applied sciences

MDPI

Article

Small Object Detection in Infrared Images: Learning from Imbalanced Cross-Domain Data via Domain Adaptation

Jaekyung Kim [1], Jungwoo Huh [1], Ingu Park [2], Junhyeong Bak [2], Donggeon Kim [2] and Sanghoon Lee [1,*]

[1] Department of Electrical and Electronic Engineering, Yonsei University, Seoul 03722, Korea
[2] LIG Nex1, Yongin 16911, Korea
* Correspondence: slee@yonsei.ac.kr

Abstract: Deep learning-based object detection is one of the most popular research topics. However, in cases where large-scale datasets are unavailable, the training of detection models remains challenging due to the data-driven characteristics of deep learning. Small object detection in infrared images is such a case. To solve this problem, we propose a YOLOv5-based framework with a novel training strategy based on the domain adaptation method. First, an auxiliary domain classifier is combined with the YOLOv5 architecture to compose a detection framework that is trainable using datasets from multiple domains while maintaining calculation costs in the inference stage. Secondly, a new loss function based on Wasserstein distance is proposed to deal with small-sized objects by overcoming the problem of the intersection over union sensitivity problem in small-scale cases. Then, a model training strategy inspired from domain adaptation and knowledge distillation is presented. Using the domain confidence output of the domain classifier as a soft label, domain confusion loss is backpropagated to force the model to extract domain-invariant features while training the model with datasets with imbalanced distributions. Additionally, we generate a synthetic dataset in both the visible light and infrared spectrum to overcome the data shortage. The proposed framework is trained on the MS COCO, VEDAI, DOTA, ADAS Thermal datasets along with a constructed synthetic dataset for human detection and vehicle detection tasks. The experimental results show that the proposed framework achieved the best mean average precision (mAP) of 64.7 and 57.5 in human and vehicle detection tasks. Additionally, the ablation experiment shows that the proposed training strategy can improve the performance by training the model to extract domain-invariant features.

Keywords: computer vision; object detection; deep learning; YOLO; infrared small-target detection

check for updates

Citation: Kim, J.; Huh, J.; Park, I.; Bak, J.; Kim, D.; Lee, S. Small Object Detection in Infrared Images: Learning from Imbalanced Cross-Domain Data via Domain Adaptation. *Appl. Sci.* **2022**, *12*, 11201. https://doi.org/10.3390/app122111201

Academic Editors: Junchi Yan and Minghao Guo

Received: 11 October 2022
Accepted: 2 November 2022
Published: 4 November 2022

1. Introduction

Object detection aims to retrieve the classes of specific targets along with their size and location in visual data, such as images or videos. In recent years, the rapid evolution of many-core processors has accelerated the development of object detection algorithms based on deep learning techniques. As the most common deep learning framework, convolutional neural network (CNN)-based models exhibit powerful feature extraction capabilities, resulting in state-of-the-art object detectors, such as SSD [1], EfficientDet [2], and YOLO series [3–5]. These existing object detection models are optimized for detecting common objects from visible-spectrum RGB images using large-scale datasets such as ImageNet [6], PASCAL VOC [7], and MS COCO [8]. In terms of real-world usage, a variety of object detection applications based on task-specific annotated datasets are being widely adopted. The powerful feature extraction capability of deep learning networks combined with large-scale datasets has led to the high detection performance and robustness of models, allowing the object detection models to be one of the most practical usages of deep learning algorithms. It is thus not only widely used in our daily life, such as in autonomous driving [9], pedestrian detection [10], and face detection [11], but it is also used in industries and military applications, such as defect detection [12], robot automation [13], and target detection [14].

However, in such areas where a large-scale dataset is unavailable, the data-driven nature of deep learning algorithms acts as a weakness.

In this paper, the proposed framework aims to solve a small object detection task in infrared images, where datasets comprised of elements satisfying both “small” and “infrared” conditions are insufficient. Although this task is very important in scenarios such as maritime surveillance, early-warning systems, remote sensing and medical imaging [15,16], there have been few studies on this topic due to the insufficiency of datasets. In terms of small object detection, several works have been proposed, which direct the processing towards candidate areas of the input image [17,18] or extract different scales of features [19], where both approaches are based on modifications of the baseline network architecture. In the case of object detection in infrared images, similar approaches [20,21] have been proposed that optimize the network architecture to reflect characteristics of infrared images. In both cases, the proposed methods assume that there exist sufficient datasets to train the network, which is not applicable to our objective. Synthesizing datasets can be one of the approaches to overcome the data shortage [22]. However, domain shifts should be considered because a distribution mismatch between data of the source and target domains may produce a significant performance drop. In our task, where these multiple challenges coexist, it is impossible to directly apply the methods listed above. Therefore, we aim to solve this complex problem via a domain adaptation strategy that can achieve high accuracy with robustness in multiple domains without adding a complex auxiliary network structure or excessive calculations.

Human beings possess an inherent ability to perceive objects in images or videos almost flawlessly. Deep learning-based object detection models are designed to mimic this ability through the feature extraction capability of CNNs. This implies that every model focuses on the shape and visual characteristics of each visual element and combines the cues to detect the location and category of the target object. Therefore, if a model is able to extract similar visual features from multiple domains (visible light, infrared, colored image, grey-scale images, etc.), its ability to detect objects can be well adopted to other domains where objects share similar visual shapes. Based on this concept, we propose a domain adaptation framework to force a model to extract similar features regardless of the source domain. Maintaining the original modules of the YOLO-based object detection model, we added a domain classifier module, the purpose of which is not to detect the domain, but to force extracted features to be domain-invariant by propagating reversed loss to the feature-extractor network. Through this method, the extracted features of each domain become less domain-discriminable, resulting in the enhanced domain-robustness of the model.

