PROGRESSIVE DOMAIN ADAPTATION WITH CONTRASTIVE LEARNING FOR OBJECT DETECTION IN THE SATELLITE IMAGERY

Debojyoti Biswas
Computer Science
Texas State University
San Marcos, TX 78666
ubq3@txstate.edu

Jelena Tešić
Computer Science
Texas State University
San Marcos, TX 78666
jtesic@txstate.edu

November 1, 2023

ABSTRACT

State-of-the-art object detection methods applied to satellite and drone imagery largely fail to identify small and dense objects. One reason is the high variability of content in the overhead imagery due to the terrestrial region captured and the high variability of acquisition conditions. Another reason is that the number and size of objects in aerial imagery are very different than in the consumer data. In this work, we propose a small object detection pipeline that improves the feature extraction process by spatial pyramid pooling, cross-stage partial networks, heatmap-based region proposal network, and objects localization and identification through a novel image difficulty score that adapts the overall focal loss measure based on the image difficulty. Next, we propose novel contrastive learning with progressive domain adaptation to produce domain-invariant features across aerial datasets using local and global components. We show we can alleviate the degradation of object identification in previously unseen datasets. We create a first-ever domain adaptation benchmark using contrastive learning for the object detection task in highly imbalanced satellite datasets with significant domain gaps and dominant small objects from existing satellite benchmarks—the proposed method results in up to a 7.4% increase in mAP performance measure over the best state-of-art.

Keywords Object Detection · Small Objects · Satellite Imagery · Domain Adaptation · Aerial Imagery · Contrastive Learning

Figure 1: Visual difference between consumer images[1] and aerial images [2]

1 Introduction

There is a growing need for automated object localization and identification systems for overhead imagery for traffic control, national parks, wilderness areas, natural disaster surveillance, agriculture, maritime piracy, etc. Research efforts are underway in precision agriculture [3], emergency rescue [4], terrestrial and naval traffic monitoring [5], and industrial surveillance [6] to integrate accurate automated object localization and identification in overhead systems. The challenge lies in the fact that due to high ground sample distance (GSD), the aerial imagery content varies significantly within the same area of capture or drone flight. Several factors are responsible for this dramatic change, such as

significant changes in light conditions (time of day, season, weather) and the type of terrestrial terrain captured in the imagery. The variation between datasets, including multiple dates, terrains, missions, object distribution and sizes, and lighting conditions, is even higher. Figure 2, (a) show the variations due to image capture time and lighting conditions, Whereas Figure 2(b) illustrate that an object can be as small as 0.01% and as large as 70% of an image. Figure 2(b) also shows a loosely packed nature vs. the densely packed small object characteristics in aerial images. Further, we claim that geographical variance is a critical challenge for domain adaptation tasks on satellite images in Figure 2 (c); we also see that geographical variance can also exist in a singular experimental dataset due to image capture from regions of the world. In figure 2, the domain gap between the source and target datasets from different aspects is also noticeable by investigating row 1 (source dataset) with rows 2 and 3 as the target datasets.

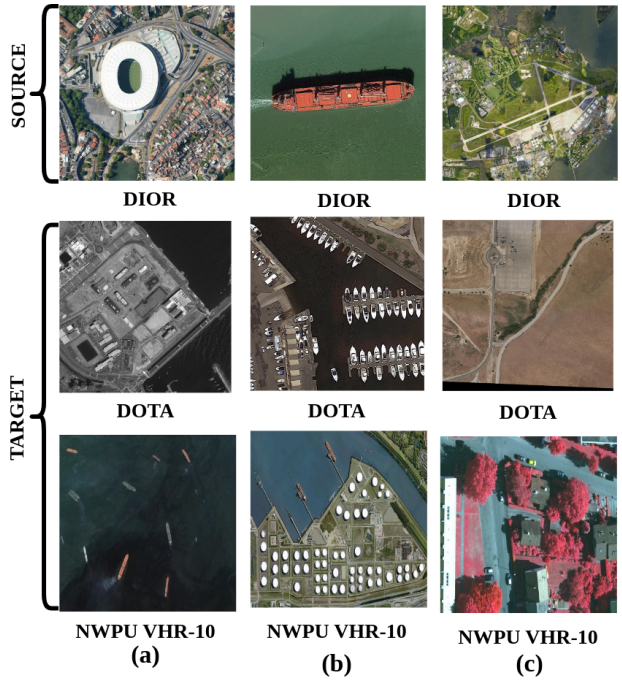

Figure 2: Illustration of variations from different aspects in our experimental datasets: (a) lighting conditions, (b) object shape and scale, and (c) variation due to geographical change

On the other hand, examples of consumer images vs. overhead images are illustrated in Figure 1. Advancements in deep neural networks and greater availability of computational resources have led to enhanced object detection techniques in consumer images [7], and the improvements include improving the Region Proposal Network, better backbone, and integration of the weight-based loss function for hard-example mining. State-of-the-art object detection and domain adaptation (DA) modeling approaches developed for consumer images do not translate to overhead imagery due to visual variation within the image, variation among the images in the collection, the relative object size w.r.t the image, the image size, and the density and number of objects in an image [11]. Recent advances in model detection in overhead images do not address the domain shift and the challenges due to high GSD in an unsupervised setting with a single pipeline [20]. The object identification models for UAV/drone images tend to perform better in low-altitude datasets with relatively fewer small and dense objects compared to our experimental datasets [19].

Deep neural networks require large and diverse amounts of annotated training data to guarantee reliable object localization in unseen datasets. Collecting and annotating aerial datasets has proven difficult and complex due to many small and dense objects per image [21]. Only three datasets with rich class distribution are being used to benchmark the results to date: DOTA2.0 [22], and DIOR [2] are two satellite image collections, and VisDrone is the overhead drone collection [23]. Domain adaptation successfully shares acquired knowledge (regarding annotations and learned models) in the source domain with the target domain. Domain adaptation has risen as one of the approaches to speed up the pseudo-labeling of objects in the target domain using source labels. Domain adaptation for object recognition in consumer image datasets successfully addresses weather, lighting conditions, geological variance, variation in image quality, and cross-camera adaptation by aligning the global feature distribution of data from the origin and target domains [24]. Recent State-of-the-art (SOTA) work of unsupervised domain adaptation for aerial imagery uses the reconstructed feature alignment method instead of adversarial-based feature alignment to avoid background noise alignment [25]. Nevertheless, limited progress in the domain adaptation task has been focused on satellite imagery.

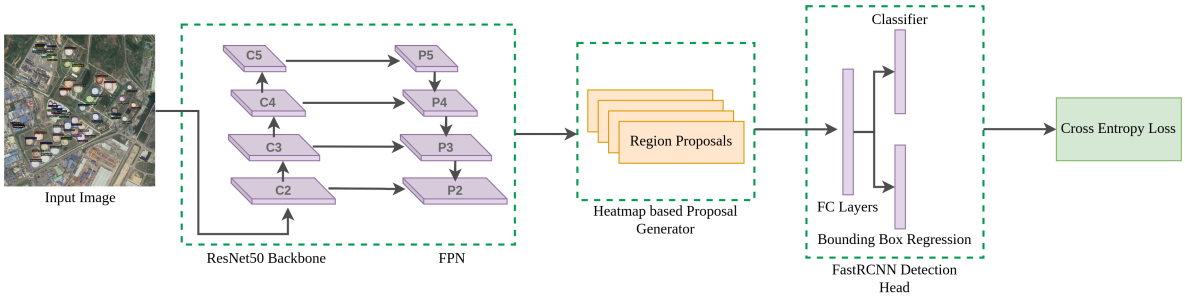

Figure 3: *CenterNet2*: Heatmap based multi-stage small object detection model used as a baseline [34]

Therefore, there is a pressing need to delve into new ideas and techniques to achieve more improved and favorable results.

In this work, we intend to explore contrastive learning on local and global image features to perform feature alignment on task-specific layers. Contrastive learning is a technique that evaluates pair-to-pair relationships by measuring the similarities between different pairs, such as query-positive or query-negative. It groups similar features closely and dissimilar features at a distance in the feature space. We use a random image feature as the query sample and its augmented version as the positive sample. Negative samples are the image features in a mini-batch not part of the query and positive samples. This paper introduces the first domain adaptation benchmark for large-scale satellite image datasets. To reduce the global gap between the source and target domains, we create two intermediate domains using the CycleGAN modeling [26]. Then we extract the local and global feature extraction from feature pyramid network (FPN) layers using the adapted domain adaptation approach. Next, we introduce difficulty-weighted focal loss (DWFL), which uses the number of foreground proposals and amount of neuron activation and assigns a difficulty score for a particular image. Finally, we introduce the noise-contrastive estimation (InfoNCE) [37] loss to produce domain-invariant features. The methodology is outlined in Section 4, and the novelties of the proposed methods are:

1. Using the local-global feature alignment from the source and target datasets using contrastive learning-based domain adaptation.
2. Integration of a novel difficulty estimation method in the domain adaptation pipeline.
3. *Multiple* numbers of negative samples for debiased contrastive learning and object detection tasks.
4. Progressive domain adaptation by creating an intermediate domain and minimizing the domain gap between source and target datasets.

The rest of this article is organized as follows. Section 2 summarizes related work, and Section 3 introduces the proposed methodology for the difficulty-based small object detection and training pipeline. Next, Section 4 describes the contrastive learning approach and the different DA modules in the pipeline. In Section 5 we evaluate the proposed framework using the latest cross-domains detection benchmarks over three different high-altitude (DIOR, NWPU VHR-10, and DOTA2.0) remote sensing datasets and finally summarize the findings in Section 6.

