applied sciences

MDPI

Article

Syn2Real Detection in the Sky: Generation and Adaptation of Synthetic Aerial Ship Images

Yaoyuan Wu [1], Weijie Guo [1], Zhuoyue Tan [1], Yifei Zhao [2], Quanxing Zhu [2], Liaoni Wu [3] and Zhiming Guo [3,*]

[1] Institute of Artificial Intelligence, Xiamen University, Xiamen 361005, China; wjyxmu@stu.xmu.edu.cn (Y.W.); ccgwj@stu.xmu.edu.cn (W.G.); tanzhuoyue@stu.xmu.edu.cn (Z.T.)
[2] College of The Environment & Ecology, Xiamen University, Xiamen 361102, China; yifeizhao@stu.xmu.edu.cn (Y.Z.); starz@stu.xmu.edu.cn (Q.Z.)
[3] School of Aerospace Engineering, Xiamen University, Xiamen 361102, China; wuliaoni@xmu.edu.cn
* Correspondence: guozm@xmu.edu.cn

Abstract: Object detection in computer vision requires a sufficient amount of training data to produce an accurate and general model. However, aerial images are difficult to acquire, so the collection of aerial image datasets is a priority issue. Building on the existing research on image generation, the goal of this work is to create synthetic aerial image datasets that can be used to solve the problem of insufficient data. We generated three independent datasets for ship detection using engine and generative model. These synthetic datasets are rich in virtual scenes, ship categories, weather conditions, and other features. Moreover, we implemented domain-adaptive algorithms to address the issue of domain shift from synthetic data to real data. To investigate the application of synthetic datasets, we validated the synthetic data using six different object detection algorithms and three existing real-world, ship detection datasets. The experimental results demonstrate that the methods for generating synthetic aerial image datasets can complete the insufficient data in aerial remote sensing. Additionally, domain-adaptive algorithms could further mitigate the discrepancy from synthetic data to real data, highlighting the potential and value of synthetic data in aerial image recognition and comprehension tasks in the real world.

Keywords: Syn2Real; synthetic dataset; ship detection; aerial images

check for updates

Citation: Wu, Y.; Guo, W.; Tan, Z.; Zhao, Y.; Zhu, Q.; Wu, L.; Guo, Z. Syn2Real Detection in the Sky: Generation and Adaptation of Synthetic Aerial Ship Images. *Appl. Sci.* **2024**, *14*, 4558. https://doi.org/10.3390/app14114558

Academic Editor: Luis Javier Garcia Villalba

Received: 20 April 2024
Revised: 23 May 2024
Accepted: 24 May 2024
Published: 25 May 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In the past few decades, deep learning has made significant advancements across various research fields. In computer vision, deep learning-based methods have gained widespread adoption in scientific research and industrial applications, particularly in the domain of object detection. The performance of object detection methods heavily relies on the availability of large and diverse datasets. Several prominent examples of such datasets include ImageNet [1], which is utilized by the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), and VOC2007 and VOC2012 [2], which were released by the PASCAL Visual Object Classes (PASCAL VOC). Additionally, there are other notable datasets like Microsoft Common Objects in Context (COCO) [3] and Google Open Images [4]. These datasets encompass a vast number of images, object instances, and categories, thereby enhancing the effectiveness of training, validation, and testing processes for object detection models.

In recent years, remote sensing images have found widespread applications in various fields, such as geographic information systems, environmental monitoring, and urban planning. The detection and analysis of remote sensing images rely heavily on the availability of large datasets. However, acquiring real remote sensing images poses numerous challenges due to signal interference, geographical conditions, lighting variations, dust, and other factors. The annotation process for these images is complex, time-consuming, and expensive. These factors impede the progress and practicality of various remote sensing

image processing tasks. To address these challenges, researchers have utilized several well-established remote sensing image datasets, including DOTA [5], UCAS-AOD [6], NWPU VHR-10 [7], DIOR [8], and TGRS-HRRSD-Dataset [9]. However, these datasets are limited in scale and struggle to meet the growing demands of remote sensing research. To overcome these limitations, the creation of synthetic remote sensing image datasets has emerged as an effective solution. In this study, we focused on investigating object detection using commonly found ships in aerial remote sensing image datasets. The synthesized image datasets were specifically generated with ships as the exclusive object.

Synthetic remote sensing image datasets comprise a series of images created by simulating the visual characteristics and geometric features of real-world scenes. These synthetic images can be generated using real remote sensing images as a basis or can be entirely controlled by the textual descriptions of a scene's main content. By incorporating variations in factors such as lighting conditions, weather conditions, and seasonal changes, a diverse range of remote sensing images can be simulated.

Some research studies have demonstrated the feasibility of synthetic image datasets. For example, Meta's realistic synthetic dataset was developed using Unreal Engine [10], Cam2BEV [11], and ProcSy [12] for synthesizing road scene images in the field of autonomous driving. Furthermore, game engine-based synthetic datasets have been established for tasks such as object recognition and video semantic segmentation.

Establishing synthetic remote sensing image datasets offers several significant advantages. Firstly, the generation process of synthetic images is controllable, allowing for the flexible adjustment of parameters and variation factors to produce synthetic images tailored to specific scenarios. This capability enables researchers to conduct large-scale experiments and tests under different conditions, thereby enhancing the robustness and reliability of algorithms. Secondly, synthetic remote sensing image datasets can provide rich label information, which is crucial for deep learning tasks. By introducing labels such as land cover categories, bounding boxes, and semantic segmentation during the generation process, accurate references can be provided for algorithm training and evaluation.

However, relying exclusively on synthetic data for training and deploying the model in real-world object detection scenarios leads to a noticeable degradation in performance. This is primarily attributed to substantial disparities in terms of texture, viewpoint, imaging quality, and color between synthetic and real images. Such dissimilarities between the synthetic domain and the real domain are commonly known as domain shift. In the context of cross-domain object detection [13–19] tasks, the synthetic domain is commonly referred to as the source domain, whereas the real domain is referred to as the target domain. To mitigate the domain shift between the source and target domains, we propose an algorithmic framework for domain adaptation that aims to adapt to various source and target domains.

The structure of this paper is as follows:

- The Introduction section describes the problem of aerial object detection and our solution using synthetic images, and it also discusses the problem of the interaction of the synthetic domain with the real domain, which needs to be solved by domain-adaptive algorithms to apply synthetic image datasets for object detection.
- The Related Works section describes existing real-world image datasets, synthetic image datasets, and domain-adaptation algorithms for object detection.
- The Methods section describes how we built the synthetic image datasets and knowledge about the domain-adaptive algorithms.
- The Experiments section describes the synthetic-to-real domain, adaptive object detection experiments, including the implements, real datasets, detailed data, and experimental conclusions.
- The Conclusion section describes the contributions of the full paper and the shortcomings of this study, and it also points out directions for future improvements.

2. Related Works

2.1. Real-World Image Datasets

With the advancement of computer vision, object detection techniques have become increasingly comprehensive and sophisticated. It is widely recognized that the size of the dataset has a significant impact on the performance of object detection. A larger dataset allows for a higher upper limit of object detection accuracy, making large datasets critical for object detection tasks. Several corporations and challenges have released their own datasets to facilitate research and development in this field. For instance, Stanford's Fei-Fei Li team created ImageNet, which consists of over 14 million images spanning 20,000 categories. The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) utilizes this dataset. PASCAL Visual Object Classes has also released image datasets with labeled data and conducted competitions over multiple years, including notable examples such as VOC 2007 and VOC 2012. Microsoft has contributed the Microsoft Common Objects in Context (COCO) dataset, which includes 91 categories and a vast number of images for each category. Additionally, Google Inc. released Google Open Images, which comprises 1.9 million images, 600 categories, and 15.4 million annotated instances.

In the domain of aerial imagery, the availability of rich datasets is also crucial to improve the practical performance of object detection. However, due to the inherent characteristics of aerial imagery, commonly used datasets in remote sensing are often small in size. This can be attributed to factors such as the high cost of satellite imagery, the susceptibility of aerial photographic tools to signal interference, interference from urban structures, expensive cameras that are suitable for aerial imaging, low resolution, and other challenges. Some well-known remote sensing datasets include DOTA [5], UCAS-AOD [6], NWPU VHR-10 [7], TGRS-HRRSD-Dataset [9], DIOR [8], VEDAI [20], and LEVIR [21]. A comparison of various prominent remote sensing datasets is presented in Table 1.

Table 1. A comparison between general and aerial datasets in terms of the number of images, categories, and instances.

Type	Name	Images	Categories	Instances	Year
General Dataset	ImageNet [1]	14,197,122	21,841	/	2009
	VOC [2]	33,043	20	79,540	2007+2012
	COCO [3]	330 K	80	1.5 M	2014
	Open Images V7 [4]	1.9 M	600	15.4 M	2022
Aerial Dataset	NWPU VHR-10 [7]	800	10	/	2014
	UCAS-AOD [6]	910	2	6029	2014
	VEDAI [20]	1210	9	3640	2015
	HRSC2016 [22]	1070	1	2976	2016
	COWC [23]	53	4	32,716	2016
	SSDD [24]	1160	1	2358	2017
	RSOD-Dataset [25]	976	4	6950	2017
	airbus-ship-detection [26]	42,556	1	81,011	2018
	DIOR [8]	23,463	20	192,472	2018
	xView [27]	1413	60	800,636	2018
	DOTA [5]	2806	15	188,282	2018
	ITCVD [28]	173	1	29,088	2018
	LEVIR [21]	21,952	3	10,467	2018
	TGRS-HRRSD-Dataset [9]	21,761	13	55,740	2019
	AIR-SARShip-1.0 [29]	31	1	461	2019
	HRSID [30]	5604	1	16,951	2020

2.2. Synthetic Image Datasets

Due to the time-consuming, labor-intensive, and costly nature of acquiring real images to construct a dataset, the synthesis of images has gained popularity as an alternative approach. Synthetic images can be generated through two main pathways: physics engine-based synthesis methods and computer vision algorithm-based synthesis methods. Physics

engine-based methods utilize platforms like Unity3D or the Unreal Engine. Computer vision algorithm-based methods encompass rule-based approaches, generative adversarial network (GAN)-based [31] methods, variant autoencoder (VAE)-based [32] methods, and diffusion model [33,34] methods.

