International Journal of Geo-Information

MDPI

Article

Techniques for the Automatic Detection and Hiding of Sensitive Targets in Emergency Mapping Based on Remote Sensing Data

Tianqi Qiu [1], Xiaojin Liang [1], Qingyun Du [1,2,3,4,*], Fu Ren [1,2], Pengjie Lu [1] and Chao Wu [5]

[1] School of Resource and Environmental Sciences, Wuhan University, Wuhan 430079, China; qtq@whu.edu.cn (T.Q.); liangxj@whu.edu.cn (X.L.); renfu@whu.edu.cn (F.R.); lupengjie@whu.edu.cn (P.L.)
[2] Key Laboratory of Geographic Information Systems, Ministry of Education, Wuhan University, Wuhan 430079, China
[3] Key Laboratory of Digital Mapping and Land Information Application Engineering, National Administration of Surveying, Mapping and Geoinformation, Wuhan University, Wuhan 430079, China
[4] Collaborative Innovation Center of Geospatial Technology, Wuhan University, Wuhan 430079, China
[5] School of Geographic and Biologic Information, Nanjing University of Posts and Telecommunications, Nanjing 210023, China; chaowu@njupt.edu.cn
* Correspondence: qydu@whu.edu.cn; Tel.: +86-27-8766-4557; Fax: +86-27-6877-8893

Abstract: Emergency remote sensing mapping can provide support for decision making in disaster assessment or disaster relief, and therefore plays an important role in disaster response. Traditional emergency remote sensing mapping methods use decryption algorithms based on manual retrieval and image editing tools when processing sensitive targets. Although these traditional methods can achieve target recognition, they are inefficient and cannot meet the high time efficiency requirements of disaster relief. In this paper, we combined an object detection model with a generative adversarial network model to build a two-stage deep learning model for sensitive target detection and hiding in remote sensing images, and we verified the model performance on the aircraft object processing problem in remote sensing mapping. To improve the experimental protocol, we introduced a modification to the reconstruction loss function, candidate frame optimization in the region proposal network, the PointRend algorithm, and a modified attention mechanism based on the characteristics of aircraft objects. Experiments revealed that our method is more efficient than traditional manual processing; the precision is 94.87%, the recall is 84.75% higher than that of the original mask R-CNN model, and the F1-score is 44% higher than that of the original model. In addition, our method can quickly and intelligently detect and hide sensitive targets in remote sensing images, thereby shortening the time needed for emergency mapping.

Keywords: emergency mapping based on remote sensing data; sensitive object detection; sensitive object hiding; mask R-CNN model; PointRend; Deepfill model

Citation: Qiu, T.; Liang, X.; Du, Q.; Ren, F.; Lu, P.; Wu, C. Techniques for the Automatic Detection and Hiding of Sensitive Targets in Emergency Mapping Based on Remote Sensing Data. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 68. https://doi.org/10.3390/ijgi10020068

Academic Editor: Wolfgang Kainz
Received: 2 January 2021
Accepted: 5 February 2021
Published: 9 February 2021

1. Introduction

Remote sensing images have the characteristics of wide coverage and high timeliness of data collection. They can provide timely and effective image surveying and mapping data for disaster areas and support governments and rescue agencies at all levels in emergency decision making, disaster assessment, and rescue deployment [1]. There have been many studies and applications of remote sensing mapping for emergency disaster response. For example, Yang Kejian et al. [2] discussed the application of remote sensing mapping technology in flood monitoring and evaluation; Fan Yida et al. [3] used remote sensing images to study an emergency disaster assessment method for the Wenchuan earthquake; and Xue Tengfei et al. [4] proposed an automatic mapping method based on

remote sensing for earthquake emergencies. Recently, automatic mapping of disaster-affected areas has been studied using a different approach of the convolutional neural network (CNN) structure. Hacefendiolu et al. [5] used a pretrained faster R-CNN to detect the earthquake-induced ground failure areas and damaged structures, and Ghorbanzadeh et al. [6] applied two main deep-learning CNN streams combined with the Dempster–Shafer model to automatic landslide mapping.

In remote sensing mapping, according to China's Surveying and Mapping Law [7] and State Secrets Law [8], sensitive geographic targets that are relevant to national security, including military installations, large-scale weapons and equipment, secret agencies, and nuclear facilities, must be hidden before remote sensing images can be publicly released and used. At present, most processing methods rely on manual or semi-automatic methods of finding specific sensitive targets and using image editing tools to capture the background textures around these targets to cover and fill in the target areas. In the emergency remote sensing mapping scenario, however, this approach is not efficient or robust; the results are easily affected by the operator's skill, and the hiding effect is not ideal. These shortcomings restrict the rapid release and use of remote sensing image map products, preventing this approach from meeting the timeliness requirements for disaster emergency response. Therefore, how to automatically detect and hide sensitive targets is a subject worthy of study.

With the development of deep learning models and methods, extensive efforts have been made to achieve the automatic detection and hiding of objects. At present, the process of target recognition and detection is usually performed by machine learning algorithms. Feature extraction algorithms for application to remote sensing images can be divided into algorithms for low-level, mid-level, and high-level feature extraction. Low-level feature extraction algorithms extract a certain aspect of images, such as gradient, color, or texture. Such algorithms include the histogram of oriented gradients (HOG) algorithm [9], the scale-invariant feature transform (SIFT) [10], local binary pattern (LBP) [11], and speeded-up robust feature (SURF) [12]. Mid-level feature extraction refers to a combination of multiple low-level feature extraction algorithms to improve the expressiveness of the extracted features. Representative methods include MultiFtr [13], integral channel features (ChnFtrs) [14], and the fastest pedestrian detector in the west (FPDW) [15].

Traditional machine learning technology does not perform well when processing raw unstructured data. With the goal of extracting more hierarchical, abstract, and high-level features, a CNN can be regarded as a feature learning tool that can learn rich hierarchically structured feature representations from raw data. With these features, great performance improvements can be achieved in target recognition and detection and even in other vision tasks. Therefore, CNNs have undergone rapid development and have seen widespread use in computer vision applications. In the past two decades, a variety of effective detection frameworks have been developed to achieve the purpose of organically combining positioning, feature extraction and other auxiliary algorithms for efficient detection, such as fast region-based CNN (R-CNN) [16], faster R-CNN [17], mask R-CNN [18], you-only-look-once (YOLO) [19], and the single-shot multibox detector (SSD) [20]. Extensive efforts have been made to study how to use these deep CNNs for high-resolution remote sensing (HRRS). Researchers [21,22] have shown that transfer learning provides a powerful tool for remote sensing scene classification, the features from pretrained CNNs generalize well to HRRS datasets and are more expressive than the low- and mid-level features. As HRRS always needs a large amount of labeled data and cannot recognize the images from an unseen scene class without any visual sample in the labeled data, to overcome this drawback, zero-shot scene classification has been used to recognize images from unseen scene classes [23,24]. CNNs have been increasingly established as adaptive methods for new challenges in the field of earth observation (EO). Hoeser et al. provided a comprehensive overview of the impact of CNNs on EO applications [25,26].

Mask R-CNN is currently the most advanced deep learning model for this purpose, with powerful object detection and instance segmentation capabilities. It has achieved promising progress in natural image recognition. However, due to the resolution, shadowing, scale, and data volume of high-resolution remote sensing images, deep learning models are not very effective for instance segmentation in remote sensing images, and because the mask size generated by mask R-CNN is 14 × 14 pixels or 28 × 28 pixels, a large amount of detailed information is lost due to the high zoom ratio at the boundaries of large objects.

Image inpainting is a technology for inferring the information content of a missing image area based on the known information around it, and then repairing it to make the image complete. This technology is widely used for repairing damaged photographs, covering redundant information, hiding sensitive targets, and similar tasks [27]. In essence, the hiding of sensitive objects in remote sensing images is a form of image restoration processing. Traditional methods can be divided into two main categories, i.e., one is based on partial differential equations, such as the self-adaptive curvature repair algorithm based on the curvature-driven diffusion (CDD) model proposed by Yin Yong et al. [28], and the other category consists of sample-based methods, such as the PatchMatch algorithm proposed by Barnes et al. [29]. A generative adversarial network (GAN) model is an unsupervised model that uses random strategies to continuously learn and understand the abstract structures and high-dimensional semantic information of images through training. However, remote sensing images have the characteristics of complex backgrounds, diverse scales, and a large amount of information. When a GAN model is used to repair large-format remote sensing images, the internal structures and textures of the repaired areas typically have a high degree of similarity, while the edge parts and other background areas exhibit obvious differences.

