Clustered Object Detection in Aerial Images

Fan Yang[1] Heng Fan[1] Peng Chu[1] Erik Blasch[2] Haibin Ling[3,1*]
[1]Department of Computer and Information Sciences, Temple University, Philadelphia, USA
[2]Air Force Research Lab, USA
[3]Department Computer Science, Stony Brook University, Stony Brook, NY, USA.
{fyang,hengfan,pchu}@temple.edu, erik.blasch@us.af.mil, hling@cs.stonybrook.edu

Abstract

Detecting objects in aerial images is challenging for at least two reasons: (1) target objects like pedestrians are very small in pixels, making them hardly distinguished from surrounding background; and (2) targets are in general sparsely and non-uniformly distributed, making the detection very inefficient. In this paper, we address both issues inspired by observing that these targets are often clustered. In particular, we propose a Clustered Detection (ClusDet) network that unifies object clustering and detection in an end-to-end framework. The key components in ClusDet include a cluster proposal sub-network (CPNet), a scale estimation sub-network (ScaleNet), and a dedicated detection network (DetecNet). Given an input image, CPNet produces object cluster regions and ScaleNet estimates object scales for these regions. Then, each scale-normalized cluster region is fed into DetecNet for object detection. ClusDet has several advantages over previous solutions: (1) it greatly reduces the number of chips for final object detection and hence achieves high running time efficiency, (2) the cluster-based scale estimation is more accurate than previously used single-object based ones, hence effectively improves the detection for small objects, and (3) the final DetecNet is dedicated for clustered regions and implicitly models the prior context information so as to boost detection accuracy. The proposed method is tested on three popular aerial image datasets including VisDrone, UAVDT and DOTA. In all experiments, ClusDet achieves promising performance in comparison with state-of-the-art detectors.

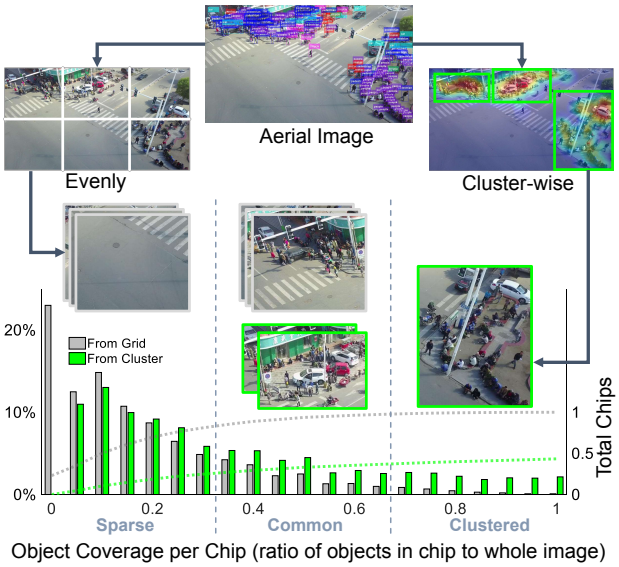


Figure 1: Comparison of grid-based uniform partition and the proposed cluster-based partition. For the narrative purpose, we intentionally classify a chip into three types: *sparse*, *common*, and *clustered*. We observe that, for grid-based uniform partition, more than 73% chips are *sparse* (including 23% chips with *zero* objects), around 25% chips are *common*, and about 2% chips are *clustered*. By contrast, for cluster-based partition, around 50% chips are *sparse*, 35% are *common*, and about 15% belong to *clustered* chips, which is 7× more than that of grid-based partition.

1. Introduction

With the advance of deep neural networks, object detection (*e.g.*, Faster R-CNN [27], YOLO [25], SSD [23]) has witnessed great progress for natural images (*e.g.*, 600×400 images in MS COCO [22]) in recent years. Despite the promising results for general object detection, the performance of these detectors on the aerial images (*e.g.*, 2,000×1,500 pixels in VisDrone [37]) are far from satisfactory in both accuracy and efficiency, which are caused by two challenges: (1) targets typically have small scales relative to the images; and (2) targets are generally sparsely and non-uniformly distributed in the whole image.

Compared with objects in natural images, the scale chal-

*Corresponding author.