Each technique has its own advantages and disadvantages. Rule-based generative models offer faster sample generation but come with higher computational complexity and numerous constraints and limitations. Generative adversarial network-based methods generate images with high fidelity but may suffer from issues such as a lack of diversity. Variational autoencoder-based generated images tend to be smoother but offer less controllability. Diffusion model-based images are of higher quality but require more iterations; furthermore, they are computationally expensive, and adjusting their parameters can be challenging.

Several synthetic image datasets have been widely and successfully utilized in various fields. Meta has developed a photorealistic synthetic dataset called Photorealistic Unreal Graphic (PUG) using Unreal Engine 5, which offers improved control over AI vision systems and enables robust evaluation and training [10]. Stable Diffusion has been used to generate facial datasets, including the first and largest artificial emotion dataset [35,36]. In the field of autonomous driving, synthetic datasets like Cam2BEV and ProcSy provide dynamic road scene images and aid in pedestrian recognition [11,12]. Virtual KITTI is another synthetic dataset used for comparison with real datasets in autonomous driving research [37]. For object recognition and segmentation, synthetic datasets built on game scenarios and physics engines include PersonX, SAIL-VOS, and GCC [38–40]. DiffusionDB is a dataset containing 2 million images with textual descriptions of the generated images [41].

There are even game engines that specialize in generating synthetic images. G2D captures video from GTA V [42]. CARLA supports the entire process of an autopilot system [43]. MINOS simulates various house scenes [44].

Therefore, it can be concluded that synthetic aerial image datasets have the potential to be applied in object detection tasks on aerial images. In fact, there have been several attempts in this direction. Cabezas et al. introduced a multi-modal synthetic dataset for this purpose [45]. Gao et al. created a large-scale synthetic urban dataset specifically for unmanned aerial vehicles research and demonstrated performance improvements in comparison with real aerial images [46]. Kiefer et al. applied a large-scale, high-resolution synthetic dataset to object detection tasks for unmanned aerial vehicles [47]. Additionally, there are synthetic aerial image datasets available, such as RarePlanes and Sim2Air [48,49].

2.3. Object Detection from the Synthetic World to the Real World

Taking the number of stages as a dimension, object detection can be categorized into one-stage algorithms and two-stage algorithms. One-stage algorithms have the advantages of fast speed and good instantaneous performance, which are important in scenarios that require real-time detection. However, they tend to have lower accuracy and may produce less satisfactory results for small object detection. Examples of one-stage algorithms include the You Only Look Once (YOLO) series [50–54], RetinaNet [55], and Fully Convolutional One-Stage Object Detection (FCOS) [56].

On the other hand, two-stage algorithms offer high accuracy and perform well in detecting small objects. However, they tend to be slower due to the need to generate a large number of candidate regions. Examples of two-stage algorithms include Faster R-CNN [57], Mask R-CNN [58], and TridentNet [59].

Using synthetic image datasets directly for object detection can improve results to some extent. However, it can also introduce issues such as dataset shift or domain shift due to the inherent differences between synthetic and real image datasets. To address this problem, domain-adaptation methods can be employed to make better use of existing synthetic datasets.

Early cross-domain object detection methods have primarily focused on Faster R-CNN, which is the most representative algorithm among two-stage detectors. One notable

example is the DA-Faster algorithm proposed by Chen et al. [13], who introduced a gradient-reversal layer [60] and were the first to incorporate image-level and instance-level alignment techniques to enhance cross-domain performance. The Stacked Complementary Losses (SCL) method developed by Shen et al. [61] utilized a gradient-detachment approach to tackle the domain shift issues in object detection. More recently, there have been breakthroughs in the application of single-stage detectors to domain adaptation for object detection (DAOD) problems, as demonstrated by works such as [13–19]. The EP-MDA method proposed by Hsu et al. [62] made adjustments to FCOS [56], and they drew inspiration from semi-supervised learning. Additionally, the Mean Teacher with Object Relations (MTOR) [63] and Unbiased Mean Teacher (UMT) [15] methods built upon the mean teacher framework for unsupervised domain adaptation.

3. Methods

3.1. Engine-Based Generation

When it comes to physics engine generation, Unity provides a simpler and more user-friendly interface and workflow. This can help reduce the learning curve and development costs associated with using the physics engine. Unity is a cross-platform game development engine that supports the creation of both 2D and 3D content. It offers a powerful visual editor, allowing developers to freely build scene terrains, models, and add objects as needed. Developers can use the C# programming language to control components and scripts, and there are abundant components and application programming interfaces (APIs) that can be integrated with Visual Studio for seamless communication. Unity also boasts powerful graphic rendering capabilities, enabling the real-time rendering of scenes. Moreover, Unity's camera component allows for the continuous capture of scene photos and enables output in layers based on their contents, greatly facilitating image processing tasks.

Virtual synthesis techniques involve constructing scenarios and objects using a 3D virtual engine based on real-world applications, resulting in a large number of images. However, to create a synthetic dataset that is both realistic and generalizable, several factors need to be considered, such as variations in the spatial resolution of each sensor, sensor viewing angles, collection of the time of day, object shadows, and illumination based on the position of the sun relative to the sensors. The platform takes in real-world metadata and images to programmatically generate 3D environments of real-world locations, incorporating various parameters such as weather conditions, collection time, sunlight intensity, viewing angles, modeling, and object distribution density. All these parameters can be adjusted using the Unity engine, allowing for the creation of diverse and heterogeneous datasets.

We utilized these tools to generate a virtual synthetic dataset using the Unity engine. A total of 77 ship models, varying in size and type, from expedition yacht to aircraft carrier, were employed. To ensure diversity, we created seven virtual scenes, each encompassing three weather conditions, four time periods, and corresponding light intensities. Additionally, we captured photographs from different heights, resulting in a rich and varied dataset for synthetic image synthesis.

For accurate annotation, we labeled the horizontal bounding box for each ship object in the standard annotation format used for object detection. Furthermore, we included the rotation box and other relevant information to establish a foundation for future research objectives. To control the ships and cameras, we utilized the API with two methods at specific spatial intervals: fixed shooting and scanning shooting.

The virtual camera in Unity can be controlled using the camera component. For our simulation of an aerial image, we set the camera mode to 'top-down'. The horizontal field of view was fixed at 60 degrees, and the image resolution was set to 2000×1600. To generate the ship objects, we randomly selected a number of ships from the ship model library and randomized their coordinates and rotation angles. To prevent object overlap, we added a collision body component to all objects. This component allows us to set collision triggers and fill the land portion with transparent collision bodies, ensuring that the ships appear on

the water surface. To generate weather conditions, we randomly generated direct sunlight within a specific spatial area, including the position of the sun, the angle of direct sunlight, and the light intensity. Additionally, other weather parameters were determined using random numbers.

The synthetic dataset we generated is named "SASI-UNITY". Sample images from the dataset are displayed in Figure 1. It consists of a total of 10,000 images, which were divided into training, validation, and test sets in a 6:2:2 ratio. The real datasets used for comparison were also divided into training, validation, and test sets following the same ratio, unless stated otherwise. We conducted an analysis of the dataset, including counting the number of instances for each ship class. The results are presented in Figure 2a. The distribution of ship object locations is illustrated in Figure 3a, and the relative sizes of the labeled bounding boxes are shown in Figure 4a.