To address the low accuracy of mask R-CNN in remote sensing image instance segmentation and the large differences at the edges of GAN-repaired patches, this paper proposes corresponding solutions. First, for mask R-CNN, we modify the loss function and optimize the region proposal network (RPN)-derived candidate frames to improve the accuracy of target detection; second, we introduce the PointRend [30] algorithm to enhance the accuracy at boundaries; and third, we combine the improved mask R-CNN model with a GAN model and apply the resulting method for the automatic detection and hiding of sensitive targets in remote sensing images. Experiments show that the accuracy and efficiency of this method in emergency remote sensing mapping are greatly improved as compared with traditional methods.

2. Materials and Methods

2.1. Overall Framework

The overall framework, as shown in Figure 1, consists of a mask R-CNN + PointRend model and a Deepfill [31] model. The mask R-CNN + PointRend model performs sensitive target detection, and the Deepfill model hides the detected sensitive targets. The combination of the two achieves the purposes of reducing computational overhead and improving efficiency.

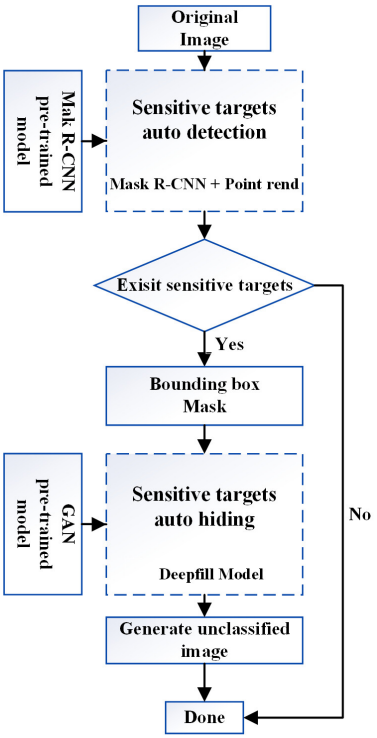


Figure 1. Two-stage model framework.

The model processing procedure is as follows: ① Input the original images and pre-trained target weight parameters, then, use the mask R-CNN + PointRend model to quickly retrieve and screen the images to detect whether sensitive targets exist. ② If no sensitive targets exist, the images can be used directly for subsequent production; if sensitive targets do exist, then, those targets are located and segmented, and the coordinates of the sensitive targets and the corresponding masks are output. ③ Use the Deepfill model to hide the marked areas in accordance with the output masks to generate unclassified images for subsequent production.

2.2. Mask R-CNN

The mask R-CNN model is a deep CNN model proposed, in 2017, by He et al. [18] for target detection tasks. It is based on the faster R-CNN model and additionally includes a branch network for predicting a segmentation mask for each region of interest (ROI) in order to generate high-quality masks for targets. The network structure of the mask R-CNN model, which is shown in Figure 2, includes a feature extraction network (a residual network, ResNet) [32], a feature pyramid network (FPN) [33], an RPN, and an ROIAlign and pixel segmentation network (a fully convolutional network, FCN) [34]. The conv2_x, conv3_x, conv4_x, and conv5_x structural blocks in the ResNet constitute four feature maps representing different levels of semantic information of the targets. The FPN performs summation operations on the feature maps at different levels to obtain high-level semantic information while preserving the spatial information of the targets. The combination of the feature maps is shown in Equation (1), where $i = 2,3,4$, $upsample$ is the upsampling convolution operation, $Conv$ is the convolution operation, and Sum represents the summation of the values in the corresponding position of the matrix.

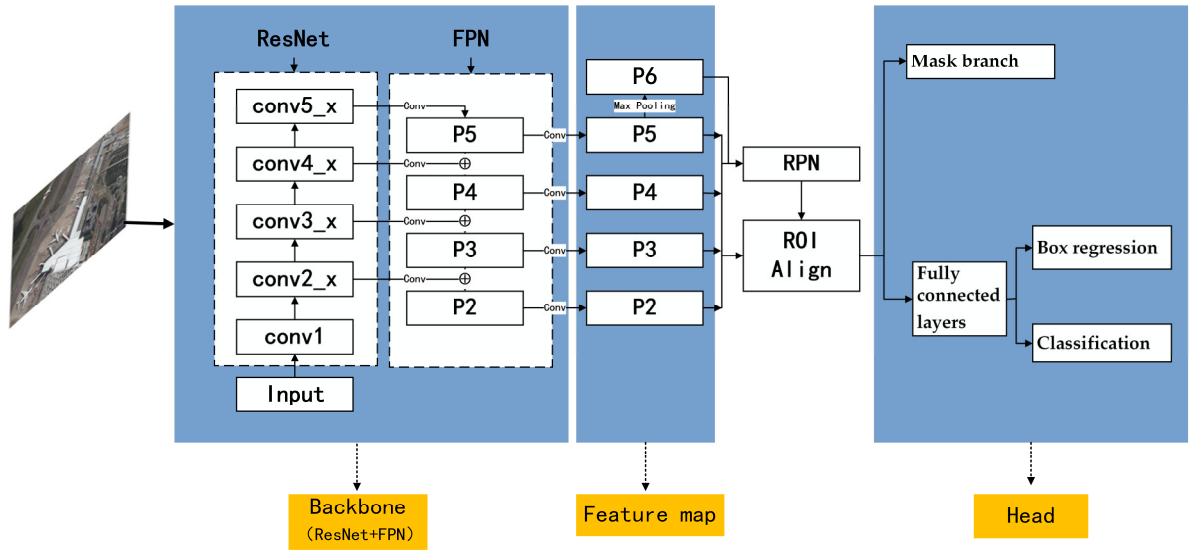


Figure 2. Mask R-CNN model structure. The backbone is composed of ResNet and feature pyramid network (FPN), and four convolution structural blocks in the ResNet constitute four feature maps representing different levels of semantic information. The region proposal network (RPN) proposes candidate object bounding boxes and an RoIAlign layer properly aligning the extracted features with the input. The ROI head is composed of mask branch, box regression, and classification.

$$\begin{cases} P_i = Sum\left(upsample(P_{i+1}), Conv(C_i)\right), \\ P_6 = MaxPooling(P_5) \end{cases} \quad (1)$$

2.2.1. Region Proposal Network (RPN)

The RPN takes five feature maps of P2, P3, P4, P5, and P6 as input and generates rectangular candidate regions of five different scales and three different aspect ratios at each position in a sliding window manner. The default scales are (32, 64, 128, 256, and 512), and the aspect ratios are (0.5, 1.0, and 2.0).

According to the possible combinations of the different scales and aspect ratios listed above, approximately 36,000 candidate regions are generated on the image. The coordinates of each candidate area and the confidence that it is a foreground or background area are calculated, and 6000 candidate areas are reserved in accordance with their degrees of confidence. Finally, by using the strategy of non-maximum suppression (NMS) [15], 2000 ROI regions are obtained.

2.2.2. RoIAlign

In the newly added branch of the mask R-CNN model, an FCN is used to calculate the pixel values of the masks with a threshold of 0.5. The RoIAlign layer generates a 14 × 14 feature map for each ROI and uses a bilinear difference method to calculate the mask boundaries.

2.2.3. PointRend

CNNs for image segmentation typically operate on regular grids. The input image is a regular grid of pixels, the hidden representations are feature vectors on a regular grid, and the outputs are label maps on regular grids [31]. This tends to result in oversampling in low-frequency areas (belonging to the same object) and undersampling in high-frequency areas (the edges of objects). In instance segmentation, the pixels that are most likely to be misjudged by the model are those at the edges of objects [35].

PointRend [31] provides a way to solve the image segmentation problem by treating it as a rendering problem. Efficient procedures such as subdivision [36] and adaptive sampling [37] refine a coarse rasterization in areas where pixel values have larger variance.

Ray-tracing renderers often use oversampling [38], a technique that samples some points more densely than the output grid to avoid aliasing effects. Here, we apply classical subdivision to image segmentation. The fuzzy segmentation points of each target edge are further predicted in a process called fine segmentation. A flowchart of the PointRend algorithm is shown in Figure 3. The main process is as follows:

1. Generate a mask prediction (coarse prediction) from a lightweight coarse mask prediction head.
2. Select the N "points" that are most likely to be different from their surrounding points (such as points at the edge of an object).
3. For each selected point, extract a "feature representation". This feature representation consists of two parts, i.e., fine-grained features, which are obtained through bilinear interpolation on the low-level feature map (similar to RoIAlign), and high-level features (coarse prediction), which are obtained in Step 1.
4. Use a simple multilayer perceptron (MLP) to calculate a new prediction from the feature representations and update coarse prediction_i to obtain coarse prediction_i+1.
5. Repeat Steps 2, 3, and 4 until the pixel requirements are met.

