g \in C_g$
15. \quad if $IoU(c^k_g, g) \leq t^k_{lower}$ then
16. \quad \quad $EN = EN \cup c^k_g$
17. \quad \quad if $count(EN) \geq \delta_1 \times count(EP)$ then
18. \quad \quad \quad break
19. \quad \quad end if
20. \quad end if
21. end for
22. for each candidate $c_g \in C_g$
23. \quad if $t^k_{lower} \leq IoU(c^k_g, g) \leq m^k_g$ then
24. \quad \quad $HN = HN \cup c^k_g$
25. \quad \quad if $count(HN) \geq \delta_2 \times count(EP)$ then
26. \quad \quad \quad break
27. \quad \quad end if
28. \quad end if
29. end for
30. return $EP, EN \cup HN$

We define $G$ as a set of ground-truth boxes in the images. For each ground-truth $g$, we select the $k$ anchor boxes whose center are closest to the center of $g$, based on $L_2$
distance. As described in Lines 1 to 7, we compute the mean and standard deviation of the top $k$ anchors, the IoU of the upper threshold as $t_{\text{upper}}^k = m_k^g + s_k^g$, and the IoU of the lower threshold as $t_{\text{lower}}^k = m_k^g - s_k^g$. Then, we select the positive samples sets and negative samples sets in from Line 8 to Line 21. When the value of the IoU is more above than the $t_{\text{upper}}^k$, the quality of candidates will be high, and they are will be divided into easy-positive samples whose that have a supposedly high IoU threshold is supposed to be high. Besides, the amount of negative samples are limited to less than $(\delta_1 + \delta_2)$ times the amount of positive samples. Thus, the candidates whose with an IoU below the $t_{\text{lower}}^k$ are divided into the sets of easy-negative samples. Moreover, the candidates whose with an IoU is no more than the mean of the top $k$ anchors and not less than $t_{\text{lower}}^k$ will be divided into hard-negative samples.

3.3. Weight-GAN Sub-Network

Due to the high frequency and noise surrounding the small objects and occluded objects in the images captured by drones, the detection task is extraordinarily challenging. In this paper, we propose a new GAN with center point weights to locally enhance the feature map, which is computed in a forward propagation of feature extracting modules. We need to train a generator model to generate the super-resolved representations for the small or occluded objects, and design a discriminator model that considers adversarial loss and center-ness loss to differentiate and supervise the generator model training.

First, we create a sub-dataset for the GAN task based on Visdrone datasets. This will be used to generate super-resolved large object-like representations for small objects. The samples of the datasets are shown as below in Figure 6. The high resolution objects are cropped from the whole images as raw samples, according to labeled bounding boxes. The sub-datasets contain the object categories of vehicles and people.

![Figure 6](image)

**Figure 6.** The sub-datasets we created for the Generative Adversarial Network (GAN) task to generate high resolution objects and down-sample images.

The Perceptual GAN [46] method provides an idea for detecting small objects by training a GAN model to transfer poor representations of small objects to super-resolved targets that are not easily distinguished by a discriminator. However, the detector based on the Perceptual GAN does not perform well with occluded objects. Due to this, the representation of targets is enhanced using Perceptual GANs, while the occluded objects will be ignored and their representation suppressed. So we proposed the use of center-ness weights to enhance the representation of the center points and suppress the representation of edges for the targets. Inspired by the Perceptual GANs, the sub-net can be described as Figure 7. (a) shows the network generator, which contains a deep residual network to enhance the features from the feature extraction module. In the meantime, the center-
ness weight aims to suppress the features of the edges. (b) shows the framework for the supervision and differentiates whether the image has high resolution target features or super-resolved features.

Figure 7. Details of the training procedure for the weight-GAN sub-network that uses super-resolved feature generation and discrimination. (a) shows the network generator, which contains a deep residual network to enhance the features from the feature extraction module. In the meantime, the center-ness weight aims to suppress the features of the edges. (b) shows the framework for the supervision and differentiates whether the image has high resolution target features or super-resolve features.