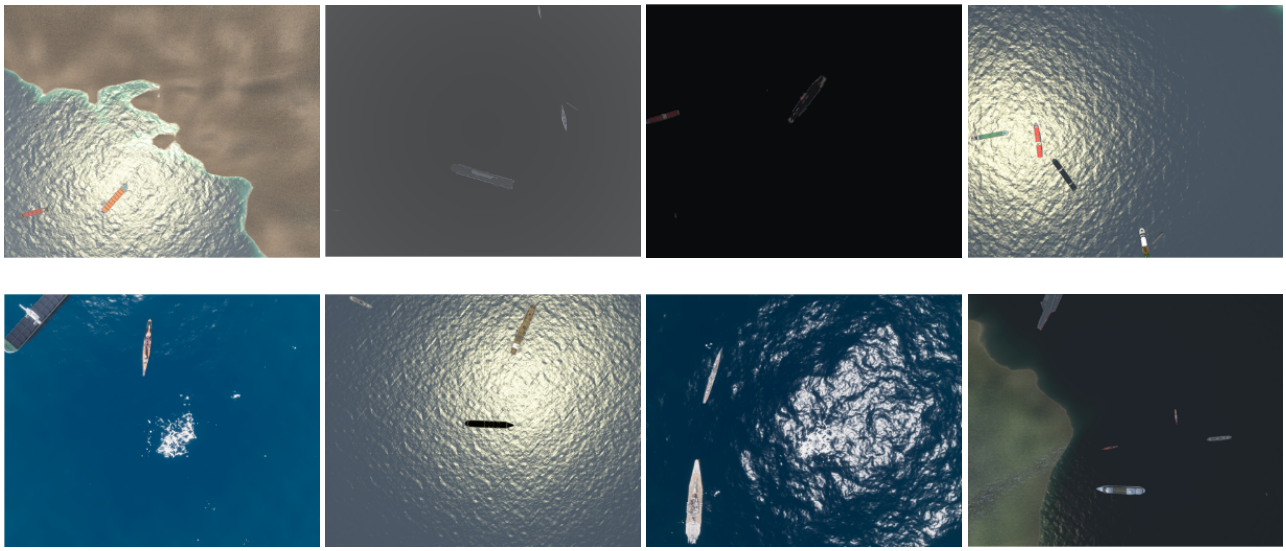
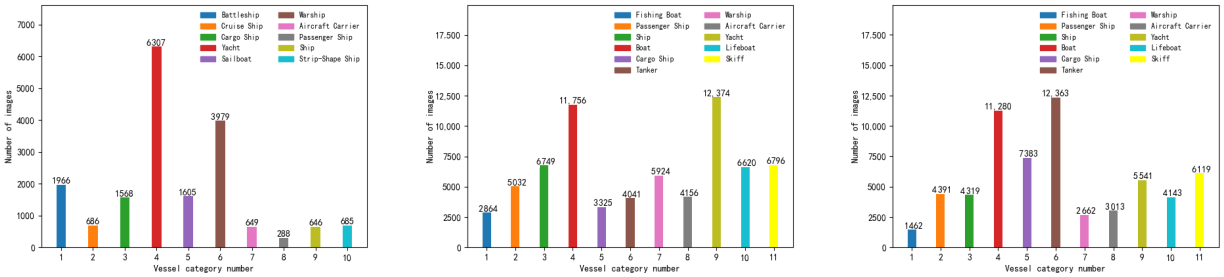
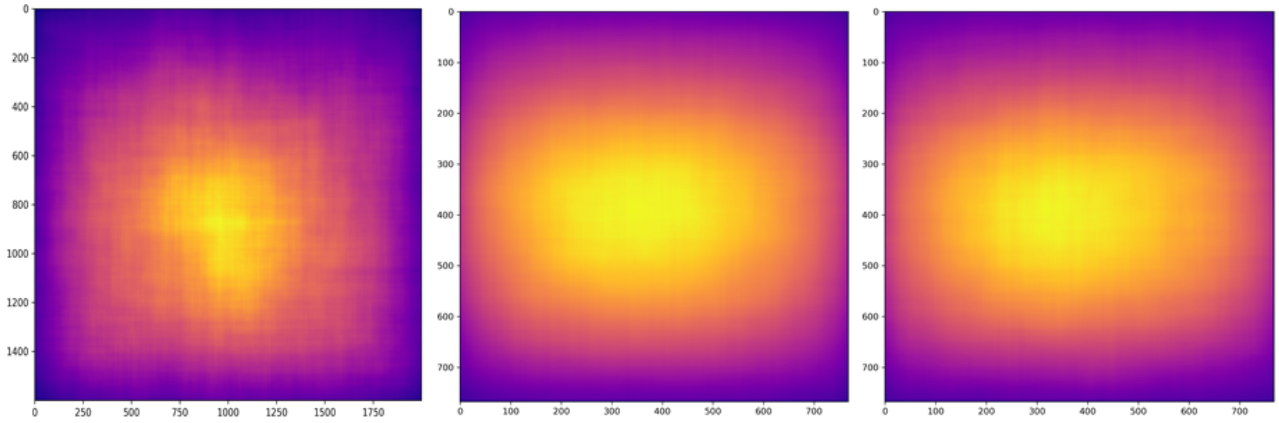


Figure 1. Sample images from the SASI-UNITY dataset.



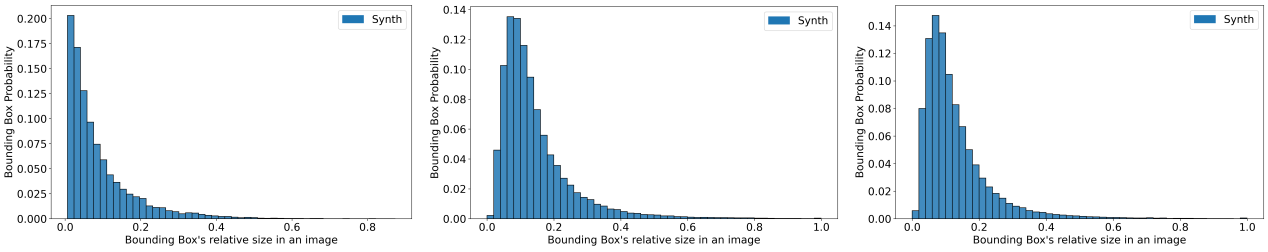
(**a**) (**b**) (**c**)

Figure 2. The number of ship object instances per category: (**a**) SASI-UNITY dataset, (**b**) SASI-ZERO dataset, and (**c**) SASI-RANDOM dataset.



(**a**) (**b**) (**c**)

Figure 3. Location distribution of the instances: (**a**) SASI-UNITY dataset, (**b**) SASI-ZERO dataset, and (**c**) SASI-RANDOM dataset.



(**a**) (**b**) (**c**)

Figure 4. The bounding box relative images sizes: (**a**) SASI-UNITY dataset, (**b**) SASI-ZERO dataset, and (**c**) SASI-RANDOM dataset.

3.2. Image Generation with Models

The DALL-E series [64,65] from OpenAI, specifically the generative model called DALL-E by Midjourney Labs, and Stability AI's Stable Diffusion [66] are all highly regarded image generation models that are widely used in the field. For this paper, we selected Stable Diffusion as the generation tool due to its open-source nature.

Stable Diffusion [66] is capable of generating images based on text and modifying existing images using input text. It consists of two main components: a text understander and an image generator. The text understander converts textual information into a numerical representation, allowing the model to understand the meaning of the text. It is implemented as a specially designed transformer language model, which is essentially a CLIP [67] model. When text is input into the text understander, each word or token is associated with a vector feature, resulting in a feature list. This feature list serves as input information for the image generator.

The image generator can be further divided into two parts: the image information creator and the image decoder. The image information creator operates in the latent space, which is crucial for Stable Diffusion. It consists of a UNet [68] neural network and a scheduling algorithm. The UNet network is responsible for transforming the arrays of latents. In Stable Diffusion, attention layers are added between the ResNet blocks of the UNet to process text tokens. The feature list is progressively denoised to transform it into an array of information, which is then passed to the image decoder. The image decoder generates an image based on the information provided by the image information creator. Figure 5 provides a simplified illustration of the image generation process using Stable Diffusion.

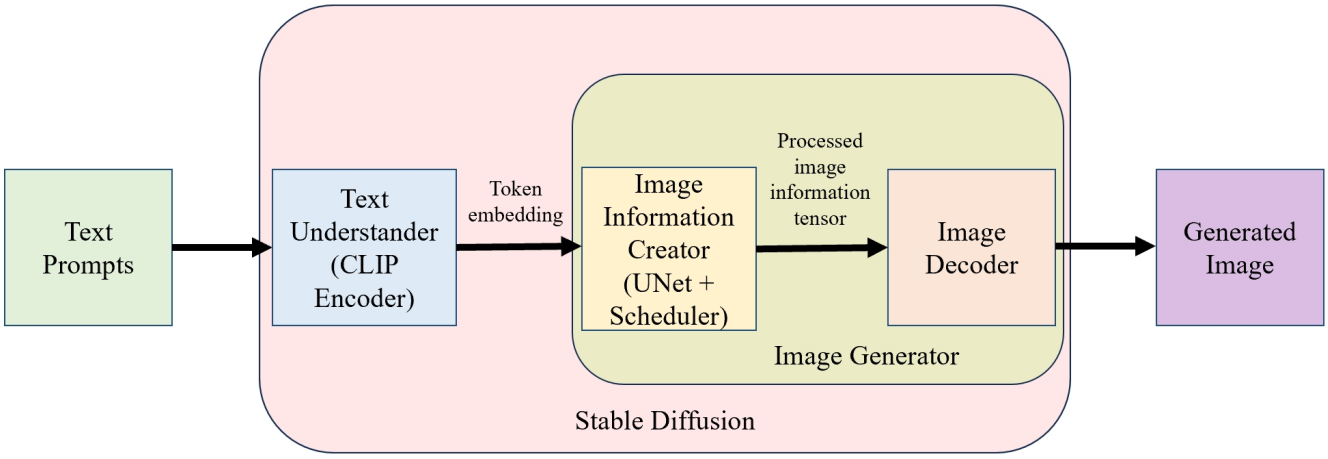


Figure 5. The overall workflow of Stable Diffusion [66].