Based on this motivation, we propose an object detection framework that can be adaptively trained using datasets from multiple domains of distinct distribution. The proposed framework adapts feature extraction capability learned from large-scale dataset in available domain and transfer it to be used in distinct domain, while preventing overfitting problem that can occur when training on small-sized datasets. Also, considering the real-world applications, we choose one of the fastest object detection algorithms YOLOv5 [5] as baseline and modify the network in a way that does not affect the inference speed. Our contributions can be summarized as follows:

1. We propose a framework for small object detection in the infrared domain based on YOLOv5, where target objects can be detected without adding excessive or complex calculations.
2. We propose an auxiliary domain classifier with a corresponding training procedure, through which the model can be trained using multi-domain datasets with instance number imbalance.

2. Related Works

2.1. Object Detection

The goal of object detection is to determine object instances in the image and return their categories with spatial positions and ranges through the bounding box format. Modern object detection algorithms are mostly based on deep learning approaches, which are divided into two main categories. The first category is the two-stage detectors such as faster R-CNN [23] and mask R-CNN [24]. In this category, the object detection process is divided into two stages. In the first stage, the model searches for candidates of detection targets and their feature information. Then, each candidate is fed into the network to perform classification and regression, identifying the location, size and category of each region. The second category is the one-stage detectors, such as the Single Shot MultiBox Detector (SSD) [1] and the You Only Look Once (YOLO) [25] algorithm. Models in this category perform classification and regression using a single network on an end-to-end basis. As there is no need to generate the candidate regions, object location, size and category can be directly predicted through the one-stage network. The two-stage detectors usually have high localization and object recognition accuracy, as they feed all candidate regions to the network, but the prolonged time of detection hinders the model to be deployed in real-time conditions, which is essential in real-world scenarios. In our work, we adopted a one-stage detector framework, as it can achieve high inference speed with simpler training procedures.

2.2. YOLO

As a family of 1-stage detectors, YOLO has been one of the most popular deep learning models over the last few years. YOLOv1 [25] conceptualized the object detection task as a regression task, simplifying the networks to allow faster models that can be used in real-time. This framework has been further improved until now, where, in this paper, we utilized YOLOv5, which is considered to be the most stable and reliable among released versions. More recent variants of YOLO are being released, while most of the approaches focus on training strategies and model adjustments for generalized multi-scale object detection, which is not suitable in our task. A comparison of YOLOv3, YOLOv4, and YOLOv5 is described in Table 1.

Table 1. Comparison between different YOLO architectures [26].

	YOLOv3	**YOLOv4**	**YOLOv5**
Neural Network Type	Full convolution	Full convolution	Full convolution
Backbone	Darknet-53	Darknet-53	CSPDarknet-53
Neck	FPN	PANet	PANet
Head	YOLO Layer	YOLO Layer	YOLO Layer
Loss Function	Binary Cross-entropy	Binary Cross-entropy	Binary Cross-entropy and Logit Loss Function

YOLOv5 inherits all the consistent ideas of the YOLO series in algorithm design. The model is comprised of three main parts, which are the backbone, neck and head. The backbone extracts feature maps of multiple scales. These extracted multi-scale features are fused through the neck part, which is comprised of three feature fusion networks. The final fused feature maps are then passed to the head part. In this stage, the confidence calculation and bounding-box regression processes are executed for each pixel in the feature maps, yielding bounding-box candidates based on preset anchors. The bounding-box candidates include information of object class confidence, box position coordinate and box size. With pre-defined object class and bounding box confidence thresholds, candidates are filtered via non-maximum suppression (NMS) [27], and final detection results are obtained. Although the performance of YOLO is very stable in various object detection scenarios, due to the pre-defined anchor size and feature fusion scale, it often fails to detect small-sized objects.

2.3. Small Object Detection

There have been numerous approaches to optimize the YOLO model for better performance in small object detection tasks. Moran et al. [28] proposed a small object detection framework modifying YOLOv3. The extracted feature maps from the backbone are fused with the feature maps of other residual layers to form a new feature prediction layer, and the anchors are refined using a clustering algorithm. It achieved high accuracy in object prediction from satelite images. Xu et al. [29] also proposed a YOLOv3-based network for detecting tiny objects in driving scenes by adding auxiliary feature extraction blocks with multiple residual connections. The added network can extract finer and more detailed features compared to the original YOLOv3 backbone, allowing the model to detect small targets more accurately. Cui et al. [30] modified the backbone of the YOLOv4 network by adding an attention module composed of an efficient channel attention module (ECAM) and a convolutional block attention module (CBAM), along with regenerating anchors based on a clustering algorithm. However, the model is focused on a single task of detecting small targets in transmission line faults rather than a generalized object detection task. Zhu et al. [31] proposed TPH-YOLOv5, where they added an additional prediction head based on Transformer to detect objects in different scales. This method showed high mAP performance in aerial-captured scenes. The listed models achieved high accuracy in each target field, but calculation complexity and total model parameters were vastly increased in the exchange of small object detection performance. Additionally, a large-scale dataset for the desired task is required to train each model.

2.4. Infrared Object Detection

As infrared imagery can capture the thermal information of objects, it can be obtained in almost all weather conditions and is less affected by night, occlusion, or fog. Therefore, there has been a steady demand for infrared image object detection algorithms. Traditionally, detection based on hand-crafted features were actively researched, such as filtering-based methods [32,33], contrast-based methods [34], and low-rank-based methods [33]. However, these methods require the tuning of hyperparameters, hindering applications in real-world situations. On the contrary, modern approaches are mainly based on CNN architectures. Two-stage detectors [20] and one-stage detectors [35–37] have both been introduced; most of them focus on the fine-tuning of networks designed for generic objects using infrared datasets.