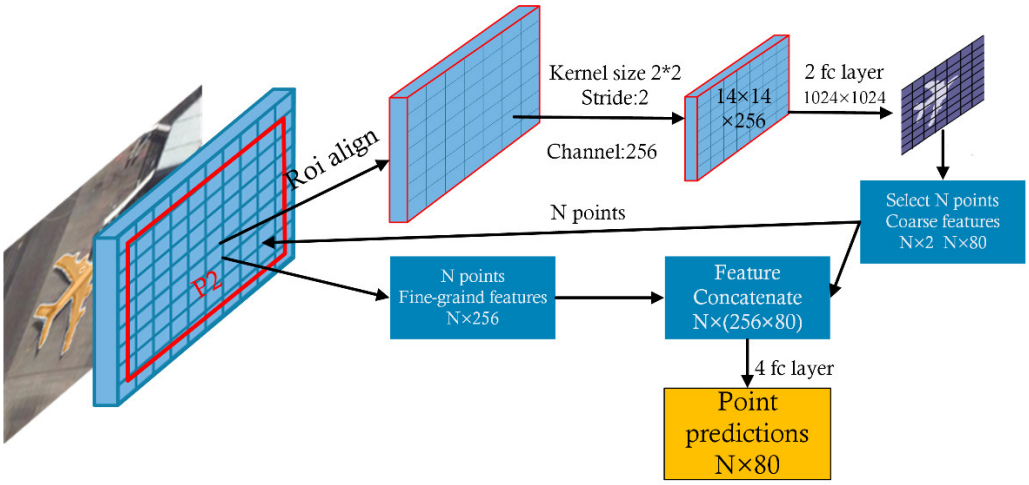


Figure 3. PointRend flowchart.

2.2.4. Deepfill

The GAN-based image inpainting method has the problem that an inpainted area often has a distorted structure or fuzzy texture that is inconsistent with its surroundings. To address this problem, Jiahui Yu et al. [32] proposed a new method based on a deep generative counter-network model in their article "Generative Image Inpainting with Contextual Attention". This model, namely the Deepfill model, can repair missing areas in an image and also use the features of the surrounding parts of the image as a reference during the network training process to extract image content at a longer distance, thereby effectively solving the problems of edge structure distortion and texture blurring in the repaired image.

1. Coarse-refinement two-stage network

The network structure of the Deepfill model, which is shown in Figure 4, consists of two stages, i.e., a coarse network and a refinement network [39].

In the first stage, a CNN with dilated convolution is used to continuously learn, predict, and update the values of the convolutional weights in the missing area to finally obtain coarse repair results. The dilated convolution method can expand the receptive field of a convolution kernel to allow the kernel to capture a larger range of information.

Then, the coarse repair results are passed to the refinement network that serves as the second stage. The refinement network consists of the following two parallel convolutional network branches: One has a dilated convolution structure similar to that of the coarse network and continues the training process based on the results from the first stage, and the other branch has a contextual attention layer structure and matches candidate regions in the original input image that are similar to the preliminary repair results from the coarse network by performing multi-classification. Then, the final repaired image is obtained by merging the results from the two branches and upsampling.

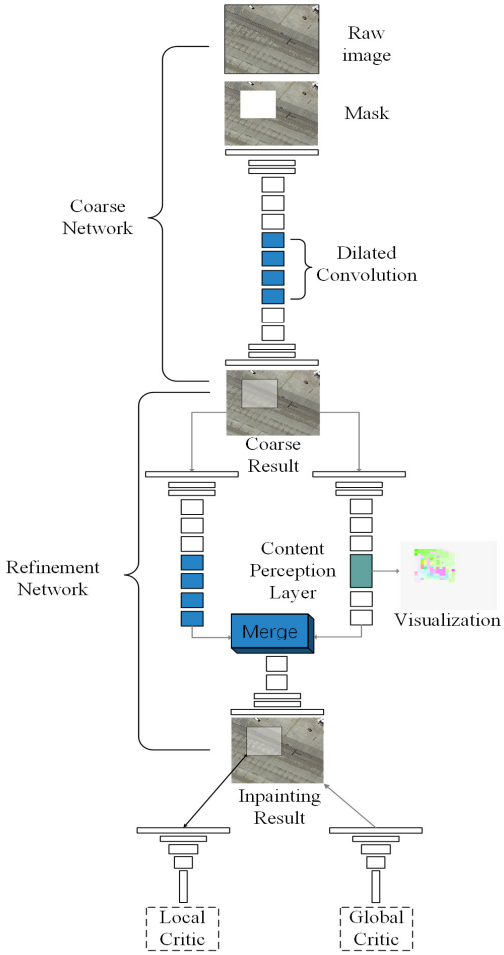


Figure 4. Deepfill model structure. On the basis of the coarse result from the first encoder-decoder network, two parallel encoders are introduced, and then merged to single decoder to obtain the inpainting result.

The entire Deepfill model consists of global and local Wasserstein GAN + gradient penalty (WGAN-GP) [40] networks. The global discriminator is used to evaluate the overall consistency of the repaired image and to determine whether the image has been successfully repaired, while the local discriminator focuses on distinguishing the consistency of each repaired area and its surroundings in the image.

2. Spatial attenuation of the reconstruction loss

In the image repair task, there are many different but acceptable repair results for the area to be repaired. Using only the distance from the original image as the criterion to measure the training loss value may cause a large number of feasible repair solutions to be eliminated, which will make the training of the entire network more difficult and "mislead" the direction of network optimization.

To address these problems, the Deepfill model includes a spatial attenuation mechanism for the reconstruction loss and an attention mechanism [41] that causes the convolutional weight in the repaired area to decrease with increasing distance from the center of the repaired area, and therefore reduces the impact on the loss value when the difference between the center of the repaired area and the original image is too large. Therefore, the training of the entire network is easier and more effective.

2.3. Evaluation Indexes

The main indicators used to evaluate the performance of the detection model are the precision, recall, miss rate, AP_{75}, and F1-score. The N represents the total number of targets in the test set, N_{TP} denotes the number of positive examples detected correctly, and N_{FP} denotes the number of backgrounds misidentified as targets. The precision represents the proportion of real positive examples among the targets judged by the detection model to be positive examples, as shown in Equation (2) as:

$$Precision = \frac{N_{TP}}{N_{TP}+N_{FP}} \quad (2)$$

The recall represents the proportion of positive examples in the test set that are correctly detected, as shown in Equation (3) as:

$$Recall = \frac{N_{TP}}{N} \quad (3)$$

The miss rate measures the degree to which the model misses targets, as shown in Equation (4) as:

$$MR = \frac{N-N_{TP}}{N} \quad (4)$$

Average precision (AP) was used to evaluate both classification and detection for the VOC2007 challenge [42], which summarizes the shape of the precision/recall curve. The AP measure can highlight differences between methods to a greater extent [42]. AP_{75} represents the area under the precise-recall curve drawn when the detected targets with IoU greater than 75% are regarded as correct statistics.

Since the precision and recall are often contradictory, the *F*1-score is introduced to comprehensively measure the precision and recall, to evaluate the model's performance, as shown in Equation (5) as:

$$F1 = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (5)$$

The performance of the hiding model is evaluated using two indicators, i.e., the peak signal-to-noise ratio (*PSNR*) and the structural similarity (SSIM). The PSNR is defined in terms of the mean square error (*MSE*). For a given $m \times n$ image I and a noisy image K, the *MSE* is defined as follows:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \quad (6)$$

The definition of the PSNR is as follows:

$$PSNR = 10.log_{10}\frac{MAX_I^2}{MSE} \quad (7)$$

where MAX_I^2 represents the maximum possible pixel value in the image. This article considers red-green-blue (RGB) color images, and the final *PSNR* is calculated by calculating the PSNR of each of the three channels (R, G, and B) and taking the average value.