There are two popular open-source versions of Stable Diffusion, namely version 1.5 and version 2.1. In practical application scenarios, version 2.1 is more industrialized and productive, while version 1.5 excels in terms of overall coherence and contextualization. Since they are trained on different bases, version 2.1 provides checkpoint files that enable the generation of 768-resolution images and improve performance on larger formats, which aligns better with the characteristics of remote sensing images. The demand for GPU memory is substantial when generating synthetic images with a resolution of 768×768. We set the model's batch size to 1, allowing the generation of one image per batch. The generation frequency needs to be adjusted to complete the generation of a large number of images. Stable Diffusion employs Classifier-Free Guidance (CFG) [69] to train the conditional diffusion model, and noise is introduced during the sampling stage. The formula used is as follows:

$$\begin{aligned} n &= \omega n_c + (1-\omega) n_u \\ &= \omega \varepsilon_\theta(x_t, t, c) + (1-\omega)\varepsilon_\theta(x_t, t), \end{aligned} \tag{1}$$

where n is noise, n_c is the conditional prediction of noise, n_u is the unconditional prediction of noise, ε is the Gauss noise, x_t is the sample obtained from the real data sampled by time, t is the time step, c is the token embeddings, and ω is the guidance scale. As ω becomes bigger, the condition plays a bigger role, i.e., the generated image is more consistent with the input text. However, when the CFG [69] value is large enough, the image will distorted, so we consistently used a value of 9.0.

According to the official example of Stable Diffusion, each run generates different depictions of the same input prompt statements. To enhance the diversity of the generated images and capture the complexity and variability of actual remote sensing images, we introduced a randomized prompt statement module. This module enables the generation of a new prompt statement for each iteration, ensuring a wider range of variations and characteristics in the generated images.

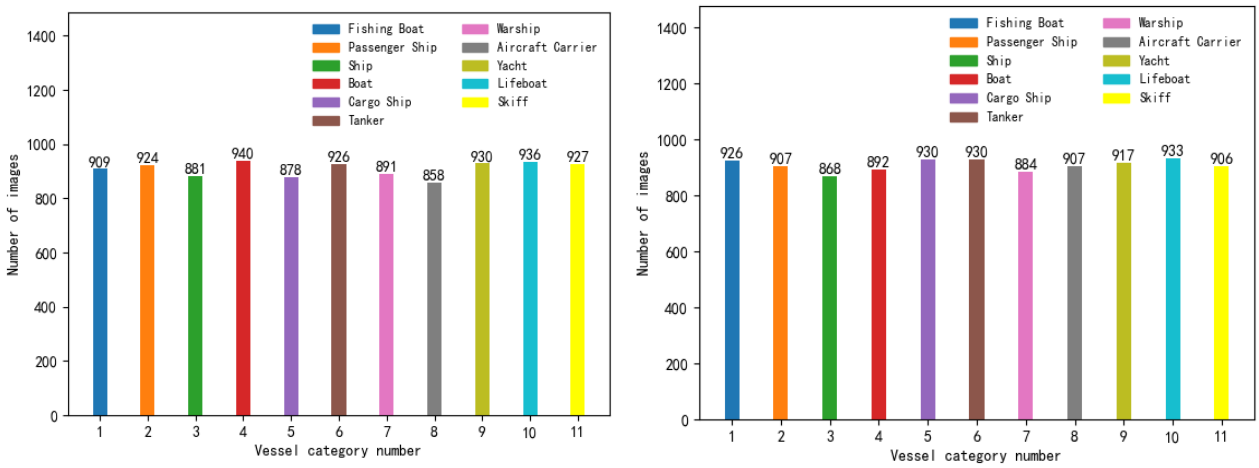
The module for random prompts is divided into two parts: the mandatory prompts section and the optional prompts section. The mandatory prompts section contains four groups of descriptors, each representing information about the synthetic images. These groups include the image type, the number of vessel objects, the vessel object category, and the environment description. The image type group controls the shooting pitch angle of the synthetic remote sensing image. The number of vessel object group approximates the number of objects in an image. The vessel object category group randomly selects from 11 vessel categories. The environment description group simulates the complex geographic location of a vessel detection in reality, including the sea, rivers, harbors, and shore. The

section on optional prompts includes eight sets of descriptors: vessel color, presence of fog, presence of wake, presence of waves, vessel alignment, sunlight intensity, weather, and water color.

Accordingly, we created two small datasets, each comprising 10,000 images: SASI-ZERO and SASI-RANDOM. The SASI-ZERO dataset was generated solely using mandatory descriptors as prompts. On the other hand, the SASI-RANDOM dataset was generated using prompts that included both mandatory descriptors and optional descriptors, with one-to-three optional descriptors randomly selected for each image.

Before conducting the detection experiments, we needed to preprocess the synthesized image dataset. This was necessary because the output images from Stable Diffusion did not contain any labeling information about the main objects present. Labeling information is crucial for the detection experiments. Thus, we annotated the synthetic dataset using common semi-supervised learning methods. We began by labeling a small sample of images and training a classification model using the labeled data. The prediction of the classification model is the result of the target labeling. As the automatic labeling of semi-supervised learning methods cannot guarantee 100% accuracy, we manually verified and corrected the labeled information of all the images.

Subsequently, we conducted a comprehensive analysis of the dataset. We counted the number of instances present in each ship class, as depicted in Figure 2b,c. We also examined the distribution of ship object locations, as illustrated in Figure 3b,c. Additionally, we determined the relative size of the labeled bounding boxes, as demonstrated in Figure 4b,c. Since the individual images in both SASI-ZERO and SASI-RANDOM contain the same type of ship, we further counted the number of images, which included each ship category, as depicted in Figure 6. Sample images from the datasets are displayed in Figure 7.



(**a**) (**b**)

Figure 6. The number of images per category: (**a**) SASI-ZERO dataset and (**b**) SASI-RANDOM dataset.



Figure 7. Sample images from the SASI-ZERO and SASI-RANDOM datasets.

3.3. Syn2Real Object Detection Framework

To facilitate operational object detection, we chose to apply a domain-adaptive object detection algorithm framework that was developed based on MMDetection [70].

In domain-adaptive object detection research, solving the domain transfer problem from the source domain to the target domain is crucial in order to address the variability between synthetic and real data. Below, we present some conceptual definitions to elucidate the principle of DAOD.

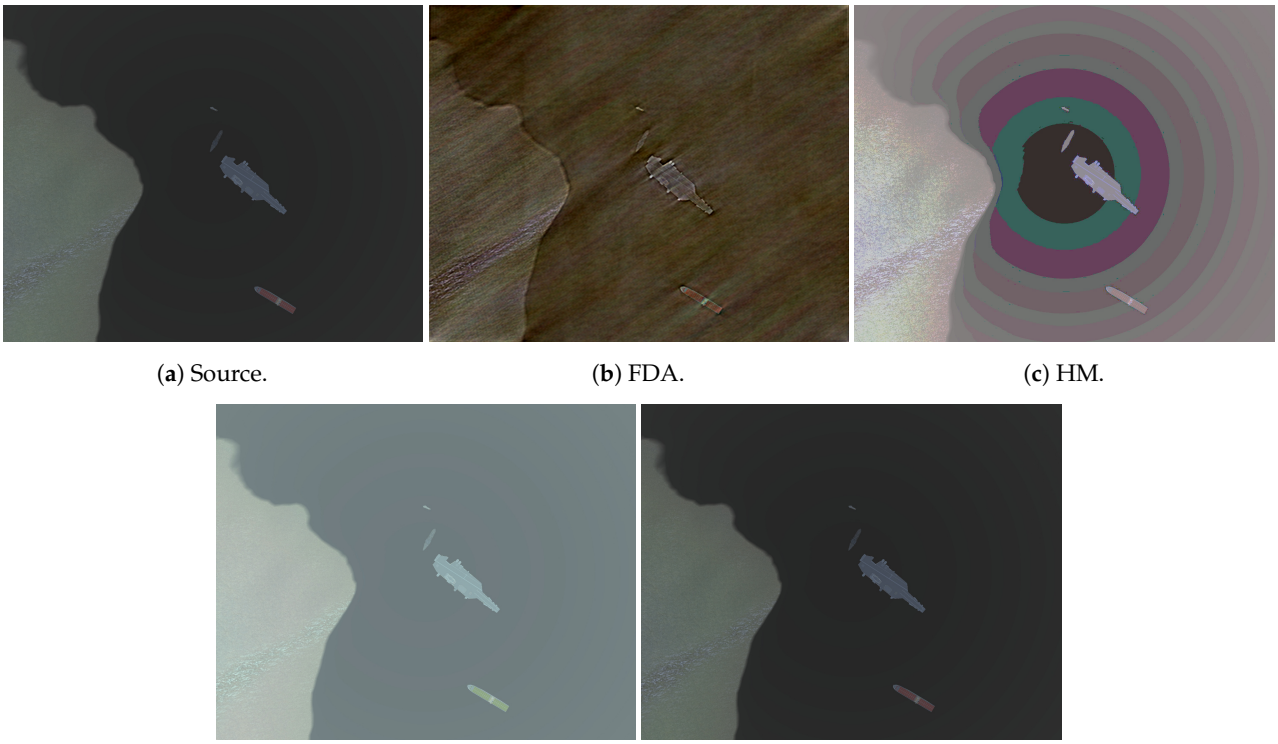
A set of source images is called $\boldsymbol{I}^s$. $\boldsymbol{I}^s$ has a total of N object bounding boxes, with the set of boxes called $\mathcal{B}^s$, $\mathcal{B} = \{B_j|_{j=1}^N, B_j = (x_j, y_j, w_j, h_j)\}$, and c object classes, with the corresponding class labels $\mathcal{C}^s$, $\mathcal{C} = \{C_j|_{j=1}^N, C_j \in (0, 1, 2, \ldots, c)\}$. Thus, we determined the representation of the source domain as $\mathcal{D}_s$, $\mathcal{D}_s = \{(\boldsymbol{I}_i^s, \mathcal{B}_i^s, \mathcal{C}_i^s)|_{i=1}^N\}$. Since the target domain is not required to be labeled, we disregarded the existence of $\mathcal{B}^t$ and $\mathcal{C}^t$ here. Corresponding to the representation of the source domain, the target images can be written as $\boldsymbol{I}^t$, and the target domain can be written as $\mathcal{D}^t$, $\mathcal{D}^t = \{\boldsymbol{I}^t|_{i=1}^N\}$.