2 Related Work

Object Detection The latest object detection techniques are classified into single-stage or multistage detectors. As the name suggests, single-stage detectors aim to predict object bounding boxes and class labels directly from a single network pass [27, 20, 28]. Single-stage detectors do not have separate neural network modules to generate object proposals and rely on anchor boxes of varying scales and aspect ratios. Single-stage object detection architectures may struggle with accurately localizing small or densely packed objects due to the limited receptive field of the network [22, 29]. Multistage detectors are often more accurate and computationally expensive than single-stage detectors due to their use of region proposal networks (RPN) and the non-maximum suppression technique (NMS) to refine the regions of interest in images [30, 31]. NMS filters out positive instances by rejecting overlapped proposal regions in the image with the help of IOU [32, 33] threshold. CenterNet2 [34] is a heatmap-based two-stage approach with balanced positive/negative samples per batch. It uses a Gaussian filter to create a heat-map peak at the object's center to define the proposal regions [34]. The anchor to the object is the region's center based on location. Thus, the one anchor per object eliminates the need for the non-maximum suppression filtering of the overlapping proposals without affecting the quality of the proposal. Due to its superior characteristics in finding densely packed and small objects, we chose CenterNet2 as our baseline architecture.

Small and dense Object Detection As the object's size decreases, the chances of losing local information in deep layers increase significantly. The outcome of the small object detection depends on how well the backbone network [35, 36] captures the region features from the input image. Next, different scale features from various stages of the backbone have been successfully used for other scale predictions. The feature pyramid network (FPN) layer also helps to strengthen standard spatially rich features by combining semantically rich features by combining low-level and high-level features with the fuse connection and up-sample method. An improved FPN module uses a similarity-based fusion method capable of extracting information for various sizes of instances [39]. Authors argue [41], pixel-level appearance features do not contain enough information to localize small objects in an image, the global context aggregation module and the feature refinement module to build Global Context-Weaving Network are required for optimal performance in small object detections [42]. Hence, context-based feature extraction is more robust for complex object and scene detection and performs better in benchmark datasets [44]. Yang et al. [40] propose *querydet*, which first predicts the coarse locations of small objects on low-resolution features and then computes the accurate detection results using high-resolution features sparsely guided by those crude positions. This work does not rely much on low-level feature queries because it is hard to distinguish small objects with very low resolution on feature maps. The above discussion verifies that small object detection requires global/high-level features to classify objects accurately, which motivates us to use features with high receptive field from later layers in conjunction with the Multi-layered perceptions (MLP) module to calculate the difficulty score for an image (see details in section 3).

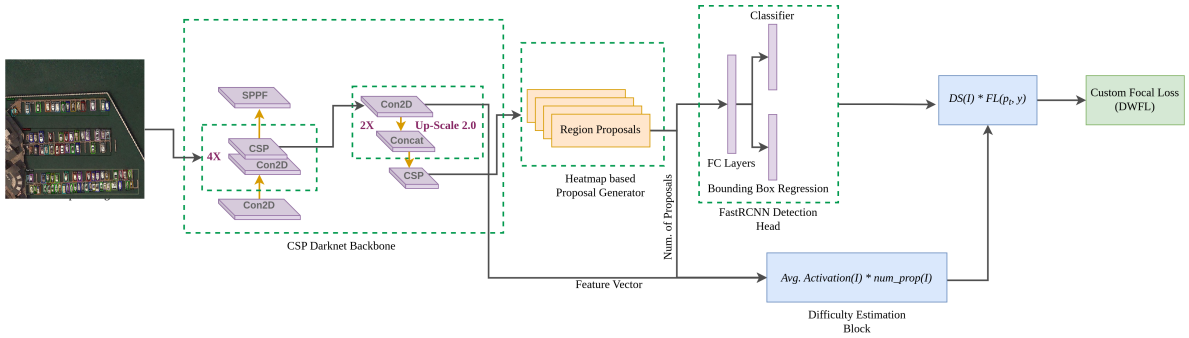

Figure 4: *SOD* model: Small Object Detection model with updated backbone, new loss function, and the difficulty estimation block.

Domain Adaptation for Object Detection Domain Adaptation techniques are used to handle the problem of domain change between source and target data sets. In the last few years, the unsupervised Generative Adversarial Network (GAN) has been critical in solving the domain shift problem. The GAN-based approach expands object detection in consumer images to other domains. The GAN-based domain adaptation model uses Gradient Reversal Layer (GRL) to learn domain invariant features. Hsu et al.[47] use adversarial learning to align the distribution of characteristics between domains and perform progressive domain adaptation to address the problem of significant domain gaps. Saito et al. show domain adaptation should not be rigorous and uniform at every point in feature space, and careful inspection is required to align features that are very close to the inter-class boundary to achieve optimal results [48].

On the other hand, the maximum mean and central moment discrepancy approaches successfully produced domain-invariant features through the alignment of feature space. Long et al. proposed DAN [50], which matches the mean embedding of different domain distributions from different task-specific layers in CNN, and Zeillinger et al. [51] proposed to use means of order-wise moment differences to match the higher-order central moments of probability distributions. State-of-the-art guides knowledge transfer between domains while maintaining consistency of the relevant semantics before and after adaptation [52]. Class-level distribution alignment across the source and target domains was achieved using the Easy-to-Hard Transfer Strategy and a Prototype Feature Alignment Network [53]. However, some adversarial adaptation methods aim to mitigate the need for uniform alignment among all samples; their effectiveness remains constrained by the challenge of achieving accurate unsupervised adaptation.

Contrastive Learning for Domain Adaptation Apart from the traditional GAN networks, contrastive learning [54] in domain adaptation has gained much attention due to its straightforward work process to produce similar features across domains. However, it maintains discriminating characteristics in task-specific features to represent different classes in a singular domain. It uses a similarity function such as cosine similarity or Euclidean distance to measure the similarity between two vectors. Then it represents the image features in the domain invariant feature space.

There are several versions of InfoNCE [37] loss for contrastive learning. Some of them focus on the number of negative examples we need for designing optimal contrastive learning. Introducing a large number of negative examples improves

performance, and Wu et al. [10] argued that real-world datasets often introduce noise, and incorporating too many noisy negative examples yield sub-optimal results. Chuang et al. [13] propose reducing bias in contrastive learning by carefully selecting negative examples. The selection of *False Negative* examples can disturb the learning process and harm the overall performance. Contrastive learning is successful not only in single-source domain adaptation but also in multi-source domain adaptation [57]. Kang et al. [57] consider class information and label the target dataset using the K-means clustering method. Kalantidis et al. [58] use *hard negative mixing* strategy to amplify the effect of negative samples with very minimum overhead computation. Motivated by these works, we choose to use contrastive learning for the domain adaptation task. We present contrastive learning that is less susceptible to False Negatives from highly imbalanced datasets by carefully selecting negative examples.

3 Small Object Detection

The satellite image has a maximum of 400 million pixels and objects are frequently smaller than 100 pixels. A typical patch of the image is 1024×1024, which equals 1.05 million pixels. If an object is 10×10 or 100 pixels, the object's size is < 0.0001 of the area of the image. The success of object detection is contingent on the reliability of the pixel- and object-feature extraction, as well as the RPN-based proposal network within the DNN architecture. An increase in the number of small, densely packed objects raises the possibility of losing pixel-level information during feature extraction. The RPN-based proposal network can miss small objects early in the processing, leading to difficulty in detection at later stages [59]. Furthermore, dense object arrangements result in extra noise from surrounding information during input and more post-processing operations.

3.1 Baseline Model for SOD

Figure 3 shows the *Base* model architecture, a pipeline adopted from the CenterNet2 model [34] with 3 components: Backbone, RPN, and Detection Head. To enhance performance on overhead datasets, the image size, output channels, and IOU were optimized in the FastRCNN Detection Head. The **Backbone** employs ResNet50 as a feature extractor and FPN for multi-scale predictions, combining features from prior layers via residual connections to prevent vanishing gradients problems. The Base model, with ResNet50, achieved leading results on COCO [1] and LVIS [31] datasets. The FPN layer extracts features at different scales from different Backbone layers, shown in the FPN block in Figure 3. The Resnet(C3), Resnet(C4), and Resnet(C5) blocks represent strides of 8, 16, and 32 in the network. P3, P4, and P5 show three scale prediction FPN outputs, which are fed into the RPN block described in the following paragraph for scale predictions.

Region Proposal Network (RPN) in the *Base* model [34] creates region suggestions through heatmaps by applying Gaussian kernels on features from the FPN at different scales. The heatmaps are produced by comparing the max-pool input and the Gaussian kernel output element-wise. Max-pooling highlights each pixel in the feature except for local maxima, which have a value of 1. Each peak in the heatmap represents the center of an object, as shown in Figure 9 (a) and (c). The features at each key point are used to determine the size of objects, leading to accurate bounding boxes even when objects are close or overlap. To improve performance on overhead imagery, the baseline model requires advanced *image augmentation* techniques, enhanced *feature extraction* methods, and an increased *proposals per image* and *detection per image*. **Detection Head** is based on the Faster-RCNN detector [32]. It takes filtered region proposals from the RPN as input and processes them as follows: 1) Convert proposals into 7×7 grids with the same number of channels via region-of-interest (ROI) pooling, 2) Flatten the pooling output and feed it into FCN for final detection output, (N, C) for class predictors for C classes, and $(N, 4)$ for N region proposals bounding boxes.