The SSIM is an index for comparing the similarity of two images x and y in terms of three aspects, i.e., brightness, contrast, and structure. μ_x is the mean of x, μ_y is the mean of y, σ_x^2 is the variance of x, σ_y^2 is the variance of y, and σ_{xy} is the covariance of x and y. $c_1 = (k_1\mathrm{L})^2$ and $c_2 = (k_2\mathrm{L})^2$ are constants, where L is the range of the image pixel values. $k_1 = 0.01$, and $k_2 = 0.03$. Generally, $c_3 = \frac{1}{2}c_2$. SSIM is obtained as follows:

$$\begin{cases} l(x,y) = \frac{2\mu_x\mu_y+c_1}{\mu_x^2+\mu_y^2+c_1} \\ c(x,y) = \frac{2\sigma_x\sigma_y+c_2}{\sigma_x^2+\sigma_y^2+c_2} \\ s(x,y) = \frac{\sigma_{xy}+c_3}{\sigma_x+\sigma_y+c_3} \end{cases} \quad (8)$$

$$SSIM(x,y) = [l(x,y)^{\rho}c(x,y)^{\omega}s(x,y)^{\tau}] \quad (9)$$

By setting ρ, ω, and τ equal to 1, we can obtain the following:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{(\mu_x^2+\mu_y^2+c_1)(\sigma_x^2+\sigma_y^2+c_2)} \quad (10)$$

3. Application of the Two-Stage Processing Model with Aircraft Targets as an Example

We take aircraft target processing in remote sensing mapping as an example to verify the performance of the proposed two-stage processing model for target detection and hiding.

3.1. Aircraft Target Characteristics

Remote sensing images contain rich spatial information, spectral features, and texture information, as well as various target categories [43]. Aircraft targets have the following four main types of distinctive characteristics: ① Spectral characteristics, i.e., brightness ratio information in terms of color and gray level. The color of an aircraft target is different from the colors of natural features and thus can be used as a typical feature for recognition. ② Texture features, i.e., regular and repeated changes in the gray level of the target within a certain spatial range. ③ Shape features, i.e., the shape of the aircraft boundary and area, including characteristic scales and aspect ratios. The shape features of most targets are within a small range and are affected by factors such as the height from which the image was acquired and the side angle of view. The scale conversion range is large, and large-scale targets also exist. These features serve as an important basis for identifying and detecting aircraft targets in this study. ④ Contextual features, i.e., the spatial relationships between various objects in the image. In particular, the aircraft targets considered in this article are generally located at an airport [44].

3.2. Reconstruction Loss Function of the Detection Model

The task of target detection in emergency remote sensing mapping is more difficult than conventional image target detection tasks. According to the above analysis of aircraft target characteristics, the detection model should have the following characteristics: ① a high-precision extraction capability for small aircraft targets in multi-scene, multi-information high-resolution remote sensing images and ② generalizability and compatible extraction capabilities for a few large aircraft targets with relatively high pixel ranges.

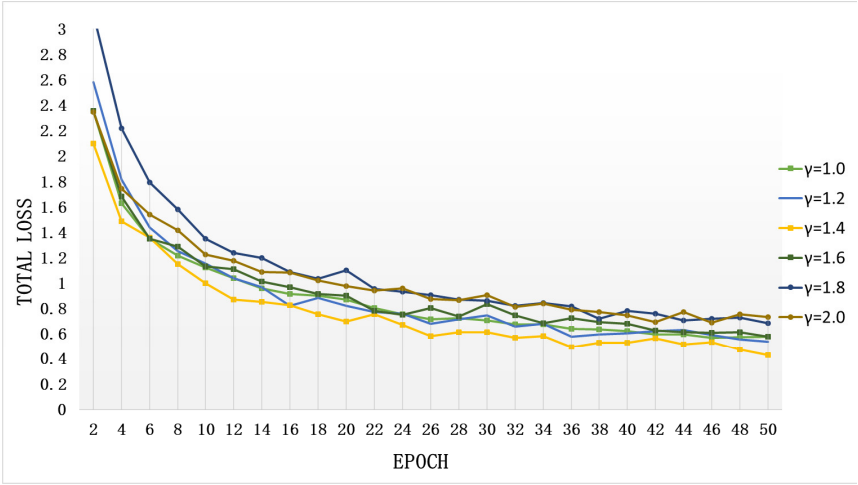
According to the mask R-CNN model, we adopt corresponding optimization strategies to improve the accuracy and robustness of the model for aircraft target detection to better satisfy the mission requirements.

3.2.1. Reconstruction Loss Function

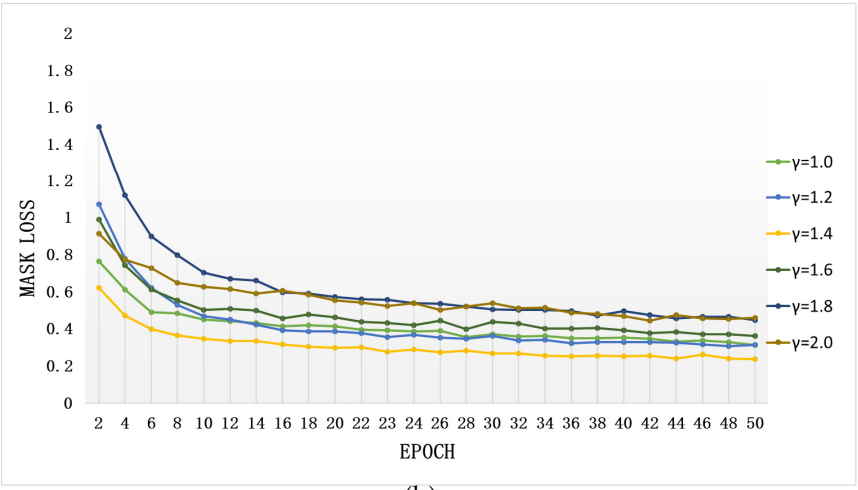
The loss function of the mask R-CNN model is defined as follows: classification loss + bounding box loss + mask loss. The mask quality has a significant impact on the range of each hidden area and the results of target-hiding processing. Therefore, this paper introduces the parameter γ, as shown in Equation (11), to increase the contribution of $Lmask$ to the loss function, adjust the direction of optimization during model training, and improve the mask quality, where γ is a constant greater than 1, as follows:

$$Loss = Lcls + Lbox + \gamma Lmask \quad (11)$$

This modification to the reconstruction loss function increases the model training cost and the difficulty of fitting. We selected multiple sets of parameters for comparison of the training results and comprehensively calculated the optimal value of γ, according to the results for the total loss value and the mask loss value. Figure 5 shows the loss curves corresponding to model training with γ values of 1.0, 1.2, 1.4, 1.6, 1.8, and 2.0. The top panel shows the total loss, and the bottom panel shows the mask loss. According to the results, γ should be set to 1.4.



(a)



(b)

Figure 5. Loss curves trained with different values of the γ parameter. (**a**,**b**) shows the change trend of total loss and the mask loss respectively.

3.2.2. Region Proposal Network (RPN) Optimization

The RPN generates different anchor boxes depending on the input aspect ratio and scale parameters. Considering the characteristics of the semantic information associated with aircraft, we modified the aspect ratio and scale parameters to match the shape characteristics of aircraft targets to reduce the redundancy in the number of anchor boxes, reduce the amount of calculation, and improve the hit rate of the anchor boxes.

According to an analysis of the statistical results shown in Table 1, the aspect ratios of common models of aircraft lie in the range of 0.9 to 1.25. Therefore, we set the anchor box aspect ratio coefficient α to $\alpha \in$ (0.8, 1.0, 1.25). Since most aircraft targets in remote sensing images are small targets, after comparing the detection results at multiple different scales, we set the scale parameter β to $\beta \in$ (8,32,64,128,256).

Table 1. Common types of aircraft parameters [45,46].

Aircraft Type	Aircraft Length (m)	Wing Length (m)	Aspect Ratio
BOEING 737-300	33.40	28.90	1.16
BOEING 737-700	33.60	34.30	0.98
BOEING 737-800	39.50	34.30	1.15
BOEING 747-300	70.60	59.60	1.18
BOEING 747-400	70.60	64.40	1.10
BOEING 747-800	76.40	68.50	1.12
BOEING 767-200	48.51	47.57	1.02
BOEING 777-200	63.73	60.93	1.05
BOEING 787-8	56.69	60.17	0.94
BOEING 787-9	63.00	60.17	1.05
BOEING 787-10	68.00	60.17	1.13
AIRBUS A300	54.10	44.84	1.21
AIRBUS A318-100	31.45	34.10	0.92
AIRBUS A319-100	33.84	34.10	0.99
AIRBUS A320-100	37.57	34.10	1.10
AIRBUS A330-200	59.00	60.30	0.98
AIRBUS A330-300	63.60	60.30	1.05
AIRBUS A340-200	59.40	60.30	0.99
AIRBUS A340-300	63.60	60.30	1.05
AIRBUS A340-500	67.90	63.50	1.07
AIRBUS A340-600	75.30	63.50	1.19
AIRBUS A350-800	60.50	64.00	0.95
AIRBUS A350-900	66.80	64.00	1.04
AIRBUS A350	73.80	64.00	1.15
AIRBUS A380-800	73.00	79.80	0.91

3.3. Mask R-CNN + PointRend

3.3.1. Model Structure

The mask R-CNN model predicts masks on a 14 × 14 (faster-rcnn-c4) or 28 × 28 (faster-rcnn-fpn) grid regardless of the object size. Consequently, for large aircraft targets, many details are lost due to high scaling. Therefore, in this experiment, we replaced the mask head in the original mask R-CNN with the PointRend algorithm, as shown in Figure 6. For the coarse features, we used a network structure similar to that of the mask head in the original mask R-CNN and finally obtained a 7 × 7 coarse segmentation map. On this basis, we sampled N points and their coarse features; then, we obtained the fine-grained features of these N points from the original CNN P2 layer. Finally, we concatenated the two sets of features to obtain the feature representations of the candidate points.