The mean teacher (MT) model [71] aims to enhance the robustness of target domain prediction. However, it is often biased toward the source domain. To address this issue, we employed three domain-adaptation augmentation methods to facilitate image style transfer, thereby making the model more favorable toward the target domain data. These three methods, as illustrated in Figure 8, include Fourier domain adaptation (FDA) [72], histogram matching (HM) [73], and pixel distribution adaptation (PDA). The loss formula is defined as follows:

$$\mathcal{L}_{det}(\boldsymbol{I}^s, \mathcal{B}^s, \mathcal{C}^s) = \mathcal{L}_{box}(\mathcal{B}^s; \boldsymbol{I}^s) + \mathcal{L}_{cls}(\mathcal{C}^s; \boldsymbol{I}^s), \tag{2}$$

$$\mathcal{L}_{det}(\boldsymbol{P}^s, \mathcal{B}^s, \mathcal{C}^s) = \mathcal{L}_{box}(\mathcal{B}^s; \boldsymbol{P}^s) + \mathcal{L}_{cls}(\mathcal{C}^s; \boldsymbol{P}^s), \tag{3}$$

where $\mathcal{L}_{box}$ is the loss of predicted bounding boxes and $\mathcal{L}_{cls}$ is the loss of the classification probability. Formula (2) is about source images. Formula (3) is about target-like images, and it only uses $\boldsymbol{P}^s$ instead of $\boldsymbol{I}^s$.



(**a**) Source. (**b**) FDA. (**c**) HM.

(**d**) PDA. (**e**) Mixture of all methods.

Figure 8. Sample translated images: (**a**) a source image from SASI-UNITY, (**b**) a target-like image translated using the FDA method, (**c**) a target-like image translated using the HM method, (**d**) a target-like image translated using the PDA method, and (**e**) a target-like image translated using all three methods.

The MT model [71] consists of a student model and a teacher model, both of which have identical architectures, as depicted in Figure 9, where the model is initially trained using the source dataset $\mathcal{D}^s$, and the resulting $\mathcal{L}_{sup}$ is equivalent to $\mathcal{L}_{det}$. The teacher model generates pseudo-labels for the target domain. By setting an appropriate confidence threshold, false positives caused by noise can be eliminated, and $\mathcal{L}_{unsup}$ is utilized to update the weights of the student model. The weights of the teacher model are then updated using the exponential moving average (EMA) formula, a commonly used smoothing technical indicator for smoothing time series data. The formulas are as follows:

$$\mathcal{L}_{unsup}(\boldsymbol{I}^s, \mathcal{B}^t, C^t) = \mathcal{L}_{box}(\mathcal{B}^t; \boldsymbol{I}^s) + \mathcal{L}_{cls}(C^t; \boldsymbol{I}^s), \tag{4}$$

$$\theta_t \leftarrow \alpha\theta_t + (1-\alpha)\theta_s, \tag{5}$$

where θ_t are the weight parameters of teacher model, θ_s are the weight parameters of the student model, and $\mathcal{B}^t$ and C^t are the pseudo-labels generated by the teacher model.

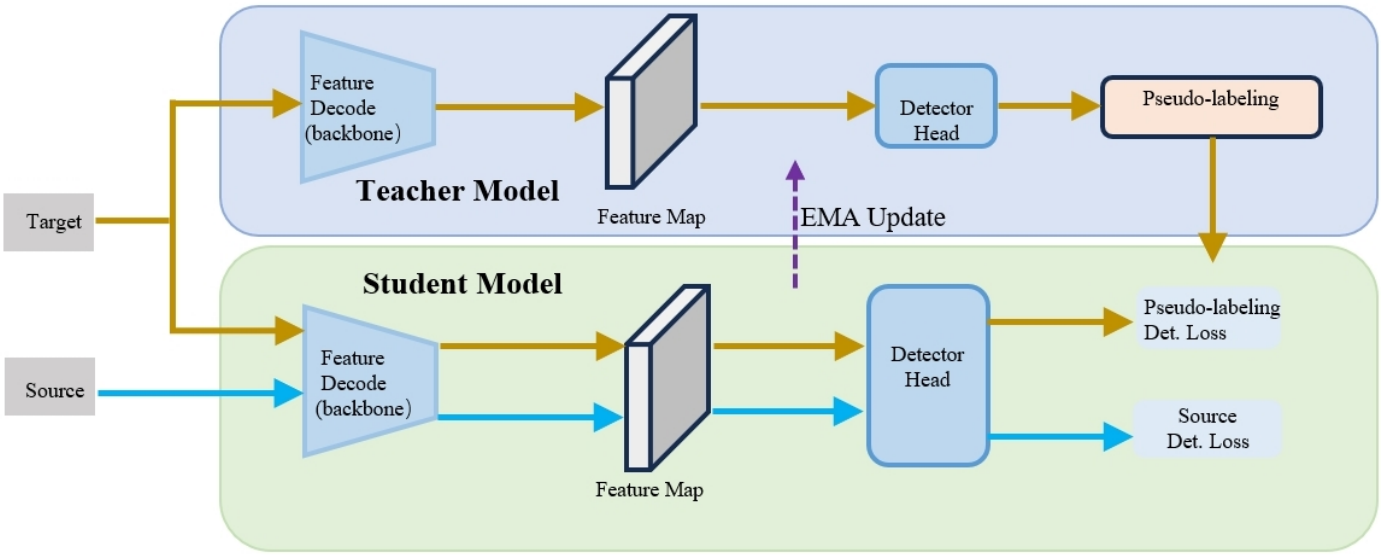


Figure 9. The architecture of the mean teacher model. The loss of pseudo-labels generated by the teacher model was used to update the student model, which updates the teacher model with the EMA formula.

4. Experiments

4.1. Target Datasets

The TGRS-HRRSD-Dataset [9] is specifically designed for object detection in high-resolution remote sensing images. It comprises over 20,000 images with spatial resolutions ranging from 0.15 to 1.2 m. The dataset contains a total of more than 50,000 object instances, spanning across 13 categories of remote sensing imaging (RSI) objects. Each category consists of a relatively balanced number of instances. For our study, we extracted the subset that focuses on ships, resulting in 2165 images and 3954 ship objects. We randomly partitioned the data, allocating 60% (1299 images) for the training set, 20% (433 images) for the validation set, and another 20% (433 images) for testing.

The LEVIR [21] dataset contains a large collection of high-resolution Google Earth images, each with dimensions of 800 × 600 pixels. The dataset comprises over 22,000 images and covers various terrestrial features of human habitats, including cities, villages, mountains, and oceans. However, extreme terrestrial environments such as glaciers, deserts, and the Gobi Desert are not included. The dataset consists of three object types: airplanes, ships (both offshore and seaward), and oil tanks. A total of 11,000 individual bounding boxes were annotated for all the images, resulting in an average of 0.5 objects per image. For our analysis, we extracted the subset that focuses on ships, resulting in 1494 images and 2961 ship objects. Similar to the previous dataset, we randomly divided the dataset into three sets: 60% (896 images) for training, 20% (299 images) for validation, and 20% (299 images) for testing.

The airbus-ship-detection [26] dataset was released in 2019 for the "Airbus Ship Detection Challenge" on Kaggle. It is one of the largest publicly available datasets for ship detection in aerial images. The original dataset provides instance segmentation labels in mask form. However, for our experiment, these labels were transformed into horizontal bounding boxes to simplify the representation of ship locations. The original training dataset consists of 192,556 images, out of which 150,000 are negative samples (no ships present). We selected only the positive samples, resulting in a subset of 42,556 images that contain a total of 81,011 ships. The dataset was randomly divided into a training set (60%—25,533 images), a validation set (20%—8511 images), and a test set (20%—8511 images).

4.2. Implements

The experiments we conducted are outlined below, and the results are presented in tabular form. All the experiments were performed on a computer equipped with an Intel Core i7 CPU and an NVIDIA GeForce RTX 3090 GPU for acceleration. To simplify the presentation, we abbreviated our datasets as UNITY, ZERO, and RANDOM. As for the models, we utilized a one-stage FCOS model and a two-stage Faster R-CNN model, both utilizing ResNet-101 as the backbone. For method selection, six distinct methods were employed for each component: mean teacher; training on the source domain only; FDA; HM; PDA; and the simultaneous combination of FDA, HM, and PDA. We maintained consistent parameter settings across all experiments, including trade-off parameters $\lambda_{unsup} = 1.0$ and $\lambda_{img} = 0.1$, confidence threshold $\sigma = 0.8$, and the weight-smoothing coefficient parameter of the exponential moving average $\alpha = 0.99$.

4.3. DAOD Experiments

The synthetic datasets generated in Sections 3.1 and 3.2 serve as the source data, while the real-world datasets serve as the target data. Initially, we performed separate object detection on the synthetic dataset to obtain the results of the "Source" method. Subsequently, the results were processed using three style transfer methods, namely "FDA", "HM", and "PDA", to obtain domain-adaptive results for a single-style transfer. For mixed-style transfer, the domain-adaptive results of the "FDA+HM+PDA" method were obtained. Finally, employing the mean teacher method, we obtained domain-adaptive results, which are marked as "MT".