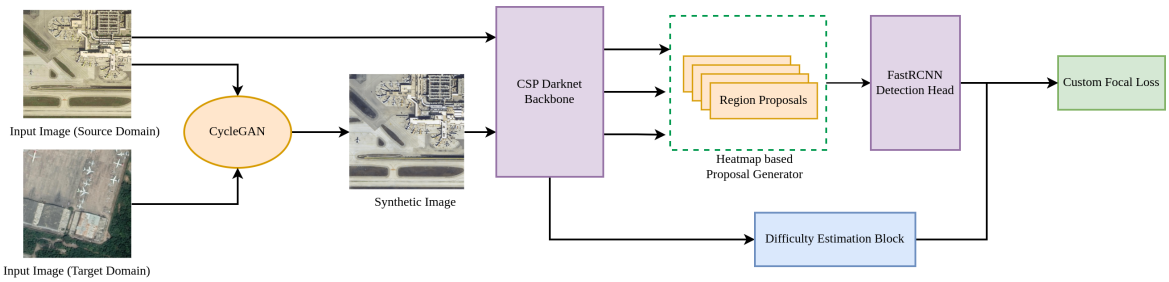

Figure 5: *HeatDA*: Heat Domain Adaptation model with transfer learning and CycleGAN translated image domain adaptation.

3.2 Small Object Detection pipeline

An extended version of our *SOD* model for small object detection [60] improves upon the Base model by optimizing the pipeline for detecting small objects. We replaced the backbone with CSP Darknet [28] and added the Difficulty Estimation block. We also switched to a modified focal loss instead of cross-entropy loss (Figure 4). The RPN module is effectiveness depends on the backbone's performance. If the backbone fails to extract meaningful features for the small object in the image, the RPN module will likely fail to include the small object in the region proposals. Our findings found that 75% of RPN proposals in the Base model were trivial and repetitive (backgrounds, partial objects) as explained in [60]. To enhance the model, we used the CSP Darknet backbone for preserving better semantic information in deeper CNN layers [28, 61]. The system introduces a partial cross stage for low/high-res aggregation and replaces max-pooling with spatial pyramid-pooling for finer feature extraction.

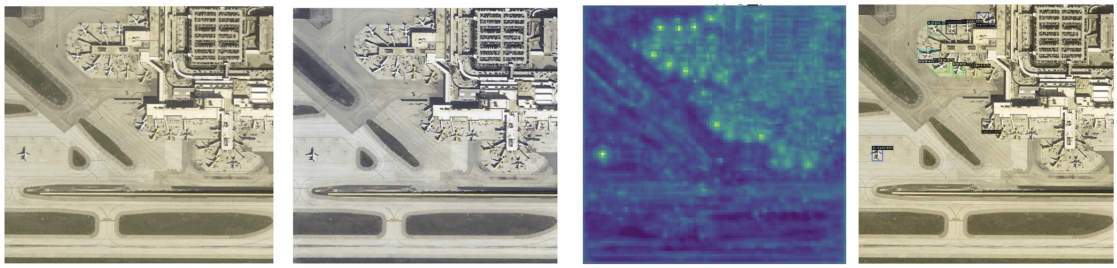

Figure 6: Domain translation from Source(S) to Source as Target(SaT) using CycleGAN [26] and Object Detection from HeatDA model.

The proposed small object detection model in Figure 4 concatenates several low-level features with high-level features channel-wise, thus propagating the semantic information from the lower to higher levels. The region proposal network considers the multi-resolution features from different CSP-Bottleneck layers for proposal generation, as illustrated in Figure 4. **Difficulty Estimator (DE)** module numerically captures the complexity of an image feature based on the number of foreground object proposals and the amount of neuron activation information in the network for an image. It calculates the overall difficulty of each image by taking feature input from different stages of the feature network. The difficulty score (DS) for an FPN feature level with a resolution of $C \times W \times H$ for the image I is calculated in Eq. 1.

$$DS(I) = \frac{1}{L}\sum_{l=1}^{L}\left(\frac{1}{C*W*H}\sum_{c=1}^{C}\sum_{w=1}^{W}\sum_{h=1}^{H} f_{c,w,h}(I)\right) * \ num\_prop(I) \tag{1}$$

The feature output channels, width, and height at FPN level (l) are represented by *C, W, H* respectively. In contrast, L represents the number of FPN levels used to calculate difficulty. The difficulty score (DS) at the FPN level is calculated by dividing the sigmoid linear unit (SiLU) activation values $f_{c,w,h}(I)$ at all pixels in an image (I) by the total dimension of C, W, H. Using this block we calculate the number of total neurons fired for a single image in the forward pass. DS is derived from three FPN levels, averaged, and multiplied by the number of proposals (See Eq. 1). The final DS value for an image (I) is obtained by normalizing the DS value between $[0.5, 1.4]$. The complexity increase for the DS block is minimal, and its computational time, expressed in Big Oh *(O)* notation, is *O(r)*, where r is the number of iterations during training.

$$FL(p_t, y) = \alpha_t * (1-p_t)^{\gamma} * CE(p, y), DWFL(x, p_t, y) = DS(I) * FL(p_t, y), \tag{2}$$

Difficulty weighted Focal Loss is calculated from the difficulty scores for each image, and we propose replacing the loss of cross-entropy with the loss of custom focalization, as illustrated in Figure 4. The difficulty scores are calculated using Eq. 1 for each image by a difficulty estimator block as a weight factor to focus more on complex images with many small objects and a high variation in pixel-level features. The basic form of the focal loss function is outlined in Eq. 2.

$$\forall c \in C, \alpha'_c = -1 * log\left(\frac{|C_c|}{|C_1 \cup C_2 \cup ...|}\right) \Rightarrow \alpha_c = \beta * \frac{\alpha'_c - min(\alpha_c)}{max(\alpha_c) - min(\alpha_c)} \tag{3}$$

The p_t is the probability distribution of the target *t*, and y is the ground truth of the object being a specific class, γ is the modulating factor, α_t is used as a weighting factor, and CE represents the cross-entropy function. We propose a new measure, the Difficulty Weighted Focal Loss (DWFL) in Eq. 2 as a product of difficulty score, $DS(I)$ in Eq. 1, and focal loss for the image, $FL(p_t, y)$ in Eq. 2. The value α is used in the $FL(p_t, y)$ calculation to control the class imbalance problem in our source and target data sets. The α_c is calculated as in Eq. 3 for each class, where the modulating factor α'_c depends on the frequency $|C_c|$ of a particular class in the data set and $|C_1 \cup C_2 \cup C_3 ...|$ is the total number of all instances of all classes in the data set. The normalized α_c values from Eq. 3 are used across different classes c, $c \in C$ to mitigate the imbalance of object class labels. In the experiment section 5 we confirm that the proposed normalization of α_c in Eq. 3 is more effective and gives a stable loss calculation for a highly unbalanced class count in the data set.

4 Domain Adaptation Methods

Different overhead image datasets are usually taken at different geographical locations, and different types of satellites were used to capture images with different orientations under various weather and lighting conditions. There are a handful of annotated overhead image collections [63], and they all have different object class annotations, both in frequency and assigned object labels. This contributes to a large domain gap between our source and target data sets. Object detection performance on target degrades drastically when the domain gap is very w.r.t source dataset. Domain adaptation (DA) methods are key to solving this problem. Using domain adaptation methods, we can perform better in unseen datasets not introduced during the training phase. The self-supervised or unsupervised domain adaptation aims to produce invariant features for a particular class across domains. In the experiment section 5, we confirm that the domain adaptation of the source in the training process improves the object detection performance.

4.1 *HeatDA* model: Heat Domain Adaptation Model

Here, we propose a pre-processing step for the pipeline outlined in Figure 4 and map the source domain into the target domain first, as illustrated in Figure 5, as we have found that closing the source and target gap using progressive domain adaptation leads to better object detection in the previously unseen overhead imagery. Using source and target image examples, we train the CycleGAN domain discriminator [26, 64]. The resulting domain discriminator model translates the source image to the target domain, as illustrated in Figure 6. This additional CycleGAN domain discriminator model illustrated in Figure 5 allows us to align pixel-level features between two domains and use the source image and the translated source image to train the SOD network, as illustrated in Figure 5. The conversion of the source to the target domain allows us to incorporate target-like domain characteristics without relying on the object-level annotations that might or might not be present. Training the *HeatDA* using target-like images helps to align pixel-level semantic information for the source and target domain, thus improving the detection performance (see Figure 6) in the target dataset.

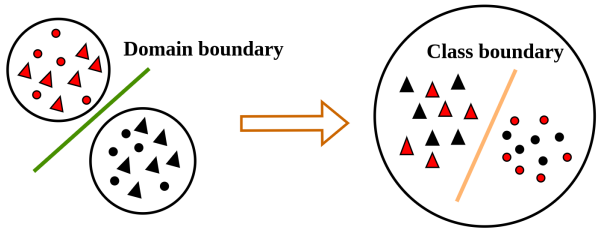

Figure 7: Domain Adaptation with contrastive learning. Here, different colors indicate different domain distributions, and the different shapes represent different classes in a domain. The Green and Orange line represents the domain and class boundary, respectively.