Figure 7 presents a comparative example of target detection after the addition of PointRend. The mask generated by the original mask R-CNN has dimensions of 28 × 28. Since the maximum resolution of the aircraft in the remote sensing image is approximately 200 × 200, we set the resolution of the PointRend output mask to 224 × 224. The images on the left show the original model predictions, and the images on the right are the predicted images generated after the addition of PointRend. We can clearly see that the fit of the edges has improved.

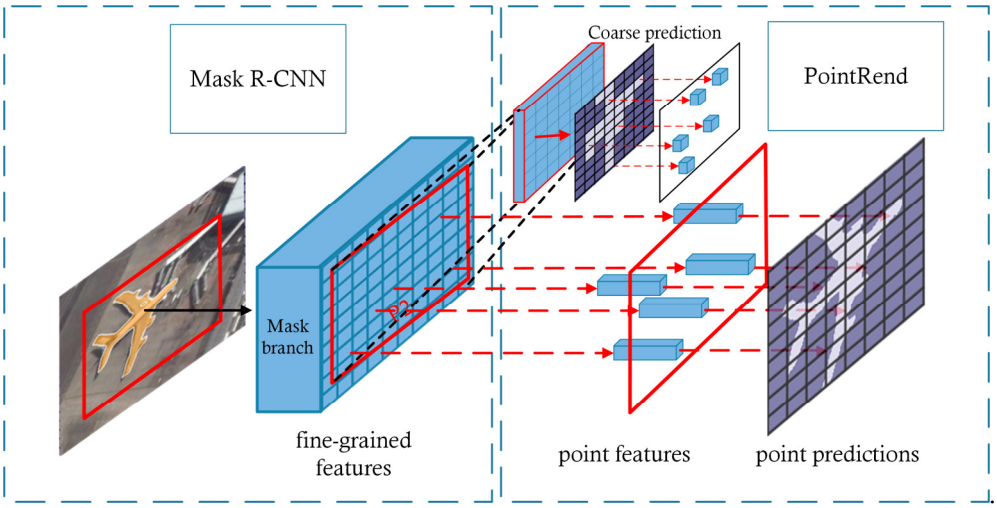


Figure 6. Mask R-CNN + PointRend structure.

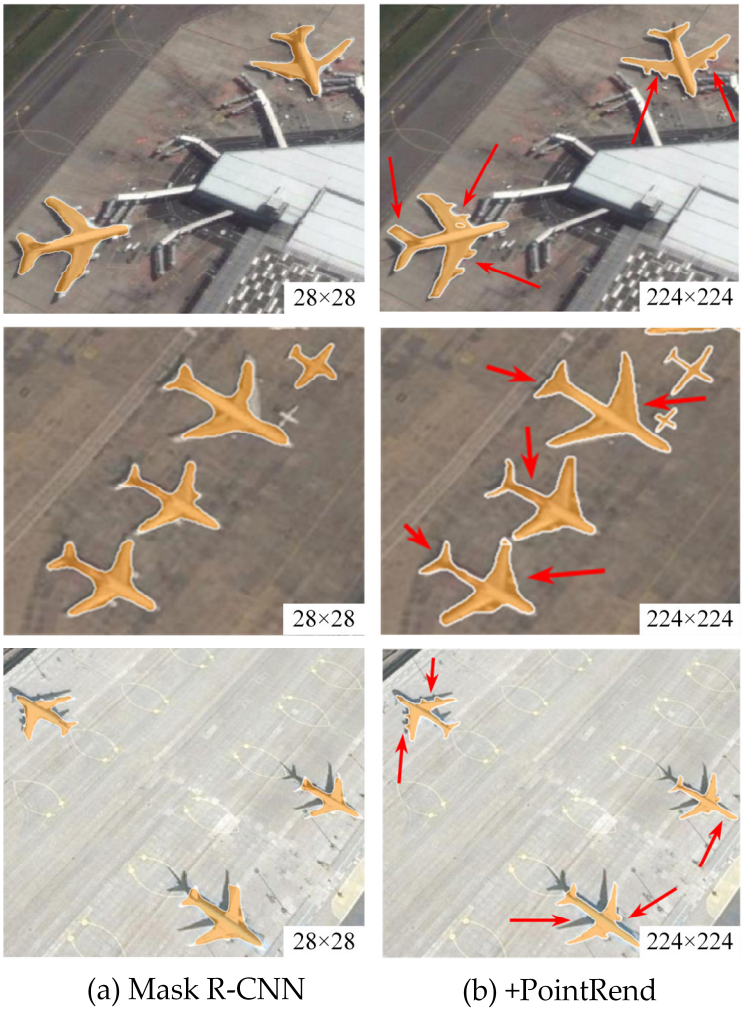


Figure 7. Comparison of Mask R-CNN vs. PointRend instance segmentation results. Example result pairs from Mask R-CNN with its standard mask head (**a**) vs. with PointRend (**b**), note how PointRend predicts masks with substantially finer detail around object boundaries

3.3.2. Mask Optimization

Limited by the quality of the mask output and the labeling quality of the training data, the original model is susceptible to a phenomenon in which some target masks may

be incomplete and fail to completely cover the target area, which increases the difficulty of hiding processing and results in poor target-hiding effects. To improve the detection process, we added a mask optimization algorithm to the mask output layer of the original model, as shown in Figure 8.

The core idea of the optimization algorithm is inspired by the dilation algorithm in image morphology [47]. In Equation (12), A is the target mask output by the original model. A is used to determine the target anchor point, and the diffusion coefficient and the convolution kernel B are determined according to the target pixel value. Then, the convolution kernel B is used to convolve A such that the mask is expanded outward along the target contour to improve the quality of the target mask and its degree of target coverage as follows:

$$A * B = \bigcup_{b \in B} A_b \tag{12}$$

Figure 9 shows a comparative example of masks generated by the original model and the model with optimization; from left to right are the original image, the mask generated by the original model, and the mask generated by the model with optimization. We can see that the mask generated by the model with optimization is of better quality, covers a higher proportion of the target, and achieves a good coverage effect for the shadows generated by the target due to factors such as the shooting angle and height of the remote sensing platform.

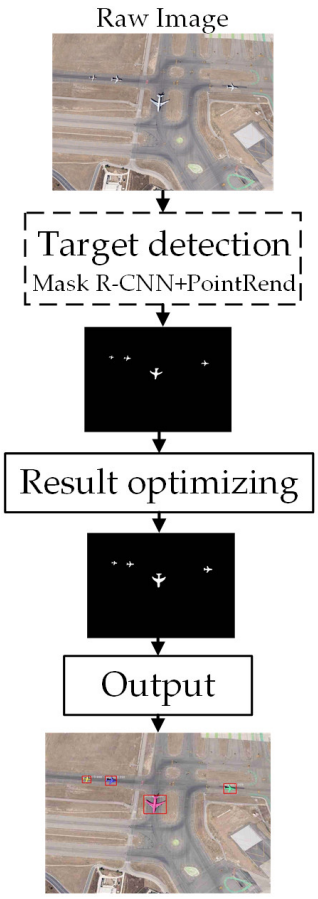


Figure 8. Detection process after adding mask optimization algorithm.

Figure 9. Mask result comparison. (**a**) is the original image shows 2 aircrafts, (**b**,**c**) show the mask generated by the original model, and the mask generated by the model with optimization respectively. Note that image (**c**) is of better quality, covers a higher proportion of the target, and achieves a good coverage effect for the shadows.

3.3.3. Modification of the Hiding Model Attention Mechanism

The target-hiding process in a remote sensing image consists of generating background content to fill in the area to be hidden. The internal structure and texture of the background area have a high degree of similarity, while the edge regions are obviously different from other background or target areas.