We use super-resolved generated GAN module with centerness weights to enhance local features and suppress the edge part features. Thus the occluded objects can be distinguished effectively by following classification and regression branches, just shown as Figure 8.

Figure 8. Occluded objects feature map visualization of FCOS and weight-GAN method.

Our model can be formulated as below. Here, $G$ represents a generator that is trained to map the features data from noisy data to the super-resolved features, and $D$ represents a discriminator that estimates the probability of a target feature coming from high resolution target features, rather than from $G$. In this paper, the functions $F_l$ and $F_s$ are the representation for high resolution object features and down-sampling features respectively. The function $f$ is the generator which learns to generate the residual representation between the representations of high resolution features and low resolution features through residual learning instead.

$$
\min_G \max_D L(D, G) \triangleq \mathbb{E}[\log D(f_l)] + \mathbb{E}[\log (1 - D(\omega_{\text{centerness}}F_s + G(\omega_{\text{centerness}}F_s|f)))]
$$

(1)
In our case, the variant $\omega_{\text{centerness}}$ represents the center-ness weight, which can be formulated as below. We define the width and height of the sample as $2w^c$, and $2h^c$ respectively, and $w^*$ is the horizontal distance between the center point and the point of the feature map horizontally, while $h^*$ is the vertical distance between the center point and the point of the feature map vertically. The closer the point of the feature map is to the center point, the higher heavier weight it has. With $\omega_{\text{centerness}}$, we can enhance the features of the center point, which has better feature representation than the edges.

$$
\omega_{\text{centerness}} = 1 - \sqrt{\frac{\min(w^*, w^c)}{\max(w^*, w^c)}} \times \frac{\min(h^*, h^c)}{\max(h^*, h^c)}
$$

Generator network: The generator network is mainly based on deep residual learning blocks which are easier to train. In order to improve the detection accuracy, the generator network aims to generate super-resolved representations for low resolution targets. In the generator network, since the details are absent from the low resolution feature, the deep residual blocks are trained, rather than the generator network being trained directly. First, the initial features are obtained by the feature extraction module and passed to $3 \times 3$ convolutional filters, then $1 \times 1$ convolutional filters are used to increase the feature dimension to align it with the output layer. The residual blocks consist of convolutional layers, batch normalization and ReLU activation.

Discriminator network: As shown in Figure 7, the discriminator network is mainly trained to distinguish the difference between the high-resolution target features and the super-resolved features which are generated based on low-resolution target features. There is one adversarial branch consisting of two fully-connected layers and an output layer with the sigmoid activation. The adversarial loss $L_{\text{adversarial}}$ is defined as below:

$$
L_{\text{adversarial}} = -\log D_\theta(\omega_{\text{centerness}} G_\theta(F_s))
$$

We denote $D_\theta$ as the adversarial function with the parameter of $\theta$ and take the generator function $G_\theta(F_s)$ as the input of the discriminator network. In order to enhance the local features and suppress the features of the edges, we introduce a parameter $\omega_{\text{centerness}}$, which calculates the proximity between the point of center and the points of the feature map from the super-resolved generator.

3.4. Classification and Regression

In this section, we reuse the branches of classification and regression based on FCOS. After the feature enhancement with weight-GAN, the classification and regression branches help the final results and improve overall performance. Just as shown in Figure 1, the detector we designed is suitable for small object detection tasks in drones scenarios. The loss of classification $L_{\text{cls}}$ is focal loss, and the loss of regression $L_{\text{reg}}$ is the IoU loss. Thus the loss can be formulated as below:

$$
L = \frac{\sum L_{\text{cls}}(p(x,y)c^*(x,y))}{N_{\text{positive}}} + \lambda \frac{\sum L_{\text{reg}}(t(x,y,w,h)t^*(x,y,w,h))}{N_{\text{positive}}}
$$

$N_{\text{positive}}$ denotes the number of positive samples, and $\lambda$ is the balance weight for $L_{\text{reg}}$ with a default value of 1. $c^*(x,y)$ denotes the ground-truth object classification, and $t^*(x,y,w,h)$ denotes the ground-truth object bounding box. In the Figure 1, the inference made by of the detector we designed is forward propagation, while. Moreover, the weight-GANs is the subnet, which needs to be trained separately.