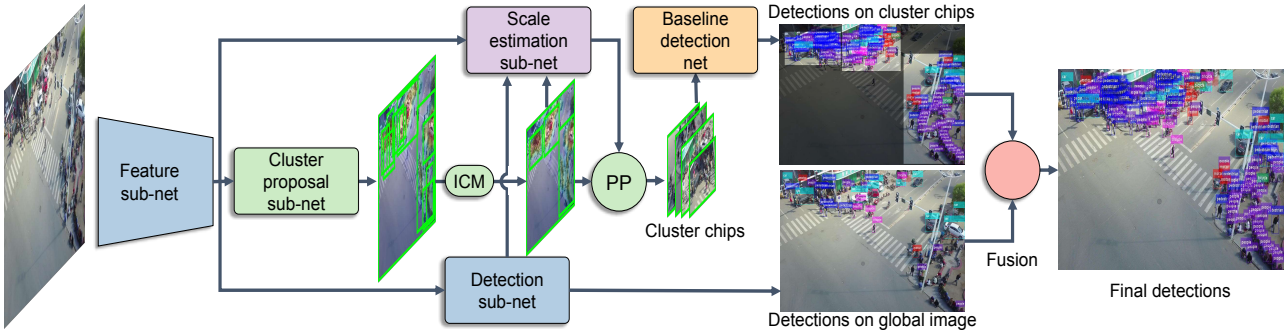


Figure 2: Clustered object Detection (ClusDet) network. The ClusDet network consists of three key components: (1) a cluster proposal subnet (CPNet); (2) a scale estimation subnet (ScaleNet); and (3) a dedicated detection network (DetecNet). CPNet serves to predict the cluster regions. ScaleNet is to estimate the object scale in the clusters. DetecNet performs detection on cluster chips. The final detections are generated by fusing detections from cluster chips and global image. The details of ICM (iterative cluster merging) and PP (partition and padding) are given in Section 3.

lenge causes less effective feature representation of deep networks for objects in aerial images. Therefore, it is difficult for the modern detectors to effectively leverage appearance information to distinguish the objects from surrounding background or similar objects. In order to deal with the scale issue, a natural solution is to partition an aerial image into several uniform small chips, and then perform detection on each of them [10, 24]. Although these approaches alleviate the resolution challenge to some extent, they are *inefficient* in performing detection due to the ignorance of the target sparsity. Consequently, a lot computation resources are inefficiently applied on regions with sparse or even no objects (see Fig. 1). We observe from Fig. 1 that, in an aerial image objects are not only *sparse* and *non-uniform* but also tend to be highly *clustered* in certain regions. For example, pedestrians are usually concentrated in squares and vehicles on highways. Hence, an intuitive way to improve detection efficiency is to focus the detector on these clustered regions where there are a large amount of objects.

Inspired by this motivation, this paper proposes a novel clustered detection (ClusDet) network for addressing both challenges aforementioned by integrating object and cluster detection in a uniform framework. As illustrated in Fig. 2, ClusDet consists of three key components including a cluster proposal sub-network (CPNet), a scale estimation sub-network (ScaleNet) and a baseline detection network (DetecNet). According to the initial detection of an aerial image, CPNet generates a set of regions of object clusters. After obtaining the clustered regions, they are cropped out for subsequent fine detection. To such end, these regions have to be firstly resized to fit the detector, which may result in extremely large or small objects in the clustered regions and thus deteriorate the detection performance [30]. To handle this issue, we present the ScaleNet to estimate an appropriate scale for the objects in each cluster chip and then rescale the chip accordingly before feeding it to a detector, which is different from [10, 24, 18] by directly resizing cropped chips. Afterwards, each clustered chip is fed to the dedicated detector, DetecNet, for fine detection. The final detection is achieved by fusing the detection results on both cluster chips and the global image.

Compared to previous approaches, the proposed ClusDet shows several advantages: (i) Owing to the CPNet, we only need to deal with the clustered regions with plenty of objects, which significantly reduces the computation cost and improves detection efficiency; (ii) With the help of the ScaleNet, each clustered chip is refined for better subsequent fine detection, leading to improvement in accuracy; and (iii) The DetecNet is specially designated for clustered region detection and implicitly models the prior context information to further boost detection accuracy. In extensive experiments on three aerial image datasets, ClusDet achieves the best performance using a single mode while with less computation cost.