Therefore, in the hiding processing task, more attention should be paid to the fusion of the background generated in the area to be hidden with the surrounding background to make the structure and texture more consistent. In this paper, the attention mechanism strategy of the original model is adopted, and the strategy for calculating the matrix M is modified according to the needs of hiding processing. In Equation (13), θ is set to 0.9 and l_i is the L2 distance from the current point to the known pixel point (x_0, y_0). We increase the weight in boundary regions, weaken the influence of the center of the area on the model training process, and improve the degree of fusion between the boundary of the area to be hidden and the surrounding background to make the overall visual effect of the image more reasonable and natural.

$$\begin{cases} M_i = \theta^{l_i^2} \\ l_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} \end{cases} \quad (13)$$

4. Experiments and Analysis

4.1. Data Collection and Preprocessing

The experimental data come from the Remote Sensing Object Detection (RSOD) dataset and the remote sensing Dataset for Object Detection in Aerial images (DOTA) [48] annotated by Wuhan University.

To improve the generalizability and robustness of the model, during the process of dataset construction, the samples for training the detection model were mainly drawn from the RSOD dataset, and a small number of samples were randomly selected from DOTA; by contrast, the samples for training the hiding model were mainly drawn from DOTA, and a small number of samples from the RSOD dataset were randomly added.

Preprocessing of the detection model training data was completed as follows: ① The sample data were normalized to 1024 × 800 pixels to facilitate labeling and training. ② LabelMe was used to relabel the aircraft targets in the original samples to construct a new remote sensing image aircraft target instance segmentation dataset. ③ The dataset was randomly divided into a training set and a validation set at a ratio of 0.8:0.2. The test set was randomly selected and stored separately from the original samples. The samples in

the test set were not used in the training process and were used only to evaluate the quality of the results of the trained model.

A total of 210 images were labeled, and data augmentation operations such as inversion, left and right mirroring, flipping, and rotation were applied. The total number of samples was 1607, with 1285 samples in the training set, and 322 samples in the validation set. Figure 10 shows several examples of target detection samples.

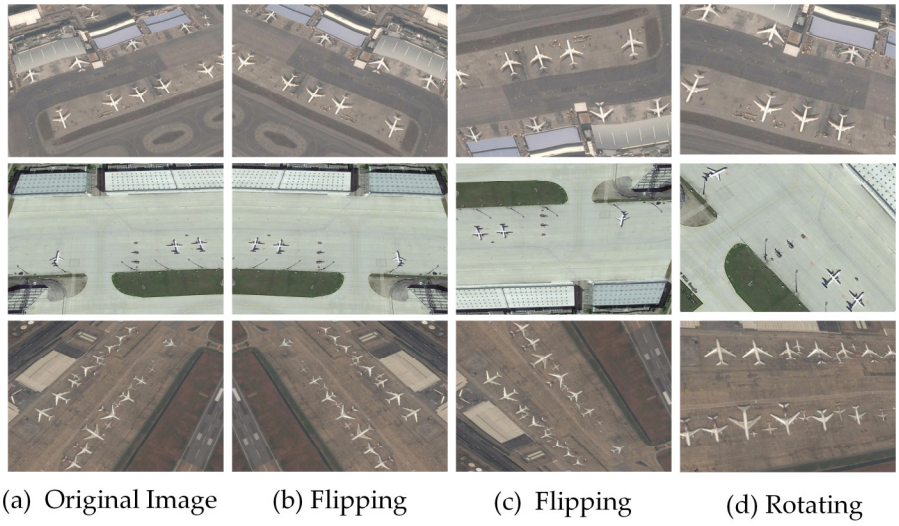


Figure 10. Examples of target detection samples**.** Images (**b**) are mirrored left and right from images (**a**); images (**c**) are obtained by flipping images (**b**) upside down; images (**d**) are generated by rotating images (**a**) by 45 degree.

Preprocessing of the hiding model training data was completed as follows: The method of directly randomly generating masks from large-format remote sensing image samples cannot satisfy the requirements for training a model to hide only specific target regions. Therefore, we cropped the aircraft and airport samples in the dataset to 256 × 256 pixels and retained only image samples depicting airport and runway background areas.

A total of 9502 samples were obtained and randomly divided into a training set and a validation set at a ratio of 0.8:0.2. The training set contained 7601 images, and the validation set contained 1901 images.

4.2. Training

The training methods and loss function calculations used for the two stages of the model are different; therefore, we trained and tuned the two models separately, and then combined them to obtain a two-stage trained model, which we applied for the tasks of the automatic detection and hiding of aircraft targets.

In this study, we utilized Pytorch as our deep learning framework. All experiments were performed on computer equipped with a 64-bit Intel i7-6700K CPU @ 4.0 GHz, 32 GB of RAM, and a GeForce GTX1070 GPU with 4 GB of memory, running under CUDA version 10.0. The operating system was Ubuntu 16.04 LTS.

The experimental parameters for the training of the detection model were set as follows: the learning rate was 0.0001, the batch size was 2, and the number of epochs was 50. After the corresponding pretrained model was obtained, 46 remote sensing image samples were randomly selected from the test set to verify the model's detection performance.

The experimental parameters for the training of the intelligent hiding model were set as follows: the batch size was eight, the number of training times per epoch was 2000, the maximum number of iterations was 1,000,000, and the number of validation times per epoch was 200. The losses in the coarse network and the refinement network were both

reduced to 0.5, and the maximum width and height parameters of the mask range were set to 256 pixels.

4.3. Analysis of Results

4.3.1. Model Performance Analysis

For the detection model, we randomly selected 46 images from the test set for detection to measure the performance of the model. The results are shown in Table 2. The actual number of targets is 475. The original mask R-CNN model detects a total of 214 targets, of which 210 are correct; the model proposed in this paper detects 411 targets, of which 389 are correct.

As shown in Figure 11, the precision of the original mask R-CNN model is 98.13%, while the recall is only 44.21% and the AP_{75} is 65.4%. Consequently, the F1-score is only 60.96% because a large number of targets are not detected. After optimization, the precision of the detection model proposed in this paper is still as high as 94.87%, while the recall reaches 81.68% and the AP_{75} reaches 79.2%, which are 84.75% and 21.1% higher than those of the original mask R-CNN model, respectively. Accordingly, the F1-score reaches 87.78%, which is 44% higher than that of the original model.

Table 2. Common types of aircraft parameters.

Indicators	Mask R-CNN	Optimized Model
Number of test images	46	46
True target number	475	475
Number of detected targets	214	411
Number of correct detected targets	210	389
Number of missed targets	265	86
Bounding box AP_{75}	65.4%	79.2%

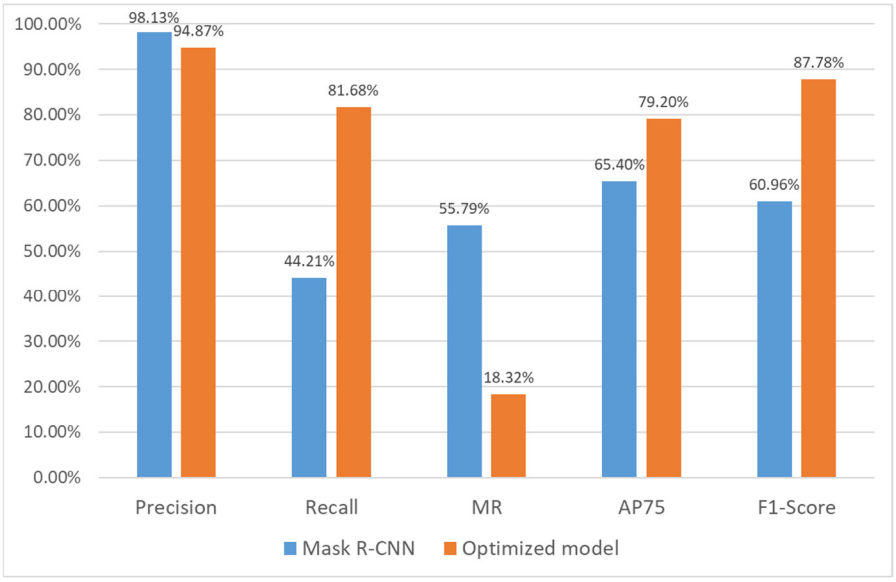


Figure 11. Detection model performance comparison.

For the hiding model, we tested the model on the 46 images from the test set subjected to target detection processing as input images. The PSNR and SSIM values are shown in Figure 12. The average PSNR value is 32.26 and the average SSIM value is 0.98.