2.5. Domain Adaptation

In deep learning cases where the statistics of the trained data and the target are distinct, a deterioration in performance can occur. This is called the domain shift phenomenon. Ideally, this can be prevented by labeling large amounts of data in the target domain to additionally train the model in a method similar to transfer learning. However, in many cases, this solution is not available due to the extensive time and labor it requires. Domain adaptation aims to eliminate the domain discrepancy with little supervision on the target domain and improve model performance with annotated source data. Domain adaptation has been widely applied to classification tasks using distance metrics such as maximum mean discrepancy (MMD) to measure the domain shift and supervise the model to learn domain-invariant features [38]. Afterwards, an adversarial training strategy using domain classifiers and a gradient reversal layer (GRL) can be introduced to learn robust domain-invariant features during the training phase. By this means, the domain classifier and backbone model are mutually trained such that a classifier becomes better at distinguishing source and target domain features, while the backbone learns to generate more domain-invariant features as the gradient of classifier is reversely back-propagated to the base network. Domain adaptation for object detection inherits and extends the same training strategy. As a localization task is added to the classification task, the domain adaptation methods are first introduced to the two-stage detectors by applying the adaptation technique to each task [39–41]. There are only a few studies on domain adaptation applied to one-stage detectors. Sasagaa et al. [42] merged multiple pre-trained YOLO models using domain adaptation, while Hnewa et al. [43] introduced multi-scale image level domain adaptation for the YOLO model.

2.6. Knowledge Distillation

As a representative type of model compression and acceleration, the knowledge distillation method effectively learns a small student model from a large teacher model. Especially in classification tasks, the labelled ground-truth is fed into the network in the one-hot encoding style. However, during the distillation process, important knowledge information of the teacher model is transferred to the student model by giving soft-prediction outputs of the teacher model as the ground truth. Through this process, the student model can obtain not only task knowledge of correct answers but the residual probability out of non-answer prediction values. Chen et al. [44] made use of these hinted information from soft-prediction targets to optimize a two-stage detector, while Sasagawa et al. [42] adopted this method in YOLO by distilling multiple pre-trained YOLO networks to one student network, allowing the model to perform object detection in low-light situations.

3. Materials and Methods

In this section, we describe a modified YOLOv5-based network for small object detection in infrared images. The overall architecture of the proposed model is shown in Figure 1. Details of model components, loss functions and the training strategy are described in this section.

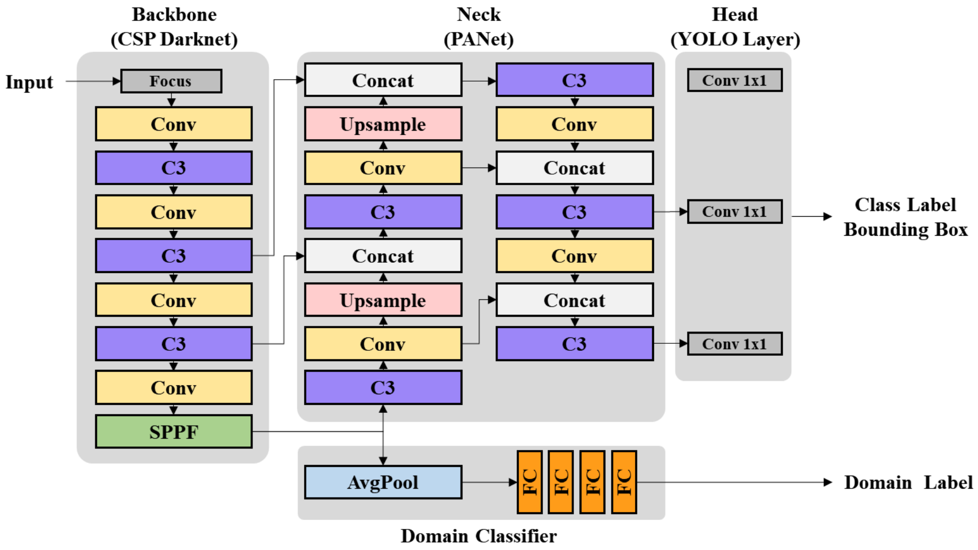


Figure 1. Overall architecture of the proposed framework.

3.1. Model Architecture

The main structure of YOLOv5 can be divided into 3 major parts: the backbone, neck and head. The backbone's main role is in extracting multi-scale visual features from input images by stacking multiple convolution blocks (convolution + batch normalization + sigmoid linear unit (SiLU) activation) and C3 modules [45]. The C3 module can be regarded as a specific implementation of CSPNet with 3 convolution modules. This module ensures that the backbone achieves a rich feature extraction ability along with reducing heavy inference computations that occur by duplicate gradients. A fast implementation of spatial pyramid pooling (SPPF) is attached at the last layer. Compared to the original spatial pyramid pooling (SPP) layer, this layer simplifies the repeated downsampling process by successive max-pooling layers. Its role is in assisting the multi-scale feature expression ability of the backbone and removing the fixed-size constraint of the network. In the neck, the down-sampled feature maps from the backbone are up-sampled to generate multiple new feature maps to detect objects of different scales. In the head, different scales of feature maps are gathered to yield final bounding box candidates and class labels. However, in the case of small object detection, the uppermost module of the head only has role in detecting large bounding boxes. Therefore, we detached the layer in the inference stage for faster calculations and sustained the layer in training stage to allow the model to learn more diverse features.

Additionally, we attached an auxiliary domain classifier to the YOLOv5 base architecture. Based on this module, the domain adaptation strategy is implemented to the proposed framework. The module is comprised of an average pooling layer followed by four fully connected (FC) layers. The feature map is downsampled and flattened to a 256 dimension vector. The first three 256 dimension FC layers are followed by SiLU activation, and the last FC layer gives a 4-channel output of domain confidence. As the backbone extracts base features from input images, the role of the domain classifier is to infer the domain of the input image. To force the base network to extract domain-invariant features, we calculate domain loss with the output of the domain classifier in the direction of confusing the main classifier. The gradients are then backpropagated, allowing the model to extract visual-shape-oriented features, which are more domain-invariant. In this paper, the domain classifier network classifies 4 categories, which are a combination of two binary classes: (1) spectrum (visible light, infrared) and (2) data source (real and synthesized). Therefore, the domain classifier allows the model to learn its feature extraction capability from multiple datasets while maintaining the detection performance in the domain with less data.