In summary, the paper has the following contributions:

1) Proposes a novel ClusDet network to simultaneously address the scale and sparsity challenges for object detection in aerial images.
2) Presents an effective ScaleNet to alleviate nonuniform scale issue in clustered chips for better fine detection.
3) Achieves state-of-the-art performance on three representative aerial image datasets including VisDrone [37], UAVDT [8], DOTA [33] with less computation.

The rest of this paper is organized as follows. Section 2 briefly reviews the related works. In Section 3, we describe the proposed approach in details. Experimental results are shown in Section 4, followed by the conclusion in Section 5.

2. Related work

Object detection has been extensively explored in recent decades with a huge amount of literature. In the following, we first review three lines of works that are the most relevant to ours, and then highlight the differences of CLusDet with existing approaches.

Generic Object Detection. Inspired by the success in image recognition [17], deep convolutional neural networks (CNNs) have been dominated in object detection. According to the detection pipeline, existing detectors can roughly be categorized into two types: region-based detectors and region-free detectors. The region-based detectors separate detection into two steps including proposal extraction and object detection. In the first stage, the search space for detection is significantly reduced through extracting candidate regions (*i.e.*, proposals). In the second stage, these proposals are further classified into specific categories. Representatives of region-based detectors include R-CNN [12], Fast/er R-CNN [11, 27], Mask R-CNN [14] and Cascade R-CNN [3]. On the contrary, the region-free detectors, such as SSD [23] YOLO [25], YOLO9000 [26], RetinaNet [21] and RefineDet [36], perform detection without region proposal, which leads to high efficiency at the sacrifice of accuracy.

Despite excellent performance on natural images (*e.g.*, 500×400 images in PASCAL VOC [9] and 600×400 images in MS COCO [22]), these generic detectors are degenerated when applied on high-resolution aerial images (*e.g.*, 2,000×1,500 images in VisDrone [37], and even larger in UAV captured imagery [19]). Note that detection in high resolution imagery recently has gained an increasing amount of research attention [32].

Aerial Image Detection. Compared to detection in natural images, detection in aerial image is more challenging because (1) objects have small scales relative to the high-resolution aerial images and (2) targets are sparse and nonuniform and concentrated in certain regions. Since this work is focused on deep learning, we only review some relevant works using deep neural networks for aerial image detection. In [28], a simple CNNs based approach is presented for automatic detection in aerial images. The method in [2] integrates detection in aerial images with semantic segmentation to improve performance. In [31], the authors directly extend the Fast/er R-CNN [11, 27] for vehicle detection in aerial images. The work of [6] proposes a coupled region-based CNNs for aerial vehicle detection. The approach of [7] investigates the problem of misalignment between Region of Interests (RoI) and objects in aerial image detection, and introduces a ROI transformer to address this issue. The algorithm in [35] presents a scale adaptive proposal network for object detection in aerial images.

Region Search in Detection. The strategy of region search is commonly adopted in detection to handle small objects. The approach of [24] proposes to adaptively direct computational resources to sub-regions where objects are sparse and small. The work of [1] introduces a context driven search method to efficiently localize the regions containing a specific class of object. In [4], the authors propose to dynamically explore the search space in proposal-based object detection by learning contextual relations. The method in [10] proposes to leverage reinforcement learning to sequentially select regions for detection at higher resolution scale. In a more specific domain, vehicle detection in wide aerial motion imagery (WAMI), the work of [18] suggests a two-stage spatial-temporal convolutional neural networks to detect vehicles from a sequence of WAMI.

Our Approach. In this paper, we aim at solving two aforementioned challenges for aerial image detection. Our approach is related to but different from the previous region search based detectors (*e.g.*, [24, 10]), which partitions high-resolution images into small uniforms chips for detection. In contrast, our solution first predicts cluster regions in the images, and then extract these clustered regions for fine detection, leading to significant reduction of the computation cost. Although the method in [18] also performs detection on chips that potentially contain objects, our approach significantly differs from it. In [18], the obtained chips are directly resized to fit the detector for subsequent detection. On the contrary, inspired by the observation in [30] that objects with extreme scales may deteriorate the detection performance, we propose a ScaleNet to alleviate this issue, resulting in improvement in fine detection on each chip.