4. Experiments

In this paper, we propose an object detection framework based around the problems of small targets and the presence of too many negative samples in UAV object detection. In this section, we will discuss the following aspects:
• Whether the Sample Balance Strategies (SBS) we proposed for Positive and Negative, and Easy and Hard Samples can improve the AP performance for the selection of positive candidates.
• Whether the weight-GAN subnet can effectively enhance the local features of objects and improve the accuracy of small object detection.
• Verify the detection framework we proposed that combines the Sample Balance Strategies and weight-GANs subnet for the detection of objects from drones, and evaluate it against CornetNet [27], FPN [20], FCOS+ATSS [40], Perceptual GANs [46] and so forth.

For the software environment, all experiments are implemented based on the Pytorch [54] and mmdetection [55] frameworks. The hardware environment of this experiment is the Intel Core i7-6500k CPU 3.4 GHz, and with two GPUs of TITAN RTX 24G memory. We initialize our backbone networks and FPN network with the weights pre-trained on ImageNet [49]. We verify our detection framework on the VisDrone2020 DET dataset [12], which consists of 10,209 images in unconstrained challenging scenes, including 6471 images in the training subset, 548 in the validation subset, 1580 in the test-challenge subset, and 1610 in the test-dev subset. Since our experiment is mainly based on the mmdetection framework, which requires the COCO data format, we convert the VisDrone2020 datasets into the COCO data format. During the training process, we first train the weight-GAN based on the dataset we generated. Then, by the forward propagation of weight-GAN, the local features will be enhanced after the generation of the candidates’ boxes. Finally, the classification and regression will be regressed by the framework we proposed.

### 4.1. Implementation and Performance Evaluation of Samples Balance Strategies

For the observation of sample balancing, we consider balancing the positive and negative, and easy and hard samples using SBS, which is short for Sample Balance Strategies. In this paper, the SBS method aims to balance the distribution of positive and negative samples and suppress the excessive negative samples through easy and hard samples. In order to reduce the computation of IoU, we propose that the Offset-Head replace the original FCOS-head which is based on the anchor-free detector FCOS. The Offset-Head directly regresses the offset \((\Delta x, \Delta y)\) between the anchor point and the center point of the ground-truth object. In order to verify SBS’s effectiveness, we first use the same datasets, such as MSCOCO minival set, to compare the ATSS with our own algorithm, as shown below in Table 1.

| Method        | AP [%] | AP50 [%] | AP75 [%] | APs [%] | AM [%] | AL [%] |
|---------------|--------|----------|----------|--------|-------|-------|
| FCOS + ATSS   | 39.21  | 57.38    | 41.36    | 22.34  | 42.54 | 49.80 |
| Ours w/SBS    | 39.44  | 57.41    | 41.11    | 24.13  | 42.55 | 49.77 |

Our method with SBS can achieves AP 39.44%, and improves detection by 0.23% on AP, 0.03% on AP50, 1.79% on AP75, 0.03% on APs. To be specific, SBS can significantly improve the AP performance of small objects significantly. Because of the balance between positive and negative, many negative samples are suppressed. However, 0.03% are declined on APL. As shown in Figure 9, excluding its for small object detection capabilities, the samples balance method is not much different.

We apply our method with SBS only on the VisDrone2020 dataset and the result can be found as in the below Figure 9. The candidates’ boxes with higher accuracy and a lower amount of false-positives can be better regressed by our method than by FPN + RPN and FCOS w/ATSS.
By comparing the experimental verification and analysis, we can draw the following conclusions for our SBS method:

- Small object detection can achieve a higher performance with sample balance strategies for positive and negative samples, and easy and hard samples.
- In this task, the AP performance of our method is higher than others’ performance, such as those from FCOS w/ATSS.