3.2. Loss Function

3.2.1. Detection Loss

In the proposed framework, we optimized the original YOLOv5 loss function to better fit a small-sized object detection task. Additionally, an additional loss function for domain adaptation is proposed. The vanilla YOLOv5 model utilizes a detection loss comprised of 3 loss functions: class loss, objectness loss and box loss. The class loss measures cross-entropy to calculate difference between the predicted and ground-truth class label. The objectness loss is based on binary cross-entropy to measure the object presence and confidence of each bounding box candidate. The box loss is a measure of predicted bounding box precision, where the complete intersection over union (CIoU) concept is used. Intersection over union (IoU) is originally based on overlapping proportions of predicted and ground truth bounding boxes, which could not be calculated in non-overlapping cases. CIoU considers the ratio of horizontal and vertical length along with the distance between box central points to overcome this. It not only allows a model to increase the overlapping area of the ground truth and the predicted box but also helps to minimize their central point distance and box aspect ratio difference. However, as shown in Figure 2, the IoU-based metrics may not work well in small-sized object detection. As the location and shape of bounding boxes are defined on a pixel grid, the sensitivity of IoU is insufficient to track bounding box accuracy in small-sized objects. Therefore, we modify the box loss to better optimize a model to detect smaller objects by assuming the bounding boxes as 2D Gaussian distributions [46].

For a bounding box $R = \{c_x, c_y, w, h\}$ where c_x, c_y are box center coordinates and w, h are the width and height of the bounding box, we assume the bounding box is a 2D Gaussian distribution where the center pixel has the highest value and the value decreases from the center to the boundary. Then, the inscribed ellipse of the box can be represented as

$$\frac{(x - c_x)^2}{(\frac{w_x}{2})^2} + \frac{(y - c_y)^2}{(\frac{w_y}{2})^2} = 1, \tag{1}$$

and the probability density function of a 2D Gaussian distribution can be written as

$$f(z|\mu, \Sigma) = \frac{exp(-0.5(z - \mu)^\intercal)\Sigma^{-1}(z - \mu))}{2\pi|\Sigma|^{0.5}}, \tag{2}$$

where z denotes the coordinate location and μ, Σ each denote the mean vector and the covariance matrix of a Gaussian distribution. When

$$(z-\mu)^{\intercal}\Sigma^{-1}(z-\mu)=1, \tag{3}$$

the represented ellipse is a density contour of the Gaussian distribution. This denotes that the bounding box R can be modeled into a 2D Gaussian distribution $\mathcal{N}(\mu,\Sigma)$ by calculating the equation as

$$\mu=\begin{bmatrix}c_x\\ c_y\end{bmatrix}, \Sigma=\begin{bmatrix}(\frac{w_x^2}{4}) & 0\\ 0 & (\frac{w_y^2}{4})\end{bmatrix}. \tag{4}$$

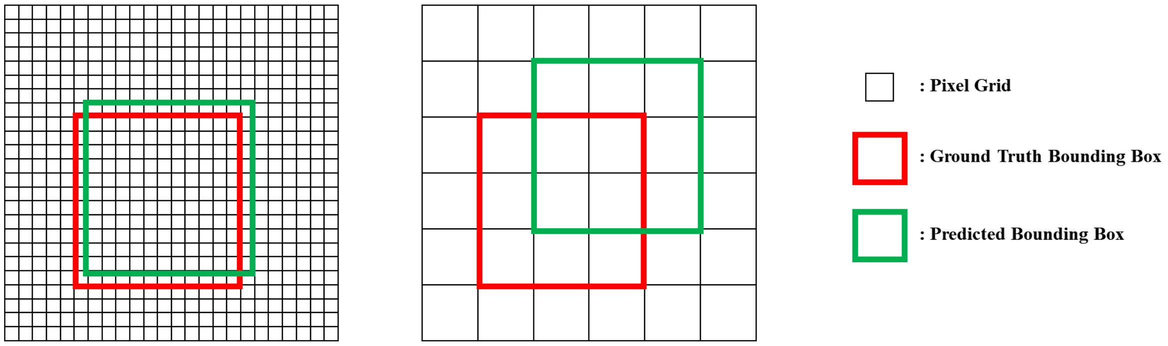
To calculate the loss between two Gaussian distributions, we can use the second-order Wasserstein distance. The Wasserstein distance between $\mu_1=\mathcal{N}_1(\mu_1,\Sigma_1)$ and $\mu_2=\mathcal{N}_2(\mu_2,\Sigma_2)$ is defined as

$$W_2^2(\mu_1,\mu_2)=\left\|\left([c_{1x},c_{1y},\frac{w_1}{2},\frac{h_1}{2}]^{\intercal},[c_{2x},c_{2y},\frac{w_2}{2},\frac{h_2}{2}]^{\intercal}\right)\right\|_2^2. \tag{5}$$

To use this distance as a loss metric, it must be normalized. Therefore, it is normalized as an exponential form, yielding the normalized Wasserstein loss (WL) as:

$$WL(\mathcal{N}_1,\mathcal{N}_2)=exp\left(-\frac{\sqrt{W_2^2(\mu_1,\mu_2)}}{C}\right), \tag{6}$$

where C is an empirically set constant, which is set as 12.8 in this paper. In conclusion, the final detection loss is comprised of the sum of the class loss, objectness loss and Wasserstein Loss.



(a) Normal Sized Object **(b) Small Sized Object**

Figure 2. The sensitivity difference of IoU on normal-sized objects and small-sized objects. As the location of bounding boxes can only change discretely, the sensitivity of IoU is much lower in small-sized objects.