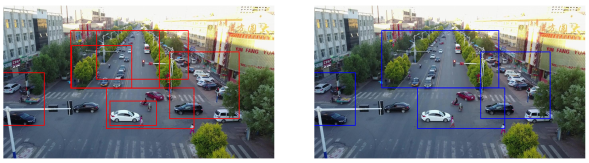
3. Clustered Detection (ClusDet) Network

3.1. Overview

As shown in Fig. 2, detection of an aerial image consists of three stages: cluster region extraction, fine detection on cluster chips and fusion of detection results. In specific, after the feature extraction of an aerial image, CPNet takes as input the feature maps and outputs the clustered regions. In order to avoid processing too many cluster chips, we propose an iterative cluster merging (ICM) module to reduce the noisy cluster chips. Afterwards, the cluster chips as well as the initial detection results on global image are fed into the ScaleNet to estimate an appropriate scale for the objects in cluster chips. With the scale information, the cluster chips are rescaled for fine detection with DetecNet. The final detection is obtained by fusing the detection results of each cluster chip and global image with standard non-maximum suppression (NMS).

3.2. Cluster Region Extraction

Cluster region extraction consists of two steps: initial cluster generation using cluster proposal sub-network (CPNet) and cluster reduction with iterative cluster merging (ICM).



(a) cluster detections (b) cluster detections + ICM

Figure 3: Illustration of merging of cluster detections. The red boxes are the cluster detections from CPNet. The blue boxes represent clusters after iterative cluster merge (ICM).

3.2.1 Cluster Proposal Sub-network (CPNet)

The core of the cluster region extraction is the cluster proposal sub-network (CPNet). CPNet works on the high-level feature maps of an aerial image, and aims at predicting the locations and scales of clusters[1]. Motivated by the region proposal networks (RPN) [27], we formulate CPNet as a block of fully convolutional networks. In specific, CPNet takes as input the high-level feature maps from feature extraction backbone, and utilizes two subnets for regression and classification, respectively. Although our CPNet shares the similar idea with RPN, they are different. RPN is used to propose the candidate regions of objects, while CPNet aims at proposing the candidate regions of clusters. Compared to the object proposal, the size of cluster is much larger, and thus CPNet needs a larger receptive field than that of RPN. For this reason, we attach CPNet on the top of the feature extraction backbone.

It is worth noting that the learning of CPNet is a supervised process. However, none of existing public datasets provide groundtruth for clusters. In this work, we adopt a simple strategy to generate the required groundtruth of clusters for training CPNet. We refer the readers to supplementary material for details in generating cluster groundtruth.

3.2.2 Iterative Cluster Merging (ICM)

As shown in Fig. 3 (a), we observe that the initial clusters produced by CPNet are dense and messy. These dense and messy cluster regions are difficult to be directly leveraged for fine detection because of their high overlap and large size, resulting in extremely heavy computation burden in practice. To solve this problem, we present a simple yet effective iterative cluster merging (ICM) module to clean up clusters. Let $\mathcal{B} = \{B_i\}_{i=1}^{N_\mathcal{B}}$ represent the set of $N_\mathcal{B}$ cluster bounding boxes detected by CPNet, and $\mathcal{R} = \{R_i\}_{i=1}^{N_\mathcal{B}}$ denote the corresponding cluster classification scores. With a pre-defined overlap threshold τ_{op} and maximum number $N_{\max}$ of clusters after merging, we can obtain the merged cluster set $\mathcal{B}' = \{B'_i\}_{i=1}^{N_{\mathcal{B}'}}$ with $N'_\mathcal{B}$ clusters with Alg. 1.

Algorithm 1: Iterative Cluster Merging (ICM)