4.2 *LGDA* Model: Local Global Domain Adaptation Model

Contrastive learning [54, 55] is a simple process of measuring pair-to-pair relationships based on the similarities between different pairs, such as query-positive or query-negative. Figure 7 illustrates the functional strategy of contrastive loss. Feature representation of the source and target objects differs in the feature space, and there is a huge gap due to lightning, geographic, weather, and acquisition differences, and the difference is illustrated by a green line in Figure 7. Contrastive learning brings similar points to close together and pushes dissimilar points separate from each other by calculating similarities between pairs [57, 24]. A pair of feature vectors with high similarity are placed close together, and vector pairs with low similarity are placed distantly in feature space. In the ideal case, the contrastive domain adaptation maps the feature space of the source dataset to the target dataset so that the features representing objects

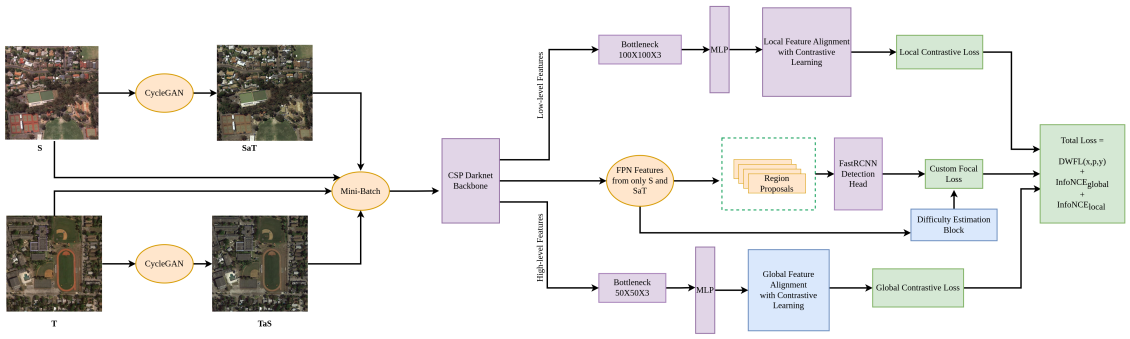

Figure 8: Local and Global Domain Adaptation(LGDA) model with contrastive learning.

in the same class in the source and the target domain dataset are closer together. In this light, we propose to enforce contrastive learning on local features to minimize the domain gap w.r.t the local characteristics in the image i.e. color and texture captured by deep features of the pixel and its nearest surroundings. Here, we produce domain-invariant object features for source and target domains by increasing similarities between *Query-Positive* pair and decreasing similarities between *Query-Negative* pairs. We introduce the Informative Noise Contrastive Estimation (InfoNCE) loss measure of finding similarities and dissimilarities between features in Equation 4. Similarity of two features u and v is captured by a cosine score $sim(u, v) = u^T/(||u|| * ||v||)$. The *Query* is from the source, the *Positive* example is from the translated source domain denoted as SaT in Figure 8, and negative examples are with different classes than the *Query* example and from the target domain. Contrastive learning increases the similarity between *Query* and the *Positive* sample and increases the dissimilarity between *Query* and N negative samples $Negative_n, n \in [1, N]$ s outlined in Eq. 4.

$$InfoNCE = -log\frac{exp(sim(Query, Pos)/\tau)}{\sum_{n=1}^{N} exp(sim(Query, Neg_n)/\tau)} \tag{4}$$

The Informative Noise Contrastive Estimation (InfoNCE) loss measure is low when the similarity between the *Query* and the *Positive* example is high and when then the similarity between Query to all *Negative* examples is low. Using this loss, we are learning domain invariant features. N is the mini-batch size during the training phase, and τ is the temperature that controls the strength of penalties in hard negative samples. Our implementation ensures that we find a similar example as *Positive* case and dissimilar examples as *Negative* cases. Figure 8 illustrates the proposed Local Global Domain Adaptation *LGDA* model and incorporates the proposed approach in the small object detection framework. This architecture focuses on performing domain adaptation on a highly class-label imbalance dataset where labeled objects are small compared to the image size. We added two modules to two new modules: Local Feature Alignment and Global Feature Alignment for contrastive learning. Our proposed model takes input from four different distributions; among them, two are the source and target domains, and the other two are new intermediate domains, Source as Target (SaT) and Target as Source(TaS), from the source and target datasets, respectively, generated from the CycleGAN [26] network to reduce the gap between the source (S) and target (T) domains.

As shown in Figure 8, mini-batch inputs are passed into *CSP DarkNet* backbone. Next, features extracted from the backbone are fed into Local Feature Alignment and Global Feature Alignment modules for calculating *Local Contrastive Loss* and *Global Contrastive Loss*, respectively. However, RPN produces region proposals for object detection tasks from only S and SaT domain features passed from the backbone because we have ground truth for these two domains. The later part of the architecture is a traditional RCNN-style object detector with classification and regression modules.

Local Domain Adaptation focuses on the local features in an image and assumes there is no ground truth object labeling for the target dataset, only for the source dataset. Local features capture low-level descriptions of a pixel and its neighbors in an image. The images from a mini-batch pass through the backbone and local features are saved from the earlier layers of the backbone. The saved local features are in the shape of $256 \times 100 \times 100$, where dimensions are in the form of C, W, H, respectively. To reduce the necessity of computational power and GPU memory and improve similarity computation performance, we pass the features into the bottleneck module as shown in Figure 8 and downgrade the shape to $3 \times 100 \times 100$. In Figure 8, S and T are images drawn from the source and target datasets, respectively, where SaT and TaS are the corresponding images of S and T in translated form, produced by the CycleGAN network. Let us denote the local feature vectors from *S, SaT, T and TaS* as L_k^S, L_k^{SaT}, L_k^T, and L_k^{TaS}, respectively. Where k is the index of the mini-batch. For the adaptation of the S and SaT domain, we select a local feature $L_i^S \in L^S$ as a query and the corresponding feature from $L_i^{SaT} \in L^{SaT}$ as the positive case. Negative cases are all other local characteristics $L_j^{SaT} \in L^{SaT}$, where $j \neq i$. The local domain loss between *S, SaT* and *T and TaS* are calculated in Eq. 5 and 6.

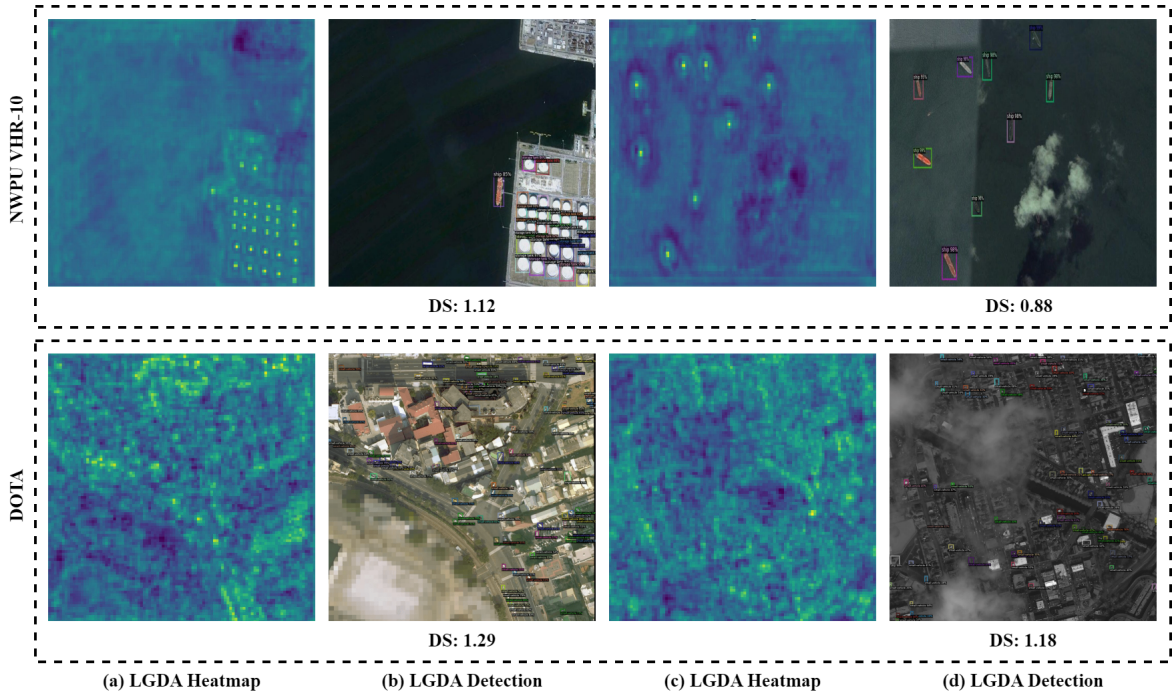

Figure 9: Proposals heatmaps and detection results from the *LGDA* model for DOTA2.0 and NWPU VHR-10 target datasets. Here DS = Difficulty Score for a particular image.

$$InfoNCE_{local}^{S,SaT} = -log\frac{exp(sim(L_i^S, L_i^{SaT})/\tau)}{\sum_{j=1}^{N} exp(sim(L_i^S, L_j^{SaT})/\tau)} - log\frac{exp(sim(L_i^{SaT}, L_i^S)/\tau)}{\sum_{j=1}^{N} exp(sim(L_i^{SaT}, L_j^S)/\tau)}, j \neq i \quad (5)$$

$$InfoNCE_{local}^{T,TaS} = -log\frac{exp(sim(L_i^T, L_i^{TaS})/\tau)}{\sum_{j=1}^{N} exp(sim(L_i^T, L_j^{TaS})/\tau)} - log\frac{exp(sim(L_i^{TaS}, L_i^T)/\tau)}{\sum_{j=1}^{N} exp(sim(L_i^{TaS}, L_j^T)/\tau)}, j \neq i \quad (6)$$

After accumulating loss for all query images in a minibatch, the total bidirectional local domain adaptation loss can be formulated as below in Eq. 7.

$$InfoNCE_{local} = InfoNCE_{local}^{S,SaT} + InfoNCE_{local}^{T,TaS} \quad (7)$$

Global Domain Adaptation relies on the global alignment of features between the source and the target dataset. The global features represent a more abstract formation of objects in the image and are saved from the last layer of the backbone. The shape of the global features is $256 \times 25 \times 25$, where the dimensions are C, W, and H, respectively. Again, the same as for local features, we use a bottleneck module to reduce the size of global features to $3 \times 25 \times 25$. Global features are high-level features in the DNN pipeline. Global domain adaptation and feature alignment are also performed at the mini-batch level to restrict computational and GPU memory expense.