3.2.2. Domain Loss

Domain loss is comprised of two elements: domain classification loss and domain confusion loss. Both are calculated with the softmax value of the domain classifier output. First, the domain classification loss is used to train the domain classifier module. In the same manner as the class loss in detection loss, the distance between the classifier prediction and the domain label is measured with cross-entropy. Second, the domain confusion loss is used in the domain adaptation stage. The trained domain classifier would give us 4 domain probabilities of input data origin. For domain adaptation, we use these values to train the model to extract domain-invariant features. If the extracted features from the backbone are indistinguishable between domains, it would imply that the features are domain-invariant. This state can be considered as an equilibrium state where the 4 output values of the domain classifier are identical, which is 0.25, being the softmax value. Inspired from the knowledge distillation studies, we tried to give hints to the base model of domain discriminability using the minor values of the domain classifier output. Therefore, the domain confusion loss is designed as the L2 distance between domain classifier output values and normalized identical domain confidences ($[0.25, 0.25, 0.25, 0.25]$). This loss is backpropagated after freezing the trained domain classifier, only in the domain adaptation stage.

3.3. Training Process

The overall training process is described in Figure 3. Using domain-annotated datasets from multiple domains in training, the overall process is divided into 3 stages.

In the first stage, the base model is trained with multiple datasets without considering the input domain. The domain category can include "visible light" and "infrared" images sourced from "real condition" and "synthesized condition". Based on the MS COCO pre-trained weights, the model is trained to detect objects using only the detection loss. Domain classifier layers are frozen at this stage.

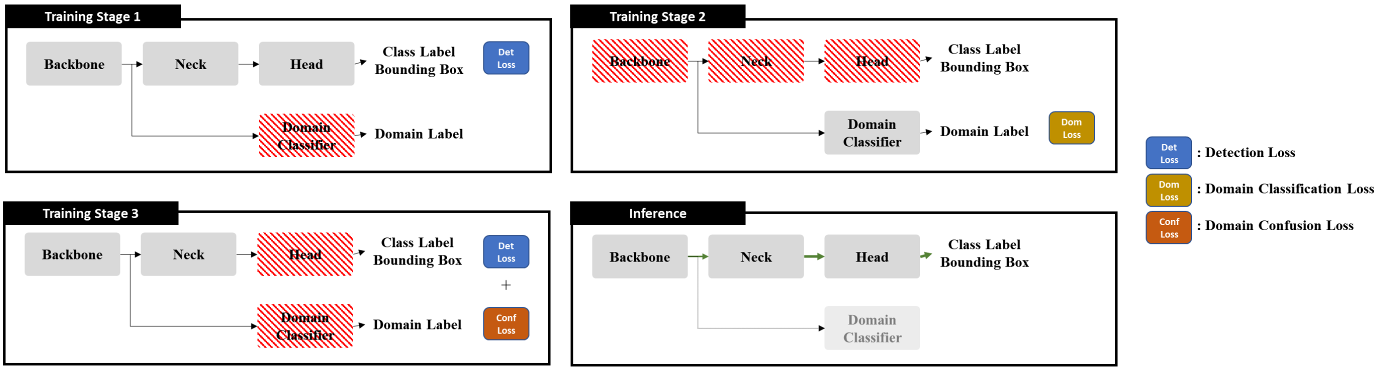


Figure 3. Three training stages of the proposed framework. Modules painted in red stripes denote frozen layers, where model weights are not updated during the stage. Loss elements used to calculate gradients are denoted at each stage.

In the second stage, all the base modules, including the backbone, neck, and head, are frozen. The domain classifier module is the only trainable part at this stage. Using the domain classification loss, the classifier learns to identify the source domain of the input data using the extracted features of the backbone.

The third stage can be summarized as a domain adaptation process. In this stage, the head module and domain classifier are frozen. However, the loss of the domain classifier is substituted to domain confusion loss at this stage. With the total loss being the sum of the detection loss and the domain confusion loss, the model learns to extract domain-invariant features while maintaining the detection performance of the base model. In our approach, a data domain is classified into two sets of two binary categories of spectrum and data source. Therefore, the model learns the ability to extract shape-derived features regardless of input domain while dealing with the domain shift between real and synthetic data to obtain high accuracy. After the three stages of training, the domain classifier is detached for calculation efficiency.

3.4. Datasets

There exists only a few open-source datasets for infrared small object detection tasks. In order to train and test our proposed model, we made use of multiple datasets including "small" or "infrared" instances. Additionally, we constructed additional instances to supplement the data shortage satisfying both criteria. Number of extracted or generated images of each dataset is depicted in Table 2, where samples from both domains are shown in Figure 4. A detailed description is given below.

Table 2. Number of used images in each dataset. Images containing small-sized objects in human- and vehicle-related categories are sampled for training.

Dataset	Image Count
MS COCO [8]	3231
FLIR ADAS [47]	3086
VEDAI [48]	2538
DOTA [49]	1680
Generated Visible Light	21,225
Generated Infrared	3000

3.4.1. Open Datasets

As individual open datasets have different annotation label categories, we extracted instances containing vehicle- and human-related objects. All instances were resized or cropped to 640 × 640 images.

- Visible Light Datasets: Considering the infrared target application, all visible light images were converted to greyscale via post-processing. For pre-training of the base network, MS COCO [8] was used. Image instances which contained small-sized objects (<30 × 30 pixels) in the vehicle and human (person) categories were chosen. Additionally, the VEDAI [48] and DOTA [49] datasets are used in training. Both datasets are based on aerial imagery containing small objects. Images containing vehicle categories were selected for both training and benchmarks. The size of the target objects were concentrated from 12 × 12 pixels to 25 × 25 pixels.
- Infrared datasets: The Teledyne FLIR Free ADAS Thermal Dataset v2 [47] provides pairs of thermal and visible spectrum images for object detection. Among the instances, images containing people, bikes, cars, buses, and trucks were extracted. The infrared images were additionally annotated as "infrared, real", and RGB pairs were annotated as "visible light, real" for domain adaptation training.