All experimental results were evaluated according to the COCO standard evaluation metrics. The IoU threshold is the measure of overlap between the selected bounding box and the predicted bounding box of the object. The mAP value denotes the average precision across 10 thresholds, ranging from IoU 0.50 to 0.95 in increments of 0.05. The mAP_{50} value represents the average precision at an IoU threshold of 0.50. The mAP_{75} value represents the average precision at an IoU threshold of 0.75. The mAP_s indicates the precision of detecting small objects, mAP_m denotes the precision of detecting medium objects, and mAP_l signifies the precision of detecting large objects. In the experiments of this paper, the mAP values were the focus of attention.

The experimental results of the cross-domain object detection, obtained from applying our datasets to the TGRS-HRRSD-Dataset [9], are presented in Tables 2–4. The experimental results of the cross-domain object detection, obtained from using our datasets on the LEVIR dataset [21], are shown in Tables 5–7. Additionally, the experimental results of the cross-domain object detection, obtained by employing our datasets on the airbus-ship-detection dataset [26], are displayed in Tables 8–10.

Table 2. The Mean Average Precision (mAP, in %) of different models and methods for the cross-domain object detection on the validation set of our dataset, i.e., for SASI-UNITY to TGRS-HRRSD-Dataset [9] adaptation. The best mAPs are highlighted in bold.

Model	Method	mAP	mAP_{50}	mAP_{75}	mAP_s	mAP_m	mAP_l
Faster R-CNN	Source	25.6	49.8	24.2	0.6	24.6	27.0
	FDA	22.3	45.7	19.0	0.3	18.8	26.4
	HM	22.5	47.1	18.7	0.2	20.9	24.4
	PDA	27.8	55.4	25.2	2.2	24.6	31.6
	FDA+HM+PDA	26.2	53.9	22.4	0.9	24.3	28.8
	MT	**43.1**	**77.9**	**42.0**	**23.0**	**47.8**	**40.7**
FCOS	Source	22.9	47.3	20.8	3.2	23.7	23.7
	FDA	23.8	50.5	19.7	3.4	21.9	26.9
	HM	22.5	50.0	15.8	1.1	18.7	27.9
	PDA	27.6	56.7	24.3	**4.2**	25.8	30.8
	FDA+HM+PDA	23.9	52.6	18.2	2.0	21.7	27.6
	MT	**42.6**	**79.3**	**41.4**	0.7	**34.5**	**52.9**

Table 3. The Mean Average Precision (mAP, in %) of different models and methods for the cross-domain object detection on the validation set of our dataset, i.e., SASI-ZERO to TGRS-HRRSD-Dataset [9] adaptation. The best mAPs are highlighted in bold.

Model	Method	mAP	mAP_{50}	mAP_{75}	mAP_s	mAP_m	mAP_l
Faster R-CNN	Source	33.1	57.7	34.6	2.1	18.9	45.4
	FDA	33.1	58.8	34.0	2.1	21.1	43.8
	HM	27.0	52.5	25.5	0.3	16.5	37.4
	PDA	36.0	62.6	37.4	0.9	23.0	47.7
	FDA+HM+PDA	28.0	51.7	26.7	0.2	15.2	39.4
	MT	**44.2**	**80.7**	**43.0**	**4.5**	**40.5**	**50.3**
FCOS	Source	34.6	62.0	34.3	0.8	22.1	45.6
	FDA	31.7	59.1	30.9	0.9	19.2	42.7
	HM	29.7	59.1	26.6	0.1	18.2	41.1
	PDA	34.9	64.2	33.5	0.2	21.6	47.0
	FDA+HM+PDA	31.3	60.5	29.1	0.3	19.0	42.8
	MT	**53.6**	**86.8**	**61.2**	**2.1**	**45.3**	**62.2**

Table 4. The Mean Average Precision (mAP, in %) of different models and methods for the cross-domain object detection on the validation set of our dataset, i.e., for SASI-RANDOM to TGRS-HRRSD-Dataset [9] adaptation. The best mAPs are highlighted in bold.

Model	Method	mAP	mAP_{50}	mAP_{75}	mAP_s	mAP_m	mAP_l
Faster R-CNN	Source	33.3	56.9	35.1	**2.0**	19.2	45.9
	FDA	32.9	58.6	34.5	1.6	19.8	44.3
	HM	27.9	55.9	23.8	0.6	15.8	39.4
	PDA	35.9	64.7	36.3	1.0	22.9	47.8
	FDA+HM+PDA	30.3	57.6	27.9	0.3	16.7	42.6
	MT	**52.9**	**81.0**	**63.8**	1.9	**45.0**	**61.5**
FCOS	Source	32.6	62.0	31.1	2.9	23.6	41.2
	FDA	33.2	62.3	32.9	1.9	20.7	44.1
	HM	30.4	59.1	27.2	0.3	17.5	41.8
	PDA	36.0	66.3	35.4	1.6	25.5	46.5
	FDA+HM+PDA	32.3	62.5	31.4	0.8	20.2	43.4
	MT	**47.4**	**81.6**	**49.7**	**4.0**	**38.2**	**59.9**

Table 5. The Mean Average Precision (mAP) in percentage for various models and methods for the cross-domain object detection on the validation set of our dataset. The results are presented for the adaptation of SASI-UNITY to LEVIR [21] adaptation. The best mAPs are highlighted in bold.

Model	Method	mAP	mAP_{50}	mAP_{75}	mAP_s	mAP_m	mAP_l
Faster R-CNN	Source	8.0	21.2	4.9	0.9	9.1	11.6
	FDA	8.2	21.1	5.2	1.7	8.6	11.1
	HM	8.4	22.2	4.7	1.5	9.3	10.9
	PDA	8.4	22.6	5.3	1.2	9.3	11.5
	FDA+HM+PDA	7.7	20.6	4.5	1.3	8.5	11.0
	MT	**12.3**	**32.9**	**8.4**	**8.4**	**13.1**	**14.4**
FCOS	Source	9.2	23.6	5.5	1.4	11.2	10.6
	FDA	9.2	24.2	5.3	2.9	11.0	11.1
	HM	9.0	23.2	4.8	2.1	11.0	10.8
	PDA	8.8	22.9	4.7	2.1	10.0	10.9
	FDA+HM+PDA	9.7	24.3	5.7	2.4	11.5	**11.5**
	MT	**11.7**	**30.6**	**6.9**	**6.0**	**14.9**	10.7

Table 6. The Mean Average Precision (mAP) in percentage for various models and methods for the cross-domain object detection on the validation set of our dataset. The results are presented for the adaptation of SASI-ZERO to LEVIR [21] adaptation. The best mAPs are highlighted in bold.

Model	Method	mAP	mAP_{50}	mAP_{75}	mAP_s	mAP_m	mAP_l
Faster R-CNN	Source	6.6	18.8	3.5	0.8	7.9	10.1
	FDA	6.4	19.1	3.3	1.0	7.8	9.4
	HM	7.5	21.9	3.8	1.6	8.4	10.0
	PDA	7.3	21.7	3.7	1.3	8.3	10.6
	FDA+HM+PDA	6.9	20.3	3.4	1.4	8.3	9.1
	MT	**14.8**	**40.3**	**8.6**	**8.2**	**18.3**	**14.6**
FCOS	Source	7.2	23.6	2.5	1.9	8.7	9.4
	FDA	7.8	24.8	3.0	2.5	9.4	9.8
	HM	8.7	26.1	3.8	2.7	9.8	10.6
	PDA	8.3	25.0	3.5	2.5	9.4	10.2
	FDA+HM+PDA	8.2	25.5	2.8	2.5	9.1	10.6
	MT	**12.2**	**32.6**	**6.8**	**5.7**	**15.2**	**11.8**

Table 7. The mean Average Precision (mAP) in percentage for various models and methods for the cross-domain object detection on the validation set of our dataset. The results are presented for the adaptation of SASI-RANDOM to LEVIR [21] adaptation. The best mAPs are highlighted in bold.

Model	Method	mAP	mAP_{50}	mAP_{75}	mAP_s	mAP_m	mAP_l
Faster R-CNN	Source	6.0	17.3	3.1	0.5	7.3	9.3
	FDA	6.6	19.5	3.2	1.8	7.6	9.0
	HM	7.0	20.2	3.8	0.9	8.1	9.4
	PDA	7.2	22.2	3.3	1.4	8.7	9.7
	FDA+HM+PDA	7.3	20.6	4.1	1.4	8.4	10.0
	MT	**15.8**	**42.5**	**10.2**	**9.1**	**19.9**	**15.2**
FCOS	Source	8.9	26.5	3.6	3.3	10.4	10.0
	FDA	9.2	**27.2**	3.5	3.7	10.0	10.6
	HM	8.6	25.9	3.2	3.0	9.8	10.7
	PDA	8.7	24.8	4.0	2.2	9.7	10.5
	FDA+HM+PDA	8.8	25.2	4.4	3.4	9.9	10.5
	MT	**9.5**	25.5	**5.0**	**3.8**	**10.8**	**12.5**

Table 8. The Mean Average Precision (mAP, in %) for cross-domain object detection on the validation set of our dataset, i.e., those adapted from SASI-UNITY to airbus-ship-detection [26]. The best mAPs are highlighted in bold.