Input: Initial cluster bounding boxes $\mathcal{B} = \{B_i\}_{i=1}^{N_\mathcal{B}}$, initial cluster scores $\mathcal{R} = \{R_i\}_{i=1}^{N_\mathcal{B}}$, threshold τ_{op} and maximum number of merged clusters $N_{\max}$;
Output: Merged clusters $\mathcal{B}' = \{B'_i\}_{i=1}^{N_{\mathcal{B}'}}$;
begin
  $\mathcal{B}' \leftarrow \mathcal{B}$;
  while $|\mathcal{B}'| > N_{\max}$ **do**
    $\mathcal{B}', \mathcal{R}' \leftarrow \text{NMM}(\mathcal{B}, \mathcal{R}, \tau_{\text{op}})$
    if $|N'_\mathcal{B}| = |N_\mathcal{B}|$ **then**
      break;
    else
      $\mathcal{B} \leftarrow \mathcal{B}'; \mathcal{R} \leftarrow \mathcal{R}'$;
    end
  end
  $\mathcal{B}'' \leftarrow \{\}$;
  for $i \leq \min(N_{\max}, |\mathcal{B}'|)$ **do**
    $\mathcal{B}'' \leftarrow \mathcal{B}'' \cup \{B'_i\}$;
  end
  $\mathcal{B}' \leftarrow \mathcal{B}''$;
end

Briefly speaking, we first find the B_i with highest score, then select the clusters whose overlaps with B_i are larger than the threshold τ_{op} to merge with B_i. All the merged clusters are removed. Afterwards, we repeat the aforementioned process until $\mathcal{B}$ is empty. All the processes mentioned above correspond to the non-max merging (NMM) in Alg. 1. We conduct the NMM several times until the preset $N_{\max}$ is reached. For the details of the NMM, the readers are referred to supplementary material. Fig. 3 (b) demonstrates the final merged clusters, showing that the proposed ICM module is able to effectively merge the dense and messy clusters.

3.3. Fine Detection on Cluster Chip

After obtaining the cluster chips, a dedicated detector is utilized to perform fine detection on these chips. Unlike in existing approaches [24, 18, 10] that directly resize these chips for detection, we present a scale estimation sub-network (ScaleNet) to estimate the scales of objects in chips, which avoids extreme scales of objects degrading detection performance. Based on the estimated scales, ClusDet performs partition and padding (PP) operations on each chip for detection.

3.3.1 Scale Estimation Sub-network (ScaleNet)

We regard scale estimation as a regression problem and formulate ScaleNet using a bunch of fully connected networks.

[1] In this work, a cluster in aerial images is defined by a rectangle region containing at least three objects.

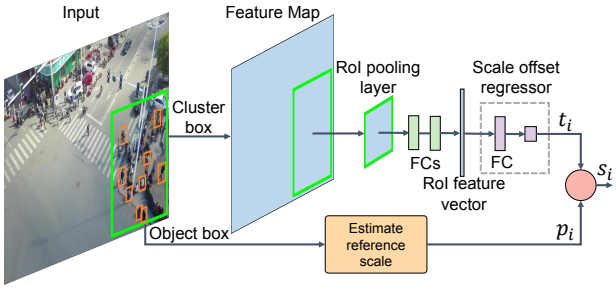


Figure 4: The architecture of the scale estimation network (ScaleNet). The cluster detections are projected to feature map space. Each cluster is pooled into a fixed-size feature map and mapped into a feature vector by fully connected layers (FCs). The network has an output per cluster, *i.e.*, the scale regression offset.

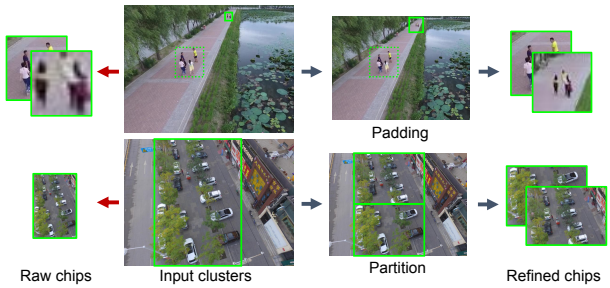


Figure 5: Illustration of the partition and padding (PP) process. The raw chips and refined chips are the input of detector without and with using PP, respectively.

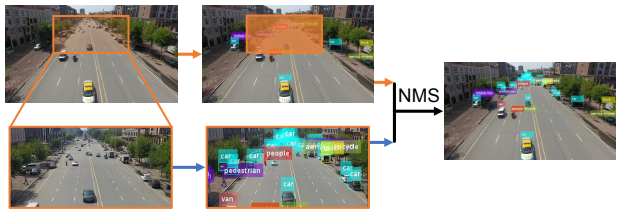


Figure 6: The illustration of fusing detections from whole images and cluster chips.The object detections in orange region from whole image are eliminated when applying fusion operation.