3.4.2. Synthetic Data Generation

The quality, quantity, and scene diversity of data impacts the base performance of deep learning-based models. Although we adopted a domain adaptation scheme, it is impossible to train a model with only a few data. Therefore, we propose a method to generate data synthetically and use the outputs in training.

- Synthetic Visible Light Images: Acquiring images and annotating objects are labor-consuming tasks. Although the small target detection datasets in the visible light spectrum are relatively abundant, in such scenarios as aerial surveillance and distant target detection, it is very difficult to generate a dataset for a specific scenario. Therefore, we used Blender to generate synthetic visible-light object detection datasets. Using 3D Google Map data, we simulated real-world terrain and buildings via 3D mesh. The target object images were fused in random locations with automatic annotations of object class and location. Here, 3D models of vehicles were used to obtain more than 20,000 images.
- Synthetic Infrared Images: Capturing infrared images requires professional equipment, making the acquision process very difficult. Although long-wave infrared range (LWIR) images are relatively simple to obtain, obtaining infrared images from other spectra, such as the mid-range infrared range (MWIR), is not an easy task. Therefore, we used the infrared signature simulation software MuSES to render MWIR synthetic images. After giving the material information and environmental conditions to the 3D object modeling software, we placed the object on a grid and rendered corresponding infrared images at each azimuth and elevation angle. The distance between the sensor and object ranged from 20 m to 3 km. Car and tank models were used to obtain object renderings. Additionally, real-world MWIR images were captured using the FLIR A8580 Compact MWIR Camera. A total of 6 scenes comprised of 3000 images were

acquired. The simulated infrared target images were fused to these infrared images, yielding a synthetic MWIR dataset.

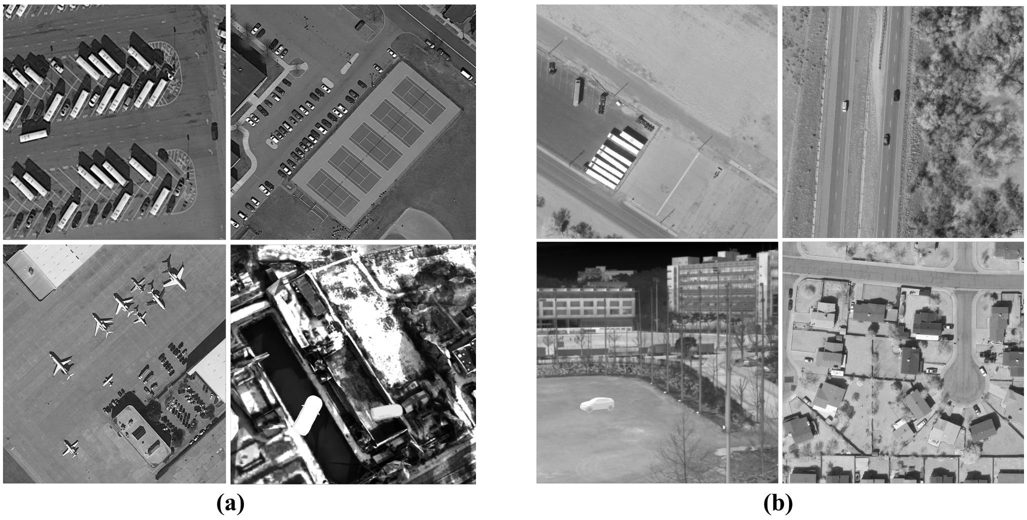


Figure 4. Sample images from datasets. (**a**) Images in visible light spectrum. (**b**) Images in infrared spectrum.

4. Results and Discussion

4.1. Experiment Environment

In the experiments, an NVIDIA RTX 2080 GPU was used to train and evaluate the proposed model. Ubuntu-18.04 LTS was used as a base operating system, with an NVIDIA CUDA v11.3 and an NVIDIA cuDNN v8 to accelerate GPU operations. All scripts were written in python 3.8 with Pytorch v1.12.1. Every dataset was split into 80:20 for training and validation. The size of the input image was 640 × 640 with a batch size of 16, and the training epochs were set to 1000 at each stage. The stochastic gradient descent (SGD) optimizer was used to optimize the network parameters. In training stages 1 and 2, the parameters of the network were optimized with a initial learning rate of 0.01, momentum of 0.8, and weight decay of 0.0005 every 50 epochs. In training stage 3, which is a fine-tuning stage, the initial learning rate was set to 0.001, with a momentum of 0.8 and with no weight decay. Additionally, the batch size was reduced to 4 in training stage 3.

4.1.1. Evaluation Criteria

For the evaluation of object detection performance, the mean average precision (mAP) index following the MS COCO standard was used. For the benchmark, we compared the mAP of the proposed model with YOLOv5, faster-RCNN [23], and YOLO-Z [19]. As similar approaches of training object detectors using multi-domain datasets are not present, we compared the object detector performance by training the network for two tasks. For the human detection task, MS COCO and Teledyne FLIR dataset pairs were used in training and as benchmarks. For detecting objects in vehicle categories, we used MS COCO, VEDAI, DOTA, and the synthesized dataset jointly in training and as benchmarks. The domain adaptation training procedure was only applied to the proposed method.