The global feature vectors of training mini-batch input: *S, SaT, T, and TaS* are indexed as G_k^S, G_k^{SaT}, G_k^T and G_k^{TaS}, where k is the index of the mini-batch. The global contrastive loss for S and SaT is calculated by selecting a query sample $G_i^S \in G^S$ and a positive case from the corresponding image feature $L_i^{SaT} \in L^{SaT}$ and vice versa. We take the negative cases as all the other global features $G_j^{SaT} \in G^{SaT}$, where $j \neq i$, and the adaptation formula for S and SaT domains in global feature space is outlined in Eq. 8.

$$InfoNCE_{global}^{S,SaT} = -log\frac{exp(sim(G_i^S, G_i^{SaT})/\tau)}{\sum_{j=1}^{N} exp(sim(G_i^S, G_j^{SaT})/\tau)} - log\frac{exp(sim(G_i^{SaT}, G_i^S)/\tau)}{\sum_{j=1}^{N} exp(sim(G_i^{SaT}, G_j^S)/\tau)}, j \neq i \quad (8)$$

The adaptation of the formula for the T and TaS domains in the global feature space is described in Eq. 9.

$$InfoNCE_{global}^{T,TaS} = -log\frac{exp(sim(G_i^T, G_i^{TaS})/\tau)}{\sum_{j=1}^{N} exp(sim(G_i^T, G_j^{TaS})/\tau)} - log\frac{exp(sim(G_i^{TaS}, G_i^T)/\tau)}{\sum_{j=1}^{N} exp(sim(G_i^{TaS}, G_j^T)/\tau)}, j \neq i \quad (9)$$

The accumulated global domain adaptation loss in a mini-batch is now calculated in Eq. 10.

$$InfoNCE_{global} = InfoNCE^{S,SaT}_{global} + InfoNCE^{T,TaS}_{global} \tag{10}$$

Finally, the *LGDA* model combines the local and global contrastive loss with the detection loss and the final loss function is now calculated as in Eq. 11:

$$TotalLoss = W_1 * InfoNCE_{global} + W_2 * InfoNCE_{local} + DWFL(x, p, y) \tag{11}$$

In the above Eq. 11, the W_1 and W_2 denotes the weight we put on the two different modules.

5 Experiments

We evaluated the proposed approach in the three largest annotated satellite image collections, DIOR [2], DOTA2.0[22], and NWPU VHR-10 [12] with four state-of-the-art domain adaptation models. DIOR is the data set for the source domain, and we used the ground-truth annotation for DIOR in our detection module and model evaluation. DOTA2.0 and NWPU VHR-10 are the target data set that adapts to the local and global domains, as presented in Table 1. We assume that the DOTA2.0 and NWPU VHR-10 annotations are unavailable at the object detection training time, and we use ground truth to evaluate the system's performance only at the testing time. In our experiments, we kept only common classes available in the DIOR and DOTA2.0 datasets. We assigned the same class label to each corresponding class in each dataset, as demonstrated for the reduced DIOR, DOTA2.0, and NWPU VHR-10 dataset in Table 1. We describe the experimental dataset in Subsection 5.1, the setup and implementation details in Section 5.2, the performance comparison in Section 5.3, and lastly, the ablation study in Section 5.4.

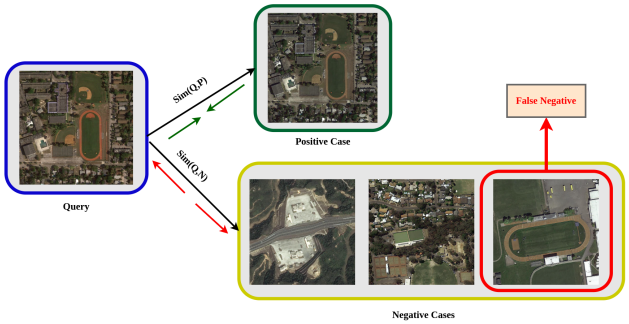

Figure 10: Example of False Negative occurrence in a highly imbalanced dataset.

Class Name	# of Ins. DIOR	# of Ins. DOTA	# of Ins. NWPU VHR-10
Bridge	176	1039	124
Vehicle	2079	85479	598
Harbor	254	5704	224
Storage.T	2623	5416	655
Stadium	40	393	-
Baseball	250	516	390
Track	138	417	163
Basketball	171	358	159
Tennis	580	1662	524
Airport	25	153	-

Table 1: Test set instance distribution statistics of the DIOR, DOTA2.0, and NWPU VHR-10 datasets across common categories.

5.1 Datasets

The **DIOR** data set originally consisted of 24,500 Google Earth images from 80 countries. However, after selecting only common classes, the reduced dataset has 11,402 images. The images varied in quality and were captured in different seasons and weather conditions. The dataset boasts a wide range of spatial resolutions, object sizes, object orientations, and a diverse class distribution, as shown in Fig. 1. The spatial resolution of the images ranges from $[0.5m, 30m]$, with each image measuring 800×800 pixels. The data set has 97,450 annotated objects, classified into ten classes [2]. Out of the 11,402 images, 10,888 are in the training set, and the remaining 512 images are in the testing set.

The **DOTA2.0** dataset comprises 2,430 overhead images sourced from Google Earth, Gaofen-2 (GF-2), and Jilin-1 (JL-1) satellites [22]. The image sizes range from 800×800 to $29,200 \times 27,620$ pixels. The ground sample distance (GSD) in the data set ranges from 0.1 to 0.87 m, and each image contains an average of 220 objects. In the experiment, large images were split into overlapping tiles of 1024×1024 pixels with a 200-pixel overlap, resulting in 23,300 images from the original DOTA2.0 dataset. The reduced DOTA2.0 dataset has 11,551 images in the training set and 3,488 in the validation set, classified into ten classes (See Table 1). In this paper, we interchangeably use DOTA and DOTA2.0 for referring to this dataset.

The **NWPU VHR-10** has in total 800 High-resolution images in the dataset, of which 715 images were collected from Google Earth, and the remaining are very-high-spatial-resolution pan-sharpened color infrared (CIR) images collected from the Vaihingen dataset [16]. The GSD in the dataset ranges from 0.5 to 2m, and the image size range from 800 to 1267. Among the 800 High-resolution images, 650 are positive images with the available target in the annotation. In contrast, the rest of the images do not contain the target object considered a negative image. We only use positive images from the dataset for our Domain Adaptation task experiments. The reduced NWPU VHR-10 dataset has 450 images in the training and 200 images in the test set, distributed among eight classes (See Table 1). In this paper, we interchangeably use NWPU VHR-10 and NWPU for referring to this dataset.

Method	Detector+Backbone	Bridge	Vehicle	Harbor	Storage Tank	Baseball Field	Track Field	B.Ball Court	Tennis Court	Stadium	Airport	DIOR → mAP	DOTA mAP
Baseline [34]	CenNet2 ResNet50	10.5	8.9	42.1	40.6	46.5	31.4	46.7	74.2	0.0	28.3	64.8	32.1
QueryDET [40]	RetinaNet ResNet50	14.1	14.5	38.2	50.8	43.0	33.4	46.6	77.5	5.3	35.1	69.7	35.8
EPM [17]	FCOS ResNet101	10.1	10.8	40.6	47.7	46.2	34.8	48.7	81.9	1.2	35.5	65.5	35.7
MGADA [15]	FCOS VGG16	13.1	10.8	45.9	48.5	46.0	37.7	50.1	84.3	0.0	37.2	66.9	37.3
SAPNET [14]	FCOS ResNet50	10.9	11.0	23.5	24.4	35.3	27.8	32.2	74.1	0.0	22.7	54.7	26.1
MGADA [15]	F-RCNN ResNet101	15.9	14.0	48.1	46.5	47.6	39.3	52.6	87.2	1.8	37.9	72.6	39.4
SOD	CenNet2 Darknet53	13.9	15.8	36.7	48.1	46.3	31.7	45.3	78.6	4.1	32.9	70.1	35.2
HeatDA	CenNet2 Darknet53	14.4	17.4	39.1	50.9	46.1	35.5	48.0	80.1	4.8	37.6	70.1	37.4
LGDA*	CenNet2 ResNet50	22.0	26.8	48.6	59.7	56.9	44.6	56.7	85.4	5.5	38.2	**74.6**	**44.3**
LGDA	CenNet2 Darknet53	24.5	27.9	51.3	62.0	59.2	47.9	58.6	87.8	6.1	38.7	**76.9**	**46.7**
Oracle	Baseline	46.2	40.4	82.6	65.8	64.4	60.0	77.2	93.5	27.2	54.3	**59.7**	**61.2**

Table 2: Quantitative performance comparisons (mAP) across classes for DIOR → DOTA benchmark(IOU=0.5), where DIOR is considered as the source and DOTA as the target dataset. Class-wise performance is presented only for target dataset.