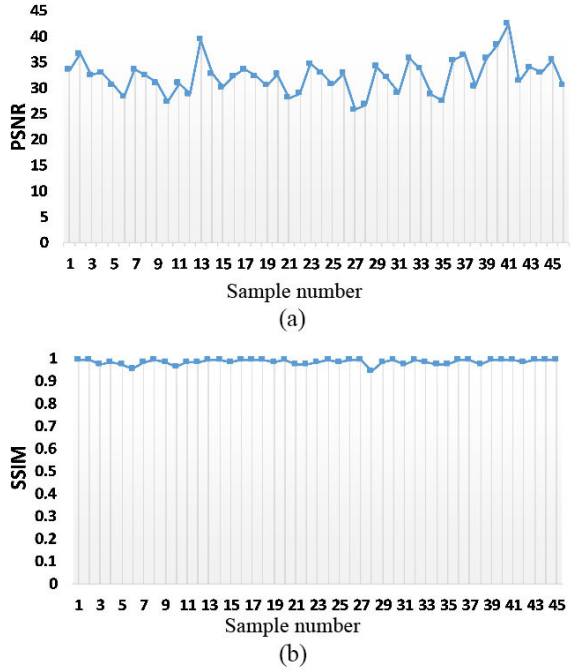


Figure 12. The peak signal-to-noise ratio PSNR and structural similarity (SSIM) results. Image (**a**) shows the PSNR of the 46 sample images, the average value is 32.26. Image (**b**) shows the SSIM, the average value is 0.98, which means the generated images are very similar to the non- classified images.

4.3.2. Comparative Analysis of Results

Figure 13 presents comparative examples of the results of target detection and hiding processing; from left to right are the original images, detection results, and hiding processing results. This figure shows three different sets of images. The first row in each group shows the processing results of the original model, and the second row shows the processing results of our method. We can see that the two-stage model proposed in this article ① achieves better detection and recognition effects and a higher recall rate for aircraft targets in remote sensing images and ② shows a more powerful hiding processing ability and a more significant background restoration effect for the detected targets.

In terms of quality, the benchmark model obviously suffers from the phenomenon that the final hiding processing may fail due to missed targets. Moreover, even for targets that are successfully detected, the generated masks sometimes do not completely cover the target areas; as a result, the aircraft texture structure is still evident after hiding processing. By contrast, the proposed two-stage model can output high-quality masks, which, combined with the powerful performance of the hiding processing model, enable more successful hiding of sensitive targets with better fusion with the surrounding scene.

In terms of efficiency, it takes approximately 15 min to process a 1:10,000 image with a size of 1 GB and a resolution of 0.2 m using our method. Depending on the operating conditions of the production unit, when the manual processing method is adopted, even when the time-consuming task of screening for sensitive targets is not considered, hiding processing alone typically takes 30 min. Thus, the processing time of our method is reduced by more than 50% as compared with that of manual processing, demonstrating that our method can greatly accelerate the speed of emergency mapping.

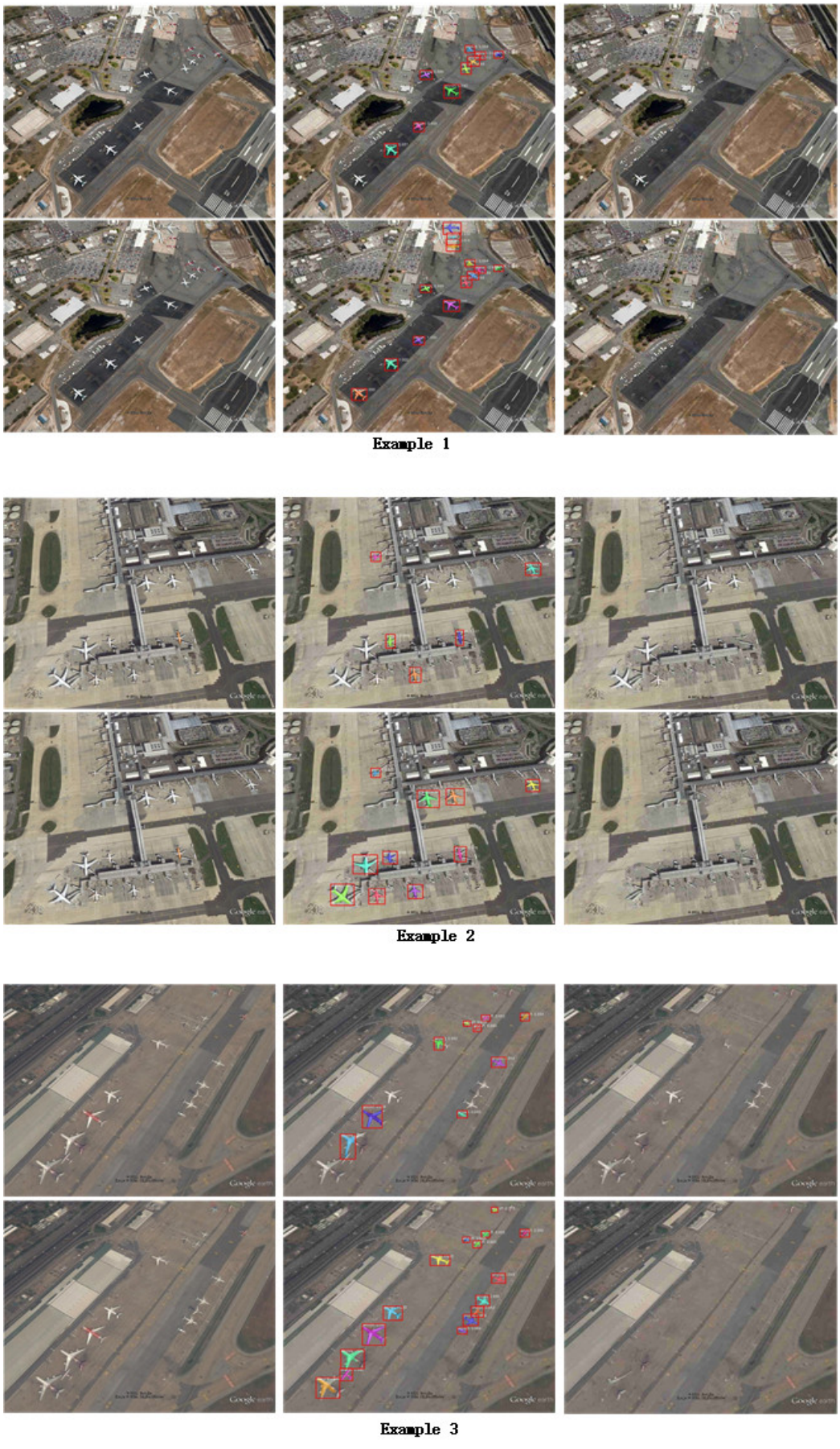


Figure 13. Comparison of object detection and hiding results.

4.4. Discussion of Experimental Results

The detection and hiding of sensitive targets are among the important links in the field of remote sensing emergency mapping. This study addressed this problem with a two-stage processing model combining deep learning methods and expanded the ideas and methods of related research in this field. First, this article proposes a complete model framework, including two stages of target detection and hiding, which basically realizes the automatic hiding of sensitive targets. Compared with the existing purely manual or semi-automatic methods, our proposed model framework guarantees a high accuracy rate and also greatly improves efficiency and meets the goal of rapid and accurate emergency mapping. Second, the detection accuracy of sensitive targets basically determines the result of target hiding, and therefore improving the accuracy of the detection stage is the key to optimizing the entire model. We combined the PointRend method on the basis of mask R-CNN, and this optimized model improved the detection accuracy of sensitive targets so that subsequent target hiding has a better effect. The optimized detection model can be extended to other similar cases. In addition, we applied the proposed target detection and hiding model to actual remote sensing data and achieved satisfactory results, which can basically replace manual work and realize automatic processing. This shows that the model we proposed has good practical application value and scalability and can be used in industrial production. In addition to the airplane case used in this article, it can be used to detect and hide other similar sensitive targets, such as warships, military depots, and military training grounds under the premise that the training dataset is large enough.

5. Conclusions

In our method, the detection recall, AP_{75} and F1-score for aircraft targets in remote sensing images are significantly improved, and the effect of hiding processing is reasonable and natural. Moreover, significant time is saved in the overall remote sensing mapping process, thus, demonstrating the practical applicability of the proposed two-stage model. Theoretically, this method is also suitable for the automatic detection and hiding of other single-type sensitive targets in emergency remote sensing mapping.

Despite being effective, some restrictions of our method still exist. The dataset sample size is not large enough and traditional data augmentation methods are used. Thus, the precision is not good enough, and the model lacks comparison with other excellent models. In the future, zero-shot and GAN can be used to solve the problem of limited available training data. More current CNN models for our aim of target detection will be considered, especially one-stage object detection models, such as RetinaNet [49], SSD, and YOLOv3 [50].