Model	Method	mAP	mAP_{50}	mAP_{75}	mAP_s	mAP_m	mAP_l
Faster R-CNN	Source	17.1	30.1	17.6	2.0	29.4	47.6
	FDA	15.2	28.0	15.5	2.4	26.2	40.6
	HM	17.4	31.5	**18.4**	2.5	30.3	47.0
	PDA	15.5	27.5	16.8	1.8	27.2	44.4
	FDA+HM+PDA	**17.6**	32.5	17.9	2.7	**30.4**	45.9
	MT	16.7	**33.0**	15.7	**3.1**	25.5	**50.9**
FCOS	Source	16.6	30.1	17.3	1.8	28.7	47.3
	FDA	17.5	34.9	16.0	3.6	29.0	43.1
	HM	18.4	34.3	18.5	3.4	31.5	45.2
	PDA	18.0	33.3	18.4	3.2	30.7	46.6
	FDA+HM+PDA	19.2	**36.7**	18.4	**3.7**	32.3	47.5
	MT	**20.6**	35.1	**22.2**	2.4	**36.0**	**59.5**

Table 9. The Mean Average precision (mAP, in %) for cross-domain object detection on the validation set of our dataset, i.e., those adapted from SASI-ZERO to airbus-ship-detection [26]. The best mAPs are highlighted in bold.

Model	Method	mAP	mAP_{50}	mAP_{75}	mAP_s	mAP_m	mAP_l
Faster R-CNN	Source	17.2	32.7	17.5	1.9	27.6	51.7
	FDA	17.7	35.7	16.0	2.2	28.8	50.4
	HM	**19.9**	**38.4**	**19.0**	**2.6**	**32.3**	54.7
	PDA	18.6	37.2	16.9	2.3	30.2	52.9
	FDA+HM+PDA	18.7	36.2	18.1	2.3	30.3	53.9
	MT	13.7	30.4	10.7	1.2	18.2	**57.0**
FCOS	Source	18.8	36.8	17.9	2.3	31.5	52.2
	FDA	18.7	36.9	17.4	2.8	31.3	51.0
	HM	**20.5**	**40.9**	**18.7**	**3.3**	**33.9**	53.8
	PDA	19.7	39.0	18.6	2.7	32.8	53.7
	FDA+HM+PDA	19.7	38.9	18.2	**3.3**	33.5	52.9
	MT	17.6	35.4	16.5	1.9	28.9	**57.2**

Table 10. The Mean Average Precision (mAP, in %) for cross-domain object detection on the validation set of our dataset, i.e., from those adapted SASI-RANDOM to airbus-ship-detection [26]. The best mAPs are highlighted in bold.

Model	Method	mAP	mAP_{50}	mAP_{75}	mAP_s	mAP_m	mAP_l
Faster R-CNN	Source	17.6	32.5	16.8	1.7	28.4	52.3
	FDA	17.3	35.7	14.5	2.7	27.8	48.1
	HM	**19.1**	**38.3**	**17.3**	**3.0**	**31.3**	50.9
	PDA	18.1	37.3	15.6	2.4	29.6	50.4
	FDA+HM+PDA	18.4	37.7	16.1	2.8	29.8	49.8
	MT	16.3	33.2	14.4	1.4	22.4	**61.9**
FCOS	Source	18.8	37.0	17.0	3.0	31.6	48.6
	FDA	18.0	37.4	14.7	3.2	29.2	48.3
	HM	**19.4**	39.3	17.2	**3.4**	**32.9**	49.9
	PDA	19.1	39.4	16.7	3.3	31.5	50.0
	FDA+HM+PDA	**19.4**	**39.7**	16.8	3.2	31.9	50.6
	MT	**19.4**	37.8	**18.7**	2.3	32.3	**57.3**

4.4. Observation and Analysis

In the experiments conducted with the TGRS-HRRSD-Dataset [9] as the target data, it was observed that (apart from the mAP_s values of the experimental comparison group where UNITY was used as the source data and FCOS as the model, as well as the experimental comparison group where RANDOM was used as the source data and Faster R-CNN as the model) the mAP values of each group were the highest for the MT method. Similarly, in the experiments using LEVIR [21] as the target data (except for the mAP_l values of the experimental comparison group with UNITY as the source data and FCOS as the model, as well as the mAP_{50} values of the experimental comparison group with RANDOM as the source data and FCOS as the model), the mAP values of each group were again the highest for the MT method. However, in the experiments where airbus-ship-detection [26] was used as the target data, no clear pattern was observed in the experimental results, and most of the highest mAP values could be attributed to the use of domain-adaptive augmentation methods.

Taken together, we can make the following observations.

(1) In the comparison between Faster R-CNN and FCOS, the Faster R-CNN model produced better results. This is because two-stage algorithms involve an additional step of generating candidate regions, which one-stage algorithms do not have. In tasks where real-time performance is not required, two-stage algorithms generally perform better.

(2) Among the six methods compared, the MT method stood out as the domain-adaptive algorithm with the most effective detection in both experiments when using TGRS-HRRSD-Dataset and LEVIR as target domain datasets. Although the results of the experiments with airbus-ship-detection as the target domain dataset do not show a clear pattern, our framework for domain-adaptive object detection algorithms still improves the detection results of training with synthetic data. This text demonstrates how synthetic data can overcome the challenge of limited real data and highlights the effectiveness of the algorithmic framework.

(3) In our generated datasets, SASI-UNITY outperformed SASI-ZERO and SASI-RANDOM in detecting small objects, but it was weaker in detecting large objects. This is because the ship models in SASI-UNITY are derived from real ships, allowing for more control over the size proportions of ships in the images, thus resulting in more small objects. On the other hand, SASI-ZERO and SASI-RANDOM were generated using Stable Diffusion-based prompts, and since ships are the central content of the images, they tend to be larger in size. Generating small objects in a target-oriented manner is generally more challenging than controlling the generation of large objects with the Stable Diffusion model.

(4) In the detection experiments conducted on the airbus-ship-detection dataset as the target domain, the domain-adaptation algorithm exhibited comparatively less enhancement in the detection results compared to the experiments conducted on the other two target domain datasets. This discrepancy can be attributed to the fact that synthetic image datasets were employed to address the issue of insufficient data, whereas the airbus-ship-detection dataset encompasses a significantly larger volume of data than both the TGRS-HRRSD-Dataset and LEVIR datasets, as well as the synthetic image datasets. Consequently, the improvements achieved by the domain-adaptation algorithm in terms of the detection results remained relatively minor. In future research, it would be advisable to broaden the scope of the target domain datasets and augment the number of synthetic images to further corroborate this perspective.

5. Conclusions

In this study, we presented a synthetic image dataset developed using Unity, along with two synthetic image datasets generated by Stable Diffusion. The datasets cover different ship types, geographical conditions, and numbers of ships, among other controllable factors. Detailed labeling and statistics were performed on the datasets images. We performed domain-adaptive object detection experiments from our synthetic image datasets to the TGRS-HRRSD-Dataset, LEVIR, and airbus-ship-detection datasets. Our experiments demonstrate that synthetic data can complete the issue of insufficient real data, and that unsupervised domain-adaptive object detection algorithms can enhance the performance of synthetic data object detection. This provides value for research and practical applications in the field of aerial remote sensing image processing. However, our study has limitations. Firstly, the scope of object detection is limited to ships, and it has not been extended to encompass other aerial remote sensing targets. Furthermore, we have not conducted more comprehensive comparative experiments to precisely determine the scenarios in which domain adaptation yields the maximum improvements. These aspects present potential avenues for future research.

Author Contributions: Conceptualization, Y.W.; methodology, Y.W.; validation, W.G., Z.T., Y.Z., and Q.Z.; formal analysis, Y.W.; investigation, Y.W.; data curation, Y.W., Z.T., Y.Z., and Q.Z.; writing—original draft preparation, Y.W.; writing—review and editing, Y.W.; visualization, Y.W.; supervision, L.W. and Z.G.; funding acquisition, L.W. and Z.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding. The APC was funded by Xiamen University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding authors.

Acknowledgments: We are very grateful for the contributions and comments of the peer reviewers.