5.2 Experimental Setup and Implementation

Source as Target (SaT) and Target as Source (TaS) synthesized data are created using the PyTorch implementation of the CycleGAN [26, 64] network with the setup: the learning rate was 0.001; the number of training epochs was 2; load size was 800; and the crop size was 640.

The proposed model for object detection architecture *LGDA* is illustrated in Figure 8 and is described in Section 4.2 is an extension of the CenterNet2 [34] and SOD [60] model. The work in SOD uses Darknet53 as the backbone as it is shown to preserve better semantic information from the small objects with the help of Cross-Stage-Partial(CSP) network than the residual-based feature extractor networks [28, 61]; RPN heatmap-based approach to identify dense small objects and remove NMS; and the detection block is Faster-RCNN [32].

We developed our code implementation by leveraging an open-source computer vision library **Detectron2** [18] and some part of **CenterNet2** [34]. We implemented two new DA modules for local and global domain adaptation using the contrastive learning technique. To train our **LGDA** model, we have resized all images to 800×800 pixels, and the mini-batch size in each epoch is set to 8. The loss of InfoNCE in Equation 4 requires the same image from different

domains as in the query and the positive case. To achieve this goal, we created a custom data loader in PyTorch that fetches the exact image of the different domains in a mini-batch. This custom data loader ensures that the Query and Positive examples are in the mini-batch sample during the training. In our experiments, eight is the size of the mini-batch, and the $8 \times 4 = 32$ images were fed into the *LGDA* model in each mini-batch. The number of negative cases and the temperature were set to 7 and 0.12, respectively, for contrastive learning. Our research found that passing the S and SaT for the object detection task during the full training time makes the model over-fitted to the source domain. To eliminate this problem, we implemented random sample selection, randomly selecting 8 of 16 images from S and SaT and passing them on to the object detection module. Li et al. [65] found that Global Average Pooling (GAP) loses important context information when working with satellite images, so we replaced GAP with the bottleneck and MLP module for channel reduction. We have used NVIDIA 2 x RTX 6000 GPU with 49GB of memory, 11^{th} generation Intel® CoreTM i9-11900K @ 3.50GHz × 16 CPU, and 167GB of system memory to carry out all experiments.

5.3 Performance Measures and Comparisons

We use Average Precision (AP), and Average Precision (mAP) to evaluate and compare the models. The precision (P), recall (R), F1 score ($F1$), and Mean Average Precision (mAP) are computed using Eq. 12 and 13. True positives (TP) are instances correctly predicted by the model, false positives (FP) are instances missed by the model, and true negatives (TN) are instances incorrectly predicted by the model. We use an IOU threshold of 0.5 for all experiments, and the number of proposals per image is 256. Precision (P) represents the fraction of relevant instances recovered by the model, while recall (R) measures the fraction of relevant instances correctly identified by the model among all relevant instances.

$$P = \frac{TP}{TP + FP},\ R = \frac{TP}{TP + FN},\ F1 = \frac{2PR}{P + R} \tag{12}$$

$$AP = \sum_{k=0}^{k=n-1} [R(k) - R(k+1)] * P(k),\ mAP = \frac{1}{n} \sum_{k=1}^{k=n} AP(k) \tag{13}$$

The $F1$ score provides a single measure of the model's performance when given a class imbalance dataset. The mAP is calculated as shown in Eq. 13, where n is the number of classes in the test set, and $AP(k)$ is the Average Precision (AP) of class k in the test set. Here, AP is the weighted sum of precision at each threshold (n is the number of thresholds), and the weight is the increase in recall (Eq. 13).

Method	Detector+Backbone	Bridge	Vehicle	Harbor	Storage Tank	Baseball Field	Track Field	B.Ball Court	Tennis Court	DIOR → mAP	NWPU mAP
Baseline [34]	CenNet2 ResNet50	32.5	27.1	76.5	48.4	30.3	66.1	63.6	67.0	66.8	52.0
QueryDet [40]	RetinaNet ResNet50	43.2	33.9	80.4	54.7	37.1	69.5	70.1	72.6	72.4	57.6
EPM [17]	FCOS ResNet101	49.6	39.4	86.7	61.6	43.9	76.1	75.7	77.0	68.9	64.0
MGADA [15]	FCOS VGG16	48.8	37.1	85.5	59.6	42.3	76.8	72.6	77.2	66.5	62.6
SAPNET [14]	FCOS ResNet50	35.8	22.6	70.1	41.4	25.5	60.1	56.5	61.4	60.8	46.7
MGADA [15]	F-RCNN ResNet101	54.2	43.9	88.5	65.0	49.9	77.2	76.6	77.0	69.2	66.8
SOD	CenNet2 Darknet53	47.5	36.4	83.9	54.5	38.5	69.0	72.6	72.8	71.5	59.7
HeatDA	CenNet2 Darknet53	48.8	38.2	85.5	56.9	49.5	70.6	74.5	73.0	71.9	62.1
LGDA*	CenNet2 ResNet50	54.0	43.9	90.5	62.0	53.9	76.2	78.6	81.0	**73.8**	**67.9**
LGDA	CenNet2 Darknet53	57.6	51.5	90.4	65.7	60.1	79.8	81.3	84.7	**74.6**	**71.4**
Oracle	Baseline	69.5	60.9	96.5	79.4	64.3	96.1	93.6	97.8	**60.6**	**84.4**

Table 3: Quantitative performance comparisons (mAP) across classes for DIOR → NWPU VHR-10 benchmark(IOU=0.5), where DIOR is considered as the source and NWPU as the target dataset. Class-wise performance is presented only for the target dataset.

Object Detection Comparisons: To compare our proposed model with recent state-of-the-art (SOTA) models, we set a lower-bound and an upper-bound on the performance for each dataset. We use *CenterNet2* as the baseline/ lower-bound for comparisons, where we use annotations from only the *source* datasets during the training phase to evaluate the target dataset. On the other hand, *oracle* is the upper bound, which uses *CenterNet* as the detection model and uses annotation from *Target* dataset while training to evaluate the target dataset. We choose CenterNet2 as our baseline model because of its ability to work better with small and dense objects leveraging the power of heatmap-based RPN. We use our extended version of the SOD pipeline for small object detection architecture. We have compared our *SOD*

model performance with a recent small object detection SOTA model *QueryDet* [40] in Table 2 and 3. We can see that our model performs very close to the *QueryDet* method for the DOTA dataset, and outperforms the *QueryDet* by 2.1% of mAP on the NWPU VHR-10 dataset. Moreover, using 12GB GPU memory *QueryDet* trains 2 images per batch whereas using *SOD* we can set batch size up to 8 images, so we decided to continue further experiments using SOD as the small object pipeline. As the domain adaptation SOTA models, we used feature alignment DA methods such as MGADA [15] and a spatial attention-based domain adaptation network SAPNet [14] for the performance measurements. Also, we introduced EPM [17], a domain adaptation framework that accounts for each pixel via predicting pixel-wise centeredness and objectness for state-of-the-art comparisons. Later we keep track of the performance improvement of our new proposed model *LGDA* and try to minimize the gap between *LGDA* and *oracle* for domain adaptation task. Except for the *oracle*, all other comparing models use annotation from only the *source* dataset and images from the *source* and *target* datasets during the training phase. Next, we evaluate the performance of the models based on the test set of both *source* and *target* datasets. By this, we show that our proposed UDA models perform satisfactorily on the target domain, and there is no performance degradation in the source domain due to induced noise from the DA operation.

We start our DA evaluation in Table 2, using DIOR as the source and DOTA as the target. Table 2 shows that the CenterNet2-based baseline model gives an mAP of 64.8% on the source and 32.1% mAP on the target dataset. We improve the baseline model with the integration of *Custom Focal Loss*, *Difficulty Estimation Block* and *Strong Backbone* as illustrated in Figure 4 and propose a new model *SOD* verified to work better on satellite imagery [60]. Table 2 shows significant improvements on both datasets for small objects such as harbors, vehicles, and storage tanks. The range of α_c' values in the DIOR dataset is 0.2 to 0.79, and the range of α_c' values in the DOTA2.0 dataset is 0.15 to 0.96, which represents a very tight scaling factor for *FL* in both data sets. The SOD model also shows reasonable promise in performance improvement with a gain of 6.0% and 3.0% of mAP for the DIOR and DOTA datasets, respectively, by dealing with small objects and challenging images. The first step towards DA operation was introducing transfer learning and using CycleGAN-generated composite target images for the DA training. We name the model as *HeatDA*, which uses the target domain pixel-level context and gives 5.3% mAP improvement on the target dataset. Our final proposed model *LGDA* is an extension of *HeatDA* model by adding local and global domain adaptation modules. Compared to the baseline model, our *LGDA* model gains 12.2% and 14.6% of mAP on the target dataset using ReseNet50 and CSP-Darknet53 backbone, respectively. The detection results from *LGDA* model are illustrated in Figure 9, where the proposals generated from heatmaps are shown in Figure 9(a) and (c), and the object detection performance is shown in Figure 9(b) and (d). Figure 9 also shows us the value of image difficulty for a particular image. It is evident that an image with a higher number of objects and pixel diversity score more difficulty value than an image with fewer objects and less pixel variations. Among the other SOTA models, MGADA performs best with an mAP of 39.4%, and SAPNet gives the lowest mAP of 26.1%. However, Our proposed novel contrastive learning method with a small object-focused pipeline helps us to outperform other SOTA models by a minimum margin of 7.3 % on the target dataset.