4.2. Implementation and Performance Evaluation of Weight-GAN Subnet

In order to verify the effectiveness of the weight-GAN method, we compare our method against several other feature enhancement methods on their ability to detect objects through drone vision. As shown in Table 2, we train and test the models on the datasets, which are all generated from VisDrone2020 as shown in Figure 6. Our method can achieve the state-of-the-art performance, and improve by 5.46% for Large Scale Images, by 3.91% for SRGAN [45], by 3.59% for ESRGAN [56] and by 1.23% for Perceptual GANs. The Large Scale Images method represents the model trained using high-resolution images by directly increasing the scale of the input image, e.g., ×4. The SRGAN and ESRGAN methods provide the generative network for the image super-resolution (SR). Perceptual GAN is able to generate super-resolved representations for small objects, but it has a lower accuracy than our method. Thus it proves that the method we have proposed to enhance local features is effective for the detection of objects in drone scenarios.

Table 2. Comparisons of detection performance with different methods on weight-GAN sub-network datasets generated from VisDrone2020.

| Method                | Top1—Accuracy [%] |
|-----------------------|-------------------|
| Large Scale Images    | 65.98             |
| SRGAN                 | 67.53             |
| ESRGAN                | 67.85             |
| Perceptual GANs       | 70.21             |
| Ours w/weight-GAN     | 71.44             |

We also visualize the intermediate results of the super-resolved features generated by weight-GAN, as shown in Figure 10. The first column shows the candidate objects in images captured by drones, while the second column and final column display the features of small objects and large objects, respectively. Because of the residual block in weight-GAN, the third column shows the residual representation features that are generated by the residual block with center-ness weights. Next, in the fourth column, the super-resolved features are generated by weight-GAN.

The extracted features of small objects are easy to be disturbed by the noise of contexts. Thus, we extract the basic feature map based on FPN with multi-level layers to generate the small object feature representation. We utilize four $H \times W \times 256$ fully convolutional layers to regress the class and bounding box position in the following classification branch and regression branch. $H \times W$ is the height and width of feature maps. The detector we proposed based on FCOS directly views the locations as training samples rather than
anchor boxes, similar to FCNs for segmentation tasks. Thus, the enhancement of local features and the suppression of edge features for one small object helps be regressed and classified via fully convolutional layers. We can observe that the result from the super-resolved features is similar to the result from the large objects, and in addition, that the center-ness of learned features are enhanced and edges are suppressed. Thus the method we proposed achieves a better performance.

![Image](image.png)

Figure 10. Visualization of the super-resolved features generated by weight-GAN.

4.3. Performance Evaluation of Detection on Visdrone Datasets

Hyper-parameter k. In this paper, one hyper-parameters $k$ is used to select the candidate positive and negative samples from each pyramid level. As shown in Table 3, we use different values of $k$ to train the detector. We observe that too large $k$ will cause lots of low-quality candidates, which decreases the AP, and too small $k$ will result in insufficient samples. Overall for the visdrone2020 dataset, the best performance can be obtained when $k$ is taken as 9.

| $k$   | 5   | 7   | 9    | 11  | 13  | 15  |
|-------|-----|-----|------|-----|-----|-----|
| AP    | 40.54 | 41.98 | **42.12** | 41.09 | 40.88 | 39.48 |

Table 3. Table analysis of different values of hyper-parameter $k$ on the Visdrone2020 car class.

Comparison. As shown in Table 4, we compare our proposed method to other famous algorithms on the Visdrone2020 benchmark. Following the benchmarks, we evaluate the performance of our method with the $AP_{IoU=0.50:0.05:0.95}$, $AP_{IoU=0.50}$, $AP_{IoU=0.75}$, $AR_{max=1}$, $AR_{max=10}$, $AR_{max=100}$ and $AR_{max=500}$ scores, which are designed in Visdrone2020. Specifically, $AP_{IoU=0.50:0.05:0.95}$ is computed by averaging over all 10 IoU thresholds over all and of across all categories, which is then used as the primary metric for ranking algorithms. $AP_{IoU=0.50}$ and $AP_{IoU=0.75}$ are computed at the single IoU thresholds of 0.5 and 0.75 over all categories, respectively. The $AR_{max=1}$, $AR_{max=10}$, $AR_{max=100}$ and $AR_{max=500}$ scores are the maximum recalls given of 1, 10, 100 and 500 detections per image, averaged over all categories and IoU thresholds.
Table 4. A table for calculating the training AP and AR of different models for the same dataset (Visdrone2020-dev).