4.1.2. Experimental Results

The first experiment compared the proposed framework with three mainstream object detection algorithms in a human detection task. Using human instances of the MS COCO and Teledyne FLIR dataset, this experiment validates the domain adaptation ability of the proposed algorithm. The experimental results are shown in Table 3. The proposed framework showed the best performance among other algorithms. As the relative sizes of objects are not extremely small, and there is only one object category, this task is an easier task compared to generalized object detection. The YOLOv5 model shows relatively high accuracy, while faster-RCNN, which is a two-stage method, fails to yield good performance. It can be interpreted that due to multi-domain training, the model fails to learn both localization and classification cues. In the case of YOLO-Z, the model sacrifices its mid- to large-sized object detection ability in exchange for small object detection accuracy. As the size distribution of target data is rather large in this task, the performance of YOLO-Z deteriorates. The second experiment compares algorithms in the vehicle detection task. Using MS COCO, VEDAI, DOTA, and our dataset, a detection task using multiple labels of vehicles (large vehicles, small vehicles, tanks, planes, helicopters) is trained. This task consists of mainly aerial images with small object size distributions. The experimental results are shown in Table 4. Compared to the first experiment, the overall performance of each model is much worse, as the task is more difficult while the datasets are more complex. Our proposed framework achieved superior accuracy compared to conventional algorithms, indicating that it has a powerful learning ability when dealing with imbalanced, multi-domain datasets.

Table 3. Benchmarks on human detection task.

Model	mAP
YOLOv5	41.3
Faster-RCNN	31.5
YOLO-Z	32.4
Proposed	64.7

Table 4. Benchmarks on vehicle detection task.

Model	mAP
YOLOv5	39.3
Faster-RCNN	21.7
YOLO-Z	26.7
Proposed	57.5

4.1.3. Ablation Test

Ablation test results are shown in Table 5. In this experiment, we compared the proposed framework with the basic YOLOv5 base model, the YOLOv5 model with a modified detection loss, and the full framework with a modified loss and domain adaptation procedure. The performance significantly rises as the proposed module is applied. In the human detection task, small-sized objects are less likely to appear. Therefore, the performance improvement is rather small when adding only the Wasserstein loss. However, in the vehicle detection task, the mAP value rises 14.6%, showing the small object detection ability of the proposed method. Additionally, in both cases, the domain adaptation module provides a significant improvement in performance. This shows the ability of the model to extract domain-invariant features, yielding the robustness of the model along with high accuracy.

Table 5. Ablation test results of the proposed model. WL denotes usage of Wasserstein loss, and DA denotes usage of domain adaptation.

Model	Task	mAP
Base Only	Human Detection	41.3
Base + WL	Human Detection	43.7
Base + WL + DA	Human Detection	64.7
Base Only	Vehicle Detection	39.3
Base + WL	Vehicle Detection	45.9
Base + WL + DA	Vehicle Detection	57.5

5. Conclusions

Aiming at the insufficiency of the existing object detection algorithms for detecting small objects in infrared images, a novel YOLOv5-based framework with a multi-domain training strategy has been proposed, including the use of the new loss function and the domain adaptation module. The modified loss using the Wasserstein distance assumes the bounding boxes as Gaussian distributions, overcoming the IoU sensitivity problem of small-object anchors. Additionally, the domain adaptation strategy increases model robustness, allowing the framework to learn domain-invariant visual features from imbalanced datasets, allowing the deep learning model to solve tasks without a large-scale dataset. In the comparative experiments with other state-of-the-art algorithms, the overall mAP is superior with a significant gap, while the ablation study shows the validity of each added module in performance improvement. The structure of the network can be further utilized in other domain tasks easily, such as in synthetic to real domain adaptation, or maritime, underwater image scenarios, as the proposed scheme is not confined to the infrared domain.

Author Contributions: Conceptualization, methodology, validation, formal analysis, investigation, and writing—original draft preparation, J.K. and J.H.; resources, data curation, writing—review and editing, J.K., J.H., I.P., J.B. and D.K.; project administration, supervision and funding acquisition, S.L. and I.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by LIG Nex1 under the project "AI-based automatic target detection/tracking algorithm research (RZ04EO-001)".

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 author. The data are not publicly available due to copyright.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016; pp. 21–37.
2. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10778–10787.
3. Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
4. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.1093.
5. Jocher, G.; Stoken, A.; Borovec, J.; Changyu, L.; Hogan, A.; Diaconu, L.; Ingham, F.; Poznanski, J.; Fang, J.; Yu, L.; et al. ultralytics/yolov5: v3.1—Bug Fixes and Performance Improvements (v3.1). *Zenodo* **2020**. [CrossRef]
6. 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.
7. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2009**, *88*, 303–338. [CrossRef]
8. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Cham, Switzerland, 2014; pp. 740–755.
9. Sarda, A.; Dixit, S.; Bhan, A. Object detection for autonomous driving using YOLO [You Only Look Once] algorithm. In Proceedings of the 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 4–6 February 2021; pp. 1370–1374.
10. Yi, Z.; Shen, Y.; Zhang, J. An improved tiny-yolov3 pedestrian detection algorithm. *Optik* **2019**, *183*, 17–23. [CrossRef]
11. Chen, W.; Huang, H.; Peng, S.; Zhou, C.; Zhang, C. YOLO-face: A real-time face detector. *Vis. Comput.* **2020**, *37*, 805–813. [CrossRef]
12. Yue, X.; Wang, Q.; He, L.; Li, Y.; Tang, D. Research on Tiny Target Detection Technology of Fabric Defects Based on Improved YOLO. *Appl. Sci.* **2022**, *12*, 6823. [CrossRef]
13. Dos Reis, D.H.; Welfer, D.; De Souza Leite Cuadros, M.A.; Gamarra, D.F.T. Mobile robot navigation using an object recognition software with RGBD images and the YOLO algorithm. *Appl. Artif. Intell.* **2019**, *33*, 1290–1305. [CrossRef]