Method	Backbone	TL	LDA	GDA	DWFL	DOTA	NWPU
Baseline	DarkNet53					32.1	52.0
w/DWFL					✓	35.2	59.7
w/TL		✓			✓	37.4	61.8
w/LDA			✓		✓	40.6	64.2
w/GDA				✓	✓	41.5	66.6
LGDA		✓	✓	✓	✓	46.7	71.4

Table 4: Ablation study for our proposed LGDA method. Here, TL= Transfer Learning, LDA= pixel-level local domain adaptation, and GDA= object-level global domain adaptation.

	Precision				Recall				F1			
Dataset	CenterNet2	SOD	HeatDA	LGDA	CenterNet2	SOD	HeatDA	LGDA	CenterNet2	SOD	HeatDA	LGDA
DOTA	32.1	35.2	37.4	**46.7**	46.9	47.7	50.4	**60.2**	38.1	46.1	42.9	**52.6**
NWPU VHR-10	52.0	59.4	61.8	**71.4**	58.8	63.5	**67.9**	66.8	61.1	61.3	64.7	**69.0**

Table 5: Comparison of Precision, Recall, and F1 score between our proposed models for the DOTA and NWPU VHR-10 target datasets.

Next, we use Table 3 to demonstrate the DA performance and SOTA comparison for DIOR and NWPU VHR-10 datasets. The DIOR and NWPU VHR-10 are two high-variability image datasets, as shown in Table 1 captured from

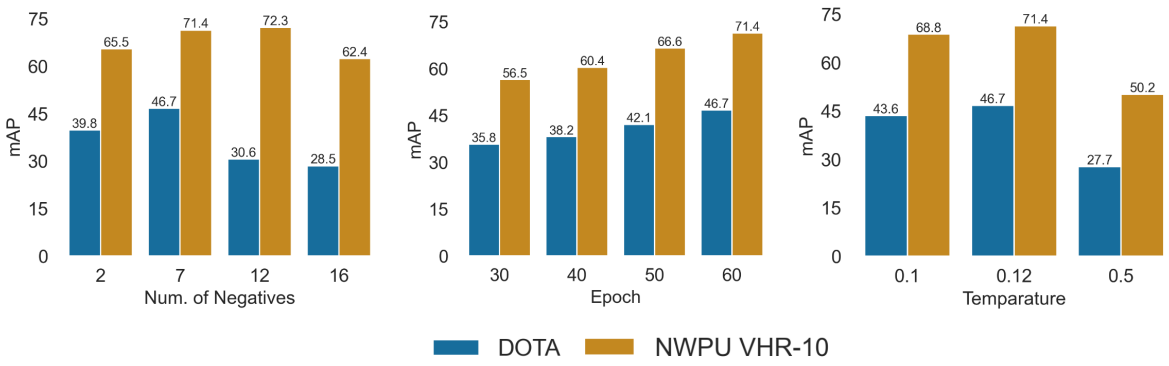

Figure 11: Ablation Study on Hyper-parameters: (a) mAP vs. Negative Examples, (b) mAP vs. Epochs, and (c) mAP vs.Temperature.

satellites. Here, we evaluate target dataset performance over eight different categories. It is observed during experiments that we have not only demonstrated outstanding performance on the target dataset but also we have attained a notable 74.6% mAP(refer to Table 3) on the source dataset. Our baseline CenterNet2 method trained on only source dataset archives 52.0% of mAP, whereas our LGDA method achieves 71.4% of mAP using contrastive learning with local and global domain adaptation. Also, we have a +4.6% gain margin compared to the best state-of-the-art *MGADA* method. Moreover, compared to the baseline model, we were able to shrink the performance gap between the oracle and our model from 32.4% to 13.0% using the Local-Golbal DA. Table 3 and Figure 9(b) and (d) demonstrate the effectiveness of our method in detecting objects from challenging and less frequent categories, including track, bridge, and basketball fields. Table 3 further illustrates that a meticulously designed backbone can augment the performance by approximately +3.5% on the target domain, mainly when dealing with densely populated objects.

Finally, we perform an in-depth performance analysis in Table 5 using Precision, Recall, and F1-Score for the four proposed models. For the DOTA dataset, we gain the optimal result in all metrics using the final version of the *LGDA* model, with a Precision, Recall, and F1-Score of 46.7, 60.2, and 52.6, respectively. However, for NWPU, we got the optimal value of Precision and F1-Score from the *LGDA* model and Recall from the *HeatDA* model.

5.4 Ablation study

In this section, we first perform an ablation study on each component of our proposed *LGDA* method to demonstrate the effectiveness of each element, as shown in Table 4. Our ablation study is carried out on two target datasets: DOTA and NWPU VHR-10. Next, we perform an ablation study on each important hyper-parameters; the summary of the ablation study is illustrated in Figure 11. Table 4 shows that integrating Difficulty Weighted Focal Loss (DWFL) was crucial for our proposed model as we made 3.1% and 7.7% increase in mAP for DOTA and NWPU, respectively. The intermediate version of our proposed model is *HeatDA*, where we investigate the amount of efficiency we can leverage from Transfer Learning, Synthetic images, and DWFL. The Synthetic image generated from GAN networks provides primary pixel-level color and texture information, and we can notice a slight gain of mAP in Table 4 for both target datasets. Then we integrate contrastive learning into our DA process and propose two new modules; the first is for aligning local features between the source and target domain, and the latter is for aligning a more abstract view of features with a high-receptive field. The local feature alignment with contrastive learning (LDA) helped us gain over 8% and 12% mAP on DOTA and NWPU VHR-10 by putting a weight of 0.1 on the loss function. On the other side, from the GDA module, we even get better results than the LDA with a weight of 0.01 on contrastive learning. Finally, integrating all small modules, we propose the LGDA method, which achieves 46.7% and 71.4% of mAP on DOTA and NWPU VHR-10 target datasets, respectively.

Next, we look for the optimal value for the number of negative examples for the hyper-parameters study, as shown in Figure 11(a). We started our experiments with a value of 2, and we can see that the model did not perform well with fewer negative examples in the contrastive loss. We increased the value to 7 to generalize learning and recorded our best performance in the DOTA target dataset. Increasing the negative examples further did not help us learn in target datasets due to the imbalanced nature of the data set. The vehicle class dominates our DOTA data set, as shown in Table 1. Increasing the number of negative examples also increases the chances of getting false negative (FN) examples, as shown in Figure 10 in contrastive learning. In highly imbalanced datasets such as DOTA2.0, the FN example makes the model biased toward a particular class, and the model's overall performance degrades significantly. However, there was

a slight improvement for the NWPU VHR-10 dataset with negative example 12, as NWPU is much more balanced than DOTA. The adverse effect of bias is evident in Figure 11(a) when trained with 16 negative examples. After careful inspection and to reduce computational expense, we set the number of negative measures equal to 7 for both datasets.

Second, our baseline model was trained for 30 epochs, and the LGDA model was trained for 30 more epochs as we added two more loss functions for contrastive learning. As illustrated in Figure 11(b), our experiments found that training for more epochs does not significantly improve performance. Therefore, all results were recorded with 60 training epochs. Lastly, the temperature value in contrastive loss is susceptible and small changes in value can drastically change the outcome. This is evident in Figure 11(c); placing a 50% penalty on contrastive loss dramatically reduces performance, and using a 12% penalty shows the optimal result on target datasets.

6 Conclusion and Future Work

Object detection in aerial images is one of the most challenging tasks in computer vision research because many small and overlapped objects exist in the photos. The success of DNN object localization depends on a large amount of annotated training data and a reliable feature extractor module in the pipeline. This paper presents a robust feature extractor that captures balanced low- and high-level features for small objects. Next, we offer the heat-map-based region proposal module to grab small things better. The domain gap in satellite images is more significant than in consumer images because of weather conditions, geographic changes, and camera orientations. We perform progressive domain alignment by creating two intermediate domains, w.r.t. source and target datasets. The proposed method *LGDA* performed exceptionally well with more than 60% mAP for several classes such as *storage tank*, *harbor*, and *tennis court* in the DOTA and NWPU VHR-10 target data sets. We also use contrastive learning to adapt to local and global domains. Careful selection of the training pipeline, the number of negative samples, the down-sampling strategy, and the temperature value can improve the effectiveness of contrastive learning. Finally, we validate our approach in two challenging high-variability target datasets that showed significant performance gain over available state-of-the-art methods. For the DOTA and NWPU VHR-10 target datasets, we outperformed the latest state-of-the-art *MGADA* method by +7.3% and +4.6% mAP, respectively. Next, we plan clustering-based pseudo-labeled for target objects, de-biased instance-level domain adaptation, and unknown class discovery for satellite images.

References

[1] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[2] Ke Li, Gang Wan, Gong Cheng, Liqiu Meng, and Junwei Han. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS Journal of Photogrammetry and Remote Sensing*, 159:296–307, 2020.

[3] João Valente, Bilal Sari, Lammert Kooistra, Henk Kramer, and Sander Mücher. Automated crop plant counting from very high-resolution aerial imagery. *Precision Agriculture*, 21(6):1366–1384, 2020.

[4] Scott Workman and Nathan Jacobs. Dynamic traffic modeling from overhead imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12315–12324, 2020.