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

References

1. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
2. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
3. 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 Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
4. Kuznetsova, A.; Rom, H.; Alldrin, N.; Uijlings, J.; Krasin, I.; Pont-Tuset, J.; Kamali, S.; Popov, S.; Malloci, M.; Kolesnikov, A. The open image dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *Int. J. Comput. Vis.* **2020**, *128*, 1956–1981. [CrossRef]
5. 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.
6. Zhu, H.; Chen, X.; Dai, W.; Fu, K.; Ye, Q.; Jiao, J. Orientation robust object detection in aerial images using deep convolutional neural network. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, 27–30 September 2015; pp. 3735–3739.
7. Cheng, G.; Han, J. A survey on object detection in optical remote sensing images. *ISPRS J. Photogramm. Remote Sens.* **2016**, *117*, 11–28. [CrossRef]
8. Li, K.; Wan, G.; Cheng, G.; Meng, L.; Han, J. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS J. Photogramm. Remote Sens.* **2020**, *159*, 296–307. [CrossRef]
9. Zhang, Y.; Yuan, Y.; Feng, Y.; Lu, X. Hierarchical and robust convolutional neural network for very high-resolution remote sensing object detection. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 5535–5548. [CrossRef]
10. Bordes, F.; Shekhar, S.; Ibrahim, M.; Bouchacourt, D.; Vincent, P.; Morcos, A. Pug: Photorealistic and semantically controllable synthetic data for representation learning. *Adv. Neural Inf. Process. Syst.* **2024**, *36*, 45020–45054.
11. Reiher, L.; Lampe, B.; Eckstein, L. A sim2real deep learning approach for the transformation of images from multiple vehicle-mounted cameras to a semantically segmented image in bird's eye view. In Proceedings of the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 20–23 September 2020; pp. 1–7.
12. Khan, S.; Phan, B.; Salay, R.; Czarnecki, K. ProcSy: Procedural Synthetic Dataset Generation towards Influence Factor Studies of Semantic Segmentation Networks. In Proceedings of the CVPR Workshops, Long Beach, CA, USA, 16–20 June 2019; p. 4.
13. Chen, Y.; Li, W.; Sakaridis, C.; Dai, D.; Van Gool, L. Domain adaptive faster r-cnn for object detection in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3339–3348.
14. Saito, K.; Ushiku, Y.; Harada, T.; Saenko, K. Strong-weak distribution alignment for adaptive object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 6956–6965.
15. Deng, J.; Li, W.; Chen, Y.; Duan, L. Unbiased mean teacher for cross-domain object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 4091–4101.
16. Khodabandeh, M.; Vahdat, A.; Ranjbar, M.; Macready, W.G. A robust learning approach to domain adaptive object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 480–490.
17. Yao, X.; Zhao, S.; Xu, P.; Yang, J. Multi-source domain adaptation for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 3273–3282.
18. Li, W.; Liu, X.; Yuan, Y. Sigma: Semantic-complete graph matching for domain adaptive object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 5291–5300.
19. Li, Y.-J.; Dai, X.; Ma, C.-Y.; Liu, Y.-C.; Chen, K.; Wu, B.; He, Z.; Kitani, K.; Vajda, P. Cross-domain adaptive teacher for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 7581–7590.
20. Razakarivony, S.; Jurie, F. Vehicle detection in aerial imagery: A small target detection benchmark. *J. Vis. Commun. Image Represent.* **2016**, *34*, 187–203. [CrossRef]
21. Zou, Z.; Shi, Z. Random access memories: A new paradigm for target detection in high resolution aerial remote sensing images. *IEEE Trans. Image Process.* **2017**, *27*, 1100–1111. [CrossRef] [PubMed]
22. Liu, Z.; Yuan, L.; Weng, L.; Yang, Y. A high resolution optical satellite image dataset for ship recognition and some new baselines. In Proceedings of the International Conference on Pattern Recognition Applications and Methods, Porto, Portugal, 24–26 February 2017; pp. 324–331.

23. 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 Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part III 14; 2016; pp. 785–800.
24. Li, J.; Qu, C.; Shao, J. Ship detection in SAR images based on an improved faster R-CNN. In Proceedings of the 2017 SAR in Big Data Era: Models, Methods and Applications (BIGSARDATA), Beijing, China, 13–14 November 2017; pp. 1–6.
25. Long, Y.; Gong, Y.; Xiao, Z.; Liu, Q. Accurate object localization in remote sensing images based on convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 2486–2498. [CrossRef]
26. Airbus. Airbus Ship Detection Challenge. 2019. Available online: https://www.kaggle.com/c/airbus-ship-detection (accessed on 31 July 2018).
27. Lam, D.; Kuzma, R.; McGee, K.; Dooley, S.; Laielli, M.; Klaric, M.; Bulatov, Y.; McCord, B. xview: Objects in context in overhead imagery. *arXiv* **2018**, arXiv:1802.07856.
28. Yang, M.Y.; Liao, W.; Li, X.; Cao, Y.; Rosenhahn, B. Vehicle detection in aerial images. *Photogramm. Eng. Remote Sens.* **2019**, *85*, 297–304. [CrossRef]
29. Xian, S.; Zhirui, W.; Yuanrui, S.; Wenhui, D.; Yue, Z.; Kun, F. AIR-SARShip-1.0: High-resolution SAR ship detection dataset. *J. Radars* **2019**, *8*, 852–863.
30. Wei, S.; Zeng, X.; Qu, Q.; Wang, M.; Su, H.; Shi, J. HRSID: A high-resolution SAR image dataset for ship detection and instance segmentation. *IEEE Access* **2020**, *8*, 120234–120254. [CrossRef]
31. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 2672–2680.
32. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013** arXiv:1312.6114.
33. Ho, J.; Jain, A.; Abbeel, P. Denoising diffusion probabilistic models. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 6840–6851.
34. Song, J.; Meng, C.; Ermon, S. Denoising diffusion implicit models. *arXiv* **2020**, arXiv:2010.02502.
35. BwandoWando. Face Dataset Using Stable Diffusion v.1.4. 2022. Available online: https://www.kaggle.com/dsv/4185294 (accessed on 8 September 2022).
36. Meijia-Escobar, C.; Cazorla, M.; Martinez-Martin, E. Fer-Stable-Diffusion-Dataset. 2023. Available online: https://www.kaggle.com/dsv/6171791 (accessed on 21 July 2023).
37. Johnson-Roberson, M.; Barto, C.; Mehta, R.; Sridhar, S.N.; Rosaen, K.; Vasudevan, R. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? *arXiv* **2016**, arXiv:1610.01983.
38. Sun, X.; Zheng, L. Dissecting person re-identification from the viewpoint of viewpoint. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 608–617.
39. Hu, Y.-T.; Chen, H.-S.; Hui, K.; Huang, J.-B.; Schwing, A.G. Sail-vos: Semantic amodal instance level video object segmentation-a synthetic dataset and baselines. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3105–3115.
40. Wang, Q.; Gao, J.; Lin, W.; Yuan, Y. Learning from synthetic data for crowd counting in the wild. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 8198–8207.
41. Wang, Z.J.; Montoya, E.; Munechika, D.; Yang, H.; Hoover, B.; Chau, D.H. Diffusiondb: A large-scale prompt gallery dataset for text-to-image generative models. *arXiv* **2022**, arXiv:2210.14896.
42. Doan, A.-D.; Jawaid, A.M.; Do, T.-T.; Chin, T.-J. G2D: From GTA to Data. *arXiv* **2018**, arXiv:1806.07381.
43. Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V. CARLA: An open urban driving simulator. In Proceedings of the Conference on Robot Learning, Mountain View, CA, USA, 13–15 November 2017; pp. 1–16.
44. Savva, M.; Chang, A.X.; Dosovitskiy, A.; Funkhouser, T.; Koltun, V. MINOS: Multimodal indoor simulator for navigation in complex environments. *arXiv* **2017**, arXiv:1712.03931.
45. Cabezas, R.; Straub, J.; Fisher, J.W. Semantically-aware aerial reconstruction from multi-modal data. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 2156–2164.
46. Gao, Q.; Shen, X.; Niu, W. Large-scale synthetic urban dataset for aerial scene understanding. *IEEE Access* **2020**, *8*, 42131–42140. [CrossRef]
47. Kiefer, B.; Ott, D.; Zell, A. Leveraging synthetic data in object detection on unmanned aerial vehicles. In Proceedings of the 2022 26th International Conference on Pattern Recognition (ICPR), Montreal, QC, Canada, 21–25 August 2022; pp. 3564–3571.
48. Shermeyer, J.; Hossler, T.; Van Etten, A.; Hogan, D.; Lewis, R.; Kim, D. Rareplanes: Synthetic data takes flight. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Virtual, 5–9 January 2021; pp. 207–217.
49. Barisic, A.; Petric, F.; Bogdan, S. Sim2air-synthetic aerial dataset for uav monitoring. *IEEE Robot. Autom. Lett.* **2022**, *7*, 3757–3764. [CrossRef]
50. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
51. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
52. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
53. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.

54. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
55. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
56. 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, Republic of Korea, 27 October–2 November 2019; pp. 9627–9636.
57. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1137–1149. [CrossRef]
58. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
59. 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, Republic of Korea, 27 October–2 November 2019; pp. 6054–6063.
60. He , Z.; Zhang, L. Domain adaptive object detection via asymmetric tri-way faster-rcnn. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; pp. 309–324.
61. Shen, Z.; Maheshwari, H.; Yao, W.; Savvides, M. Scl: Towards accurate domain adaptive object detection via gradient detach based stacked complementary losses. *arXiv* **2019**, arXiv:1911.02559.
62. Hsu, C.-C.; Tsai, Y.-H.; Lin, Y.-Y.; Yang, M.-H. Every pixel matters: Center-aware feature alignment for domain adaptive object detector. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; pp. 733–748.
63. Cai, Q.; Pan, Y.; Ngo, C.-W.; Tian, X.; Duan, L.; Yao, T. Exploring object relation in mean teacher for cross-domain detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 11457–11466.
64. Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv* **2022**, arXiv:2204.06125.
65. Betker, J.; Goh, G.; Jing, L.; Brooks, T.; Wang, J.; Li, L.; Ouyang, L.; Zhuang, J.; Lee, J.; Guo, Y. Improving image generation with better captions. *Comput. Sci.* **2023**, *2*, 8.
66. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 10684–10695.
67. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J. Learning transferable visual models from natural language supervision. In Proceedings of the International Conference on Machine Learning, Online, 18–24 July 2021; pp. 8748–8763.
68. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015; pp. 234–241.
69. Ho, J.; Salimans, T. Classifier-free diffusion guidance. *arXiv* **2022**, arXiv:2207.12598.
70. Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv* **2019**, arXiv:1906.07155.
71. Tarvainen, A.; Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1195–1204.
72. Yang, Y.; Soatto, S. Fda: Fourier domain adaptation for semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 4085–4095.
73. Risser, E.; Wilmot, P.; Barnes, C. Stable and controllable neural texture synthesis and style transfer using histogram losses. *arXiv* **2017**, arXiv:1701.08893.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.