Author Contributions: Conceptualization, Tianqi Qiu, Qingyun Du, Fu Ren and Xiaojin Liang; Data curation, Pengjie Lu; Funding acquisition, Qingyun Du; Investigation, Xiaojin Liang and Pengjie Lu; Methodology, Tianqi Qiu, Qingyun Du, Fu Ren, Xiaojin Liang and Chao Wu; Resources, Qingyun Du; Software, Tianqi Qiu; Supervision, Qingyun Du; Validation, Xiaojin Liang; Visualization, Pengjie Lu; Writing—original draft, Tianqi Qiu; Writing—review and editing, Tianqi Qiu, Qingyun Du, Xiaojin Liang and Chao Wu. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement: Restrictions apply to the availability of these data. Data are classified and not allowed to be published publicly.

Acknowledgments: We would like to acknowledges the National Key R&D Program of China and the National Natural Science Foundation of China for the financial support. We also would like to thank the editor and three anonymous reviewers for their critical comments and constructive suggestions.

Funding: This work was supported by the National Key R&D Program of China (project no. 2016YFC0803106) and the National Natural Science Foundation of China (project no. 41571438).

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

References

1. Zhu, Q.; Cao, Z.; Lin, H.; Xie, W.; Ding, Y.L. Key technologies of emergency surveying and mapping service system. *Geomat. Inf. Sci. Wuhan Univ.* **2014**, *39*, 551–555.
2. Yang, K.; Li, Q.; Fang, S. Based on remote sensing drawing monitoring, evaluate and father flood. *Map* **1998**, *4*, 22–23. (In Chinese)
3. Fan, Y.D.; Yang, S.Q.; Wang, L.; Wang, W.; Nie, J.; Zhang, B.J. Study on urgent monitoring and assessment in Wenchuan earthquake. *J. Remote Sens.* **2008**, *12*, 858–864.
4. Xu, T.; Zhang, J. Implementation of remote sensing automatic mapping used for earthquake emergency. *J. Nat. Disasters* **2017**, *26*, 19–27.
5. Hacıefendioğlu, K., Başağa, H.B., Demir, G. Automatic detection of earthquake-induced ground failure effects through faster R-CNN deep learning-based object detection using satellite images. *Nat. Hazards* **2021**, *105*, 383–403.
6. Ghorbanzadeh, O.; Meena, S.R.; Abadi, H.S.S.; Piralilou, S.T.; Zhiyong, L.; Blaschke, T. Landslide mapping using two main deep-learning Convolution Neural Network (CNN) streams combined by the dempster—Shafer (DS) model. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *14*, doi:10.1109/JSTARS.2020.3043836.
7. Chen, B. Law of Surveying and Mapping of the People's Republic of China. Available online: http://www.asianlii.org/cn/legis/cen/laws/samlotproc506/ (accessed on 5 February 2021).
8. Order of the State Council of the People's Republic of China. Available online: http://www.gov.cn/zwgk/2014-02/03/content_2579949.htm (accessed on 9 February 2021).
9. Dalal, N. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005.
10. Lowe, D.G. Distinctive image features from scale-invariant key points. *Int. J. Comput. Vis.* **2004**, *60*, 91–110.
11. Ojala, T.; Pietikinen, M.; Menp, T. Gray scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *24*, 971–987.
12. Bay, H.; Ess, A.; Tuytelaars, T.; Gool, L.V. Speeded-Up Robust Features (SURF). *Comput. Vision Image Underst.* **2008**, *110*, 346–359.
13. Wojek, C.; Schiele, B. A performance evaluation of single and multi-feature people detection. In *Joint Pattern Recognition Symposium*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 82–91.
14. Dollár, P.; Tu, Z.; Perona, P.; Belongie, S. Integral channel features. In Proceedings of the British Machine Vision Conference, London, UK, 7–10 September 2009; BMVC: London, UK, 2009.
15. Zhang, Z.; Tao, W.; Sun, K.; H, W.; Yao, L. Pedestrian detection aided by fusion of binocular information. *Pattern Recognit.* **2016**, *60*, 227–238.
16. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
17. 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.
18. He, K.; Georgia, G.; Piotr, D.; Ross, G. Mask R-CNN. In Proceedings of the Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2961–2969.
19. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
20. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.; Berg, A.C. SSD: Single shot multi box detector. In Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
21. Lima, P.D.; Sensing, M.J.R. Convolutional neural network for remote-sensing scene classification: Transfer learning analysis. *Remote Sens.* **2020**, *12*, 86.
22. Fan, H.; Gui-Song, X.; Jingwen, H.; Liangpei, Z.J.R.S. Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. *Remote Sens.* **2015**, *7*, 14680–14707.
23. Li, A.; Lu, Z.; Wang, L.; Xiang, T.; Wen, J.R. Zero-shot scene classification for high spatial resolution remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 4157–4167.
24. Chen, H.; Luo, Y.; Cao, L.; Zhang, B.; Ji, R. Generalized zero-shot vehicle detection in remote sensing imagery via coarse-to-fine framework. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19), Macao, China, 10–16 August 2019; pp. 687–693.
25. Hoeser, T.; Kuenzer, C.J.R.S. Object detection and image segmentation with deep learning on earth observation data: A review—Part I: Evolution and recent trends. *Remote Sens.* **2020**, *12*, 1667.
26. Hoeser, T.; Bachofer, F.; Kuenzer, C.J.R.S. Object detection and image segmentation with deep learning on earth observation data: A review—Part II: Applications. *Remote Sens.* **2020**, *12*, 3053.
27. Lu, C. *Research on Remote Sensing Image Inpainting Technology*; PLA Information Engineering University: Zhengzhou, China. 2011.

28. Yin, Y.; Li, D.; Hu, L. Adaptive image inpainting algorithm based on CDD model. *J. Chongqing Univ.* **2013**, *36*, 80–86.
29. Barnes, C.; Shechtman, E.; Finkelstein, A.; Dan, B.G. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.* **2009**, *28*, 24.
30. Kirillov, A.; Wu, Y.; He, K.; Girshick, R. PointRend: Image segmentation as rendering. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 9799–9808.
31. Yu, J.; Lin, Z.; Yang, J.; Shen, X.; Lu, X.; Huang, T.S. Generative image inpainting with contextual attention. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5505–5514.
32. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
33. 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 (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
34. Shelhamer, E.; Long, J.; Darrell, T. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 640–651, doi:10.1109/TPAMI.2016.2572683.
35. Li, X.; Liu, Z.; Luo, P.; Loy, C.C.; Tang, X. Not all pixels are equal: Difficulty-aware semantic segmentation via deep layer cascade. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp.3193–3202.
36. Whitted, T. An improved illumination model for shaded display. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Courses*; Association for Computing Machinery: New York, NY, USA, 1979; Volume 13, p. 14.
37. Mitchell; Don, P. Generating anti-aliased images at low sampling densities. In *Proceedings of the 14st Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH, August 1987*; Association for Computing Machinery: New York, NY, USA, 1987.
38. Zhou, K.; Hou, Q.; Wang, R.; Guo, B. Real-time KD-tree construction on graphics hardware. *ACM Trans. Graph.* **2008**, *27*, 1–11.
39. Iizuka, S.; Simoserra, E.; Ishikawa, H. Globally and locally consistent image completion. *ACM Trans. Graph.* **2017**, *36*, 107.
40. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A. Improved training of Wasserstein GANs. *arXiv* **2017**, arXiv:1704.00028.
41. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. *Eur. Conf. Comput. Vis.* **2018**, 3–19.
42. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338.
43. Zhu, X.X.; Tuia, D.; Mou, L.; Xia, G.; Zhang, L.; Xu, F.; Fraundorfer, F. Deep learning in remote sensing: A review. *IEEE Geosci. Remote Sens. Mag.* **2017**, *5*, 8–36.
44. Han, X. Study on Key Technology of Typical Targets Recognition from Large-field Optical Remote Sensing Images. Ph.D. Dissertation, Harbin Institute of Technology, Harbin, China, 2013.
45. Boeing. Available online: http://www.boeing.cn/ (accessed on 5 February 2021).
46. Airbus. Available online: https://www.airbus.com/ (accessed on 5 February 2021).
47. Zhe, W.; Jiexian, Z.; Qiqi, G. Aircraft target recognition in remote sensing images based on saliency images and multi-feature combination. *J. Image Graph.* **2017**, *22*, 532–541.
48. 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 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA 18–23 June 2018; pp. 3974–3983.
49. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *42*, 318–327.
50. Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.