[5] Debojyoti Biswas, M M Mahabubur Rahman, Ziliang Zong, and Jelena Tešić. Improving the energy efficiency of real-time dnn object detection via compression, transfer learning, and scale prediction. In *2022 IEEE International Conference on Networking, Architecture and Storage (NAS)*, pages 1–8, 2022.

[6] Jia Liu, Jianjian Xiang, Yongjun Jin, Renhua Liu, Jining Yan, and Lizhe Wang. Boost precision agriculture with unmanned aerial vehicle remote sensing and edge intelligence: A survey. *Remote Sensing*, 13(21):4387, 2021.

[7] Xiang Long, Kaipeng Deng, Guanzhong Wang, Yang Zhang, Qingqing Dang, Yuan Gao, Hui Shen, Jianguo Ren, Shumin Han, Errui Ding, et al. Pp-yolo: An effective and efficient implementation of object detector. *arXiv preprint arXiv:2007.12099*, 2020.

[8] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

[9] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[10] Wu, Chuhan and Wu, Fangzhao and Huang, Yongfeng Rethinking InfoNCE: How Many Negative Samples Do You Need? *arXiv preprint arXiv:2105.13003*, 2021

[11] Payal Mittal, Raman Singh, and Akashdeep Sharma. Deep learning-based object detection in low-altitude uav datasets: A survey. *Image and Vision computing*, 104:104046, 2020.

[12] Cheng, G. & Han, J. A survey on object detection in optical remote sensing images. *ISPRS Journal Of Photogrammetry And Remote Sensing*. **117** pp. 11-28, 2016.

[13] Chuang, C., Robinson, J., Lin, Y., Torralba, A. & Jegelka, S. Debiased contrastive learning. *Advances In Neural Information Processing Systems*. **33** pp. 8765-8775, 2020.

[14] Li, C., Du, D., Zhang, L., Wen, L., Luo, T., Wu, Y. & Zhu, P. Spatial attention pyramid network for unsupervised domain adaptation. *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*. pp. 481-497, 2020.

[15] Zhou, W., Du, D., Zhang, L., Luo, T. & Wu, Y. Multi-granularity alignment domain adaptation for object detection. *Proceedings Of The IEEE/CVF Conference On Computer Vision And Pattern Recognition*. pp. 9581-9590, 2022.

[16] Cramer, M. The DGPF-test on digital airborne camera evaluation overview and test design. *Photogrammetrie-Fernerkundung-Geoinformation*. pp. 73-82, 2010

[17] Hsu, C., Tsai, Y., Lin, Y. & Yang, M. Every pixel matters: Center-aware feature alignment for domain adaptive object detector. *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*. pp. 733-748, 2020

[18] Wu, Y., Kirillov, A., Massa, F., Lo, W. & Girshick, R. Detectron2. *https://github.com/facebookresearch/detectron2*, 2019

[19] Srishti Srivastava, Sarthak Narayan, and Sparsh Mittal. A survey of deep learning techniques for vehicle detection from uav images. *Journal of Systems Architecture*, page 102152, 2021.

[20] Xingkui Zhu, Shuchang Lyu, Xu Wang, and Qi Zhao. 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*, pages 2778–2788, 2021.

[21] Darius Lam, Richard Kuzma, Kevin McGee, Samuel Dooley, Michael Laielli, Matthew Klaric, Yaroslav Bulatov, and Brendan McCord. xview: Objects in context in overhead imagery. *arXiv preprint arXiv:1802.07856*, 2018.

[22] Jian Ding, Nan Xue, Gui-Song Xia, Xiang Bai, Wen Yang, Michael Yang, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Object detection in aerial images: A large-scale benchmark and challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021.

[23] Pengfei Zhu, Longyin Wen, Xiao Bian, Haibin Ling, and Qinghua Hu. Vision meets drones: A challenge. *arXiv preprint arXiv:1804.07437*, 2018.

[24] Ziwei Deng, Quan Kong, Naoto Akira, and Tomoaki Yoshinaga. Hierarchical contrastive adaptation for cross-domain object detection. *Machine Vision and Applications*, 33(4):1–13, 2022.

[25] Yangguang Zhu, Xian Sun, Wenhui Diao, Hao Li, and Kun Fu. Rfa-net: Reconstructed feature alignment network for domain adaptation object detection in remote sensing imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15:5689–5703, 2022.

[26] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[27] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[28] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[29] Yang Liu, Peng Sun, Nickolas Wergeles, and Yi Shang. A survey and performance evaluation of deep learning methods for small object detection. *Expert Systems with Applications*, 172:114602, 2021.

[30] Du Jiang, Gongfa Li, Chong Tan, Li Huang, Ying Sun, and Jianyi Kong. Semantic segmentation for multiscale target based on object recognition using the improved faster-rcnn model. *Future Generation Computer Systems*, 123:94–104, 2021.

[31] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019.

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[33] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[34] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Probabilistic two-stage detection. *arXiv preprint arXiv:2103.07461*, 2021.

[35] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[36] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[37] Oord, Aaron van den and Li, Yazhe and Vinyals, Oriol Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[38] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[39] Linxiang Zhu, Feifei Lee, Jiawei Cai, Hongliu Yu, and Qiu Chen. An improved feature pyramid network for object detection. *Neurocomputing*, 483:127–139, 2022.

[40] Yang, Chenhongyi and Huang, Zehao and Wang, Naiyan. Querydet: Cascaded sparse query for accelerating high-resolution small object detection In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 13668–13677, 2022

[41] Zhenxing Liu, Xiaoning Song, Zhenhua Feng, Tianyang Xu, Xiaojun Wu, and Josef Kittler. Global context-aware feature extraction and visible feature enhancement for occlusion-invariant pedestrian detection in crowded scenes. *Neural Processing Letters*, pages 1–15, 2022.

[42] Yulin Wu, Ke Zhang, Jingyu Wang, Yezi Wang, Qi Wang, and Xuelong Li. Gcwnet: A global context-weaving network for object detection in remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–12, 2022.

[43] Yiping Gong, Zhifeng Xiao, Xiaowei Tan, Haigang Sui, Chuan Xu, Haiwang Duan, and Deren Li. Context-aware convolutional neural network for object detection in vhr remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 58(1):34–44, 2019.

[44] Jin Zhang, Yanjiao Shi, Qing Zhang, Liu Cui, Ying Chen, and Yugen Yi. Attention guided contextual feature fusion network for salient object detection. *Image and Vision Computing*, 117:104337, 2022.

[45] Jiaxu Leng, Yihui Ren, Wen Jiang, Xiaoding Sun, and Ye Wang. Realize your surroundings: Exploiting context information for small object detection. *Neurocomputing*, 433:287–299, 2021.

[46] Yancheng Bai, Yongqiang Zhang, Mingli Ding, and Bernard Ghanem. Sod-mtgan: Small object detection via multi-task generative adversarial network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 206–221, 2018.

[47] Han-Kai Hsu, Chun-Han Yao, Yi-Hsuan Tsai, Wei-Chih Hung, Hung-Yu Tseng, Maneesh Singh, and Ming-Hsuan Yang. Progressive domain adaptation for object detection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 749–757, 2020.

[48] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6956–6965, 2019.

[49] Jakaria Rabbi, Nilanjan Ray, Matthias Schubert, Subir Chowdhury, and Dennis Chao. Small-object detection in remote sensing images with end-to-end edge-enhanced gan and object detector network. *Remote Sensing*, 12(9):1432, 2020.

[50] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

[51] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (cmd) for domain-invariant representation learning. *arXiv preprint arXiv:1702.08811*, 2017.

[52] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. Pmlr, 2018.

[53] Chaoqi Chen, Weiping Xie, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, Tingyang Xu, and Junzhou Huang. Progressive feature alignment for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 627–636, 2019.

[54] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[55] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[56] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

[57] Guoliang Kang, Lu Jiang, Yunchao Wei, Yi Yang, and Alexander Hauptmann. Contrastive adaptation network for single-and multi-source domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 44(4):1793–1804, 2020.

[58] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. *Advances in Neural Information Processing Systems*, 33:21798–21809, 2020.

[59] Kang Tong, Yiquan Wu, and Fei Zhou. Recent advances in small object detection based on deep learning: A review. *Image and Vision Computing*, 97:103910, 2020.

[60] Debojyoti Biswas and Jelena Tešić. Small object difficulty (sod) modeling for objects detection in satellite images. In *2022 14th International Conference on Computational Intelligence and Communication Networks (CICN)*, pages 1–6. IEEE, 2022.

[61] Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, and I-Hau Yeh. 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*, pages 390–391, 2020.

[62] Peng Sun, Guang Chen, and Yi Shang. Adaptive saliency biased loss for object detection in aerial images. *IEEE Transactions on Geoscience and Remote Sensing*, 58(10):7154–7165, 2020.

[63] Yi Wang, Syed Muhammad Arsalan Bashir, Mahrukh Khan, Qudrat Ullah, Rui Wang, Yilin Song, Zhe Guo, and Yilong Niu. Remote sensing image super-resolution and object detection: Benchmark and state of the art. *Expert Systems with Applications*, page 116793, 2022.

[64] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, 2017.

[65] Jianan Li, Yunchao Wei, Xiaodan Liang, Jian Dong, Tingfa Xu, Jiashi Feng, and Shuicheng Yan. Attentive contexts for object detection. *IEEE Transactions on Multimedia*, 19(5):944–954, 2016.