As shown in Fig. 4, ScaleNet receives three inputs including the feature maps extracted from network backbone, cluster bounding boxes and initial detection results on global image, and outputs a relative scale offset for objects in the cluster chip. Here, the initial detection results are obtained from the detection subnet.

Let $t_i^* = (p_i - s_i^*)/p_i$ be the relative scale offset for cluster i, where p_i and s_i^* represent the reference scale of the detected objects and the average scale of the groundtruth boxes in cluster i, respectively. Thus, the loss of the ScaleNet can be mathematically defined as

$$\mathcal{L}(\{t_i\}) = \frac{1}{M}\sum_{i}^{M} \ell_{\text{reg}}(t_i, t_i^*) \tag{1}$$

where $t_i = (p_i - s_i)/p_i$ is the estimated relative scale offset, s_i is the estimated scale, and M is the number of cluster boxes. The ℓ_{reg} is a smoothly ℓ_1 loss function [11].

3.3.2 Partition and Padding (PP)

The partition and padding (PP) operations are utilized to ensure that the scales of objects are within a reasonable range. Given the cluster bounding box B_i, the corresponding estimated object scale S_i and the input size S_{in} of a detector, we can estimate the object scale in the input space of the detector $S_i^{in} = S_i \times \frac{S_{in}}{S_i}$. If the scale S_i^{in} is larger than a certain range, the cluster is padded proportionally, otherwise it is partitioned into two equal chips. Note that detections in the padded region are ignored in final detection. The visualization of the process is in Fig. 5. The specific scale range setting is discussed in Section 4.

After rescaling the cluster chip, a dedicated baseline detection network (DetecNet) performs fine object detection. The architecture of the DetecNet can be any state-of-the-art detectors. The backbone of the detector can be any standard backbone networks, e.g., VGG [29], ResNet [15], ResNeXt [34].

3.4. Final Detection with Local-Global Fusion

The final detection of an aerial image is obtained by fusing the local detection results of cluster chips and global detection results of the whole image with the standard NMS post-processing (see Fig. 6). The local detection results are obtained through the proposed approach mentioned above, and the global detection results are derived from detection subnet (Fig. 2). It is worth noting that any existing modern detectors can be used for global detection.

4. Experiments

4.1. Implementation Details

We implement ClusDet based on the publicly available Detectron [13] and Caffe2. The Faster R-CNN (FR-CNN) [27] with Feature Pyramid Network (FPN) [20] are adopted as the baseline detection network (DetecNet). The architecture of the CPNet is implemented with a 5×5 convolutional layer followed by two sibling 1×1 convolutional layers (for regression and classification, respectively). In ScaleNet, the FC layers to convert feature map into feature vector are with size of 1024; The size of FC layers in the scale offset regressor are 1024 and 1 respectively. The IoU

Table 1: The ablation study on VisDrone dataset. The 'c' denotes EIP cropped images. The 'ca' indicates cluster-aware cropped images. The 'o' indicates the original validation data. The #img is the number of images forwarded to detector. The 's', 'm', and 'l' represent small, medium, and large, respectively. The inference time is measured on a GTX 1080 Ti.

Methods	backbone	test data	#img	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	s/img (GPU)
FRCNN[27]+FPN[20]	ResNet50	o	548	21.4	40.7	19.9	11.7	33.9	54.7	0.055
FRCNN[27]+FPN[20]	ResNet101	o	548	21.4	40.7	20.3	11.6	33.9	54.9	0.074
FRCNN[27]+FPN[20]	ResNeXt101	o	548	21.8	41.8	20.1	11.9	34.8	55.5	0.156
FRCNN[27]+FPN[20]+EIP	ResNet50	c	3,288	21.1	44.0	18.1	14.4	30.9	30.0	0.330
FRCNN[27]+FPN[20]+EIP	ResNet101	c	3,288	23.5	46.1	21.1	17.1	33.9	29.1	0.444
FRCNN[27]+FPN[20]+EIP	ResNeXt101	c	3,288	24.4	47.8	21.8	17.8	34.8	34.3	0.936
DetecNet+CPNet	ResNet50	o+ca	1,945	25.6	47.9	24.3	16.2	38.4	53.7	0.195
DetecNet+CPNet	ResNet101	o+ca	1,945	25.3	47.4	23.8	15.6	38.1	**54.6**	0.262
DetecNet+CPNet	ResNeXt101	o+ca	1,945	27.6	51.2	26.2	17.5	**41.0**	54.2	0.554
DetecNet+CPNet+ScaleNet	ResNet50	o+ca	2,716	26.7	50.6	24.7	17.6	38.9	51.4	0.273
DetecNet+CPNet+ScaleNet	ResNet101	o+ca	2,716	26.7	50.4	25.2	17.2	39.3	54.9	0.366
DetecNet+CPNet+ScaleNet	ResNeXt101	o+ca	2,716	**28.4**	**53.2**	**26.4**	**19.1**	40.8	54.4	0.773