| Method               | AP [%] | AP50 [%] | AP75 [%] | AR1 [%] | AR10 [%] | AR100 [%] | AR500 [%] |
|----------------------|--------|----------|----------|---------|----------|-----------|-----------|
| CornetNet            | 23.43  | 41.18    | 25.02    | 0.45    | 4.24     | 33.05     | 34.23     |
| FPN                  | 22.06  | 39.57    | 22.50    | 0.29    | 3.50     | 30.64     | 31.61     |
| FCOS + ATSS          | 23.39  | 39.52    | 26.34    | 0.41    | 3.45     | 31.49     | 31.49     |
| Perceptual GANs      | 21.32  | 38.34    | 25.11    | 0.27    | 3.55     | 32.11     | 32.07     |
| Ours w/SBS           | 23.42  | 39.77    | 26.36    | 0.47    | 3.98     | 31.65     | 31.65     |
| Ours w/WGAN          | 23.34  | 39.56    | 26.18    | 0.25    | 3.03     | 32.71     | 32.69     |
| Ours w/SBS + WGAN    | 24.17  | 40.98    | 26.71    | 0.51    | 4.29     | 33.19     | 33.19     |

Compared with the state-of-the-art CornerNet method, our method that uses the Sample Balance Strategy and weight-GAN Sub-network performs better and improves the AP by 0.74%. In terms of sample selection strategies meanwhile, we compare our method to the state-of-the-art FCOS+ATSS method. Our methods, both using SBS only, as well as using SBS and WGAN, improve the AP by 0.03% and by 0.78% respectively. This proves the effectiveness of the SBS method in the detection of objects from drones. Besides, when compared with Perceptual GANs, our method weight-GAN also achieves a better performance. The results of our detection method are shown in Figure 11.

Figure 11. The results of our detection framework on the VisDrone2020 benchmark.

For object detection tasks in UAV scenarios, there are many small targets whose feature maps usually are not easy to be distinguished by the following regression and classification branches. In order to improve the detector’s performance, we propose the weight-GAN method to enhance the local feature of small targets and introduce the sample balance strategies.

5. Conclusions

For scenarios that require object detection from drones, the training datasets are not as rich as those from the ground-based dataset, e.g., ImageNet. In addition, the detection tasks needed mostly focus on small objects. In order to improve the performance of the detector therefore, most studies usually try to improve the aspects of loss design, training sample selection, feature enhancement and so forth. In this paper, we propose an object detection framework for drone scenarios that uses Sample Balance Strategies and a weight-GANs sub-network to improve detection performance. In terms of the selection of training samples, we propose the Sample Selection Strategies method, which balances positive and negative samples, and easy and hard samples based on ATSS (Adaptive Training Sample Selection). Validated through experiments, our method’s performance has been shown to surpass that of ATSS. Furthermore, in terms of the enhancement of features, we introduce weight-GAN to generate super-resolved features and enhance the representation of small objects with center-ness weights based on Perceptual GANs. Using experiments, we can observe that our method performs in a better way than simply enlarging the scale of the image and using Perceptual GANs. Finally, we compare our proposed method to the benchmark and again achieve a better performance in UAV object detection scenarios.
Author Contributions: Conceptualization, X.H. and H.X.; methodology, X.H.; software, X.H. and K.Z.; validation, X.Y., J.X. and K.Z.; formal analysis, X.Y. and W.H.; investigation, X.H. and W.H.; resources, X.H.; data curation, X.Y., J.X. and K.Z.; writing—original draft preparation, X.H., K.Z. and J.X.; writing—review and editing, H.X. and J.X.; visualization, X.Y. and X.H.; supervision, K.Z.; project administration, X.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available on request due to restrictions e.g., privacy or ethical. The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

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

References

1. Han, J.; Zhang, D.; Cheng, G.; Liu, N.; Xu, D. Advanced deep-learning techniques for salient and category-specific object detection: A survey. IEEE Signal Process. Mag. 2018, 35, 84–100. [CrossRef]

2. Dev, S.; Wen, B.; Lee, Y.H.; Winkler, S. Ground-based image analysis: A tutorial on machine-learning techniques and applications. IEEE Geosci. Remote Sens. Mag. 2016, 4, 79–93. [CrossRef]

3. Leal-Taixé, L.; Roth, S. Computer Vision—ECCV 2018 Workshops: Munich, Germany, September 8–14. 2018, Proceedings, Part VI; Springer: Cham, Switzerland, 2019; Volume 1134.