14. Ju, M.; Luo, H.; Wang, Z.; Hui, B.; Chang, Z. The Application of Improved YOLO V3 in Multi-Scale Target Detection. *Appl. Sci.* **2019**, *9*, 3775. [CrossRef]
15. Teutsch, M.; Kruger, W. Classification of small boats in infrared images for maritime surveillance. In Proceedings of the 2010 International WaterSide Security Conference, Carrara, Italy, 3–5 November 2010; pp. 1–7.
16. Ma, T.; Yang, Z.; Wang, J.; Sun, S.; Ren, X.; Ahmad, U. Infared small target dection network with generate label and feature mapping. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5.
17. Singh, B.; Davis, L.S. An analysis of scale invariance in object detection snip. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 3578–3587.
18. Singh, B.; Najibi, M.; Sharma, A.; Davis, L.S. Scale Normalized Image Pyramids with AutoFocus for Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 3749–3766. [CrossRef] [PubMed]
19. Benjumea, A.; Teeti, I.; Cuzzolin, F.; Bradley, A. YOLO-Z: Improving small object detection in YOLOv5 for autonomous vehicles. *arXiv* **2021**, arXiv:2112.11798.
20. Dai, Y.; Wu, Y.; Zhou, F.; Barnard, K. Asymmetric contextual modulation for infrared small target detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Virtual, 5–9 January 2021; pp. 950–959.
21. McIntosh, B.; Venkataramanan, S.; Mahalanobis, A. Infrared Target Detection in Cluttered Environments by Maximization of a Target to Clutter Ratio (TCR) Metric Using a Convolutional Neural Network. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *57*, 485–496. [CrossRef]
22. Varol, G.; Romero, J.; Martin, X.; Mahmood, N.; Black, M.J.; Laptev, I.; Schmid, C. Learning from synthetic humans. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 109–117.
23. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
24. 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.
25. 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.
26. Nepal, U.; Eslamiat, H. Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs. *Sensors* **2022**, *22*, 464. [CrossRef] [PubMed]
27. Neubeck, A.; Van Gool, L. Efficient non-maximum suppression. In Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06), Hong Kong, 20–24 August 2006; pp. 850–855.
28. Moran, J.; Haibo, L.; Zhongbo, W.; Miao, H.; Zheng, C.; Bin, H. Improved YOLO V3 algorithm and its application in small target detection. *Acta Opt. Sin.* **2019**, *39*, 0715004. [CrossRef]
29. Xu, Q.; Lin, R.; Yue, H.; Huang, H.; Yang, Y.; Yao, Z. Research on Small Target Detection in Driving Scenarios Based on Improved Yolo Network. *IEEE Access* **2020**, *8*, 27574–27583. [CrossRef]
30. Cui, J.; Hou, X. Transmission line fault detection based on YOLOv4 with attention mechanism. *Foreign Electron. Meas. Technol.* **2021**, *40*, 24–29.
31. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, Montreal, BC, Canada, 11–17 October 2021; pp. 2778–2788.
32. Rivest, J.; Fortin, R. Detection of dim targets in digital infrared imagery by morphological image processing. *Opt. Eng.* **1996**, *35*, 1886–1893. [CrossRef]
33. Deshpande, S.D.; Er, M.H.; Venkateswarlu, R.; Chan, P. Max-mean and max-median filters for detection of small targets. In *Signal and Data Processing of Small Targets 1999*; SPIE: Bellingham, CA, USA, 1999; pp. 74–83.
34. Han, J.; Moradi, S.; Faramarzi, I.; Zhang, H.; Zhao, Q.; Zhang, X.; Li, N. Infrared small target detection based on the weighted strengthened local contrast measure. *IEEE Geosci. Remote Sens. Lett.* **2020**, *18*, 1670–1674. [CrossRef]
35. Shao, Y.; Zhang, X.; Chu, H.; Zhang, X.; Zhang, D.; Rao, Y. AIR-YOLOv3: Aerial Infrared Pedestrian Detection via an Improved YOLOv3 with Network Pruning. *Appl. Sci.* **2022**, *12*, 3627. [CrossRef]
36. Liu, X.; Li, F.; Liu, S. Improved SSD infrared image pedestrian detection algorithm. *Electro Opt. Control* **2020**, *20*, 42–49.
37. Dai, X.; Duan, Y.; Hu, J.; Liu, S.; Hu, C.; He, Y.; Chen, D.; Luo, C.; Meng, J. Near infrared nighttime road pedestrians recognition based on convolutional neural network. *Infrared Phys. Technol.* **2019**, *97*, 25–32. [CrossRef]
38. Glorot, X.; Bordes, A.; Bengio, Y. Domain adaptation for large-scale sentiment classification: A deep learning approach. In Proceedings of the ICML 2011, Bellevue, WA, USA 28 June 2011–2 July 2011.
39. 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, 18–22 June 2018; pp. 3339–3348.
40. 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.
41. He, Z.; Zhang, L. Multi-adversarial faster-rcnn for unrestricted object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 6668–6677.
42. Sasagawa, Y.; Nagahara, H. Yolo in the dark-domain adaptation method for merging multiple models. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Cham, Switzerland, 2020; pp. 345–359.

43. Hnewa, M.; Radha, H. Multiscale domain adaptive yolo for cross-domain object detection. *arXiv* **2021**, arXiv:2106.01483.
44. Chen, G.; Choi, W.; Yu, X.; Han, T.; Chandraker, M. Learning efficient object detection models with knowledge distillation. In Proceedings of the Advances in Neural Information Processing Systems 30, Long Beach, CA, USA, 4–9 December 2017.
45. Wang, C.Y.; Liao, H.Y.M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 390–391.
46. Han, Y.; Liu, X.; Sheng, Z.; Ren, Y.; Han, X.; You, J.; Liu, R.; Luo, Z. Wasserstein loss-based deep object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 998–999.
47. FLIR Systems, Inc. Free Flir Thermal Dataset for Algorithm Training. Available online: https://www.flir.com/oem/adas/adas-dataset-agree (accessed on 5 March 2022.).
48. Razakarivony, S.; Jurie, F. Vehicle detection in aerial imagery: A small target detection benchmark. *J. Vis. Commun. Image Represent.* **2016**, *34*, 187–203. [CrossRef]
49. 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.