4. Angelova, A.; Zhu, S. Efficient object detection and segmentation for fine-grained recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 811–818.

5. Voigtlaender, P.; Krause, M.; Osep, A.; Luiten, J.; Sekar, B.B.G.; Geiger, A.; Leibe, B. Mots: Multi-object tracking and segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7942–7951.

6. Sindagi, V.A.; Patel, V.M. A survey of recent advances in cnn-based single image crowd counting and density estimation. Pattern Recognit. Lett. 2018, 107, 3–16. [CrossRef]

7. Ma, Y.; Wu, X.; Yu, G.; Xu, Y.; Wang, Y. Pedestrian detection and tracking from low-resolution unmanned aerial vehicle thermal imagery. Sensors 2016, 16, 446. [CrossRef] [PubMed]

8. Liang, X.; Zhang, J.; Zhuo, L.; Li, Y.; Tian, Q. Small object detection in unmanned aerial vehicle images using feature fusion and scaling-based single shot detector with spatial context analysis. IEEE Trans. Circuits Syst. Video Technol. 2019, 30, 1758–1770. [CrossRef]

9. Wu, Z.; Suresh, K.; Narayanan, P.; Xu, H.; Kwon, H.; Wang, Z. Delving into robust object detection from unmanned aerial vehicles: A deep nuisance disentanglement approach. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 1201–1210.

10. 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.

11. 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, 7–12 February 2020; Volume 34, pp. 12993–13000.

12. Zhu, P.; Wen, L.; Du, D.; Bian, X.; Hu, Q.; Ling, H. Vision meets drones: Past, present and future. arXiv 2020, arXiv:2001.06303.

13. Bazzi, Y.; Melgani, F. Convolutional SVM networks for object detection in UAV imagery. IEEE Trans. Geosci. Remote Sens. 2018, 56, 3107–3118. [CrossRef]

14. 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]

15. Lu, Q.; Liu, C.; Jiang, Z.; Men, A.; Yang, B. G-CNN: Object detection via grid convolutional neural network. IEEE Access 2017, 5, 24023–24031. [CrossRef]

16. Chou, J.; Guo, Z.; Leng, L. Object detection based on multi-layer convolution feature fusion and online hard example mining. IEEE Access 2018, 6, 19959–19967. [CrossRef]

17. Dai, J.; Li, Y.; He, K.; Sun, J. R-FCN: Object detection via region-based fully convolutional networks [EB/OL]. arXiv 2018, arXiv:1605.06409.

18. Li, Y.; Chen, Y.; Wang, N.; Zhang, Z. Scale-aware trident networks for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 6054–6063.

19. Kong, T.; Yao, A.; Chen, Y.; Sun, F. HyperNet: Towards Accurate Region Proposal Generation and Joint Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
20. 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.

21. Tan, Z.; Nie, X.; Qian, Q.; Li, N.; Li, H. Learning to Rank Proposals for Object Detection. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27–28 October 2019.

22. Pang, J.; Chen, K.; Shi, J.; Feng, H.; Ouyang, W.; Lin, D. Libra R-CNN: Towards Balanced Learning for Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.

23. 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.

24. Chen, K.; Li, J.; Lin, W.; See, J.; Zou, J. Towards Accurate One-Stage Object Detection with AP-Loss. In Proceedings of the IEEE/CVF International Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.

25. Li, S.; Yang, L.; Huang, J.; Hua, X.S.; Zhang, L. Dynamic Anchor Feature Selection for Single-Shot Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019.

26. Liu, S.; Huang, D. Receptive field block net for accurate and fast object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 385–400.

27. Law, H.; Deng, J. CornerNet: Detecting objects as paired keypoints. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 734–750.

28. Law, H.; Teng, Y.; Russakovsky, O.; Deng, J. CornerNet-lite: Efficient keypoint based object detection. \textit{arXiv} 2019, \textit{arXiv}:1904.08900.

29. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. CenterNet: Keypoint triplets for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 6569–6578.

30. Zhou, X.; Zhuo, J.; Krahenbuhl, P. Bottom-up object detection by grouping extreme and center points. In Proceedings of the IEEE/CVF International Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 830–839.

31. Zhou, X.; Wang, D.; Krahenbuhl, P. Objects as points. \textit{arXiv} 2019, \textit{arXiv}:1904.07850.

32. Yang, Z.; Liu, S.; Hu, H.; Wang, L.; Lin, S. Reppoints: Point set representation for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 9657–9666.

33. Shafiee, M.J.; Chywl, B.; Li, F.; Wong, A. Fast YOLO: A fast you only look once system for real-time embedded object detection in video. \textit{arXiv} 2017, \textit{arXiv}:1709.05943.

34. Huang, L.; Yang, Y.; Deng, Y.; Yu, Y. Densebox: Unifying landmark localization with end to end object detection. \textit{arXiv} 2015, \textit{arXiv}:1509.04874.

35. Zhu, C.; He, Y.; Savvides, M. Feature selective anchor-free module for single-shot object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 840–849.

36. Kong, T.; Sun, F.; Liu, H.; Jiang, Y.; Li, L.; Shi, J. Foveabox: Beyond anchor-based object detection. \textit{IEEE Trans. Image Process.} 2020, 29, 7389–7398. [CrossRef]

37. Tian, Z.; Shen, C.; Chen, H.; He, T. FCos: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 9627–9636.

38. Shrivastava, A.; Gupta, A.; Girshick, R. Training region-based object detectors with online hard example mining. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 761–769.

39. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.

40. Zhang, S.; Chi, C.; Yao, Y.; Lei, Z.; Li, S.Z. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9759–9768.

41. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. \textit{Adv. Neural Inf. Process. Syst.} 2014, 3, 2672–2680. [CrossRef]

42. Denton, E.; Chintala, S.; Szlam, A.; Fergus, R. Deep generative image models using a laplacian pyramid of adversarial networks. \textit{arXiv} 2015, \textit{arXiv}:1506.05751.

43. Li, C.; Wand, M. Combining markov random fields and convolutional neural networks for image synthesis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2479–2486.

44. Yeh, R.; Chen, C.; Lim, T.; Hasegawa-Johnson, M.; Do, M. Semantic image inpainting with perceptual and contextual losses, vol. 2. \textit{arXiv} 2016, \textit{arXiv}:1607.07539.

45. Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4681–4690.

46. Li, J.; Liang, X.; Wei, Y.; Xu, T.; Feng, J.; Yan, S. Perceptual generative adversarial networks for small object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1222–1230.

47. 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.

48. Everingham, M.; Eslami, S.A.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes challenge: A retrospective. \textit{Int. J. Comput. Vis.} 2015, 111, 98–136. [CrossRef]
49. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* 2015, 115, 211–252. [CrossRef]

50. Mundhenk, T.N.; Konjevod, G.; Sakla, W.A.; Boakye, K. A large contextual dataset for classification, detection and counting of cars with deep learning. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 785–800.

51. Hsieh, M.R.; Lin, Y.L.; Hsu, W.H. Drone-based object counting by spatially regularized regional proposal network. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4145–4153.

52. Xia, G.S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A large-scale dataset for object detection in aerial images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 3974–3983.

53. Targ, S.; Almeida, D.; Lyman, K. Resnet in resnet: Generalizing residual architectures. *arXiv* 2016, arXiv:1603.08029.

54. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv* 2019, arXiv:1912.01703.

55. Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; et al. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv* 2019, arXiv:1906.07155.

56. Wang, X.; Yu, K.; Wu, S.; Gu, J.; Liu, Y.; Dong, C.; Loy, C.C.; Qiao, Y.; Tang, X. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, Munich, Germany, 8–14 September 2018.