threshold for merging clusters in NMM process is set to 0.7. Following the definition in the COCO[22] dataset, the object scale range in cluster chip partition and padding is set to $[70, 280]$ pixels.

Training phase. The input size of the detector is set to $600 \times 1,000$ pixels on the VisDrone [37] and UAVDT [8] datasets and $1,000 \times 1,000$ pixels on the DOTA [33] dataset. On the three datasets, the training data is augmented by dividing images into chips. On the VisDrone [37] and UAVDT [8] datasets, each image is uniformly divided into 6 and 4 chips without overlap. The reason of setting a specific number of chips is that the size of cropped chip can be similar with that in COCO [22] dataset. On the DOTA [33] dataset, we use the tool provided by the authors to divide the images. When training model on the VisDrone [37] and UAVDT [8] datasets by using 2 GPUs, we set the base learning rate to 0.005 and total iteration to 140k. After the first 120k iterations, the learning rate decreases to 0.0005. Then, we train the model for 100k iterations before lowering the learning rate to 0.00005. A momentum of 0.9 and parameter decay of 0.0005 (on weights and biases) are used. On the DOTA [33] dataset, the base learning and the total iterations are set to 0.005 and 40k, respectively. The learning rate is deceased by a factor of 0.1 after 30k and 35k iterations.

Test phase. The input size of detector is the same with that in training phase whenever not specified. The maximum number of clusters (TopN) in cluster chip generation is empirically set to 3 on VisDrone [37], 2 on UAVDT [8], and 5 on the DOTA [33]. In fusing detection, the threshold of the standard non-max suppression (NMS) is set to 0.5 in all datasets. The final detection number is set to 500.

4.2. Datasets

To validate the effectiveness of the proposed method, we conduct extensive experiments on three publicly accessible

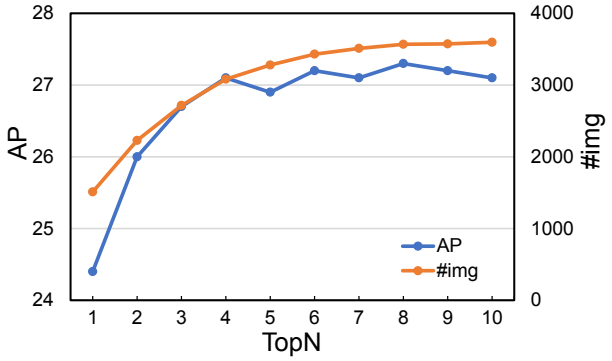


Figure 7: The AP and number of forwarded images over different settings of TopN in ClusDet.

datasets: VisDrone [37], UAVDT [8], and DOTA [33].

VisDrone. The dataset consists of 10, 209 images (6,471 for training, 548 for validation, 3,190 for testing) with rich annotations on ten categories of objects. The image scale of the dataset is about $2,000 \times 1,500$ pixels. Since the evaluation server is closed now, we cannot test our method on the test dataset. Therefore, the validation dataset is used as test dataset to evaluate our method.

UAVDT. The UAVDT [8]] dataset contains 23,258 images of training data and 15,069 images of test data. The resolution of the image is about $1,080 \times 540$ pixels. The dataset is acquired with an UAV platform at a number of locations in urban areas. The categories of the annotated objects are car, bus, and truck.

DOTA. The dataset is collected from multiple sensors and platforms (e.g. Google Earth) with multiple resolutions (800×800 through 4,000×4,000 pixels) at multiple cities. Fifteen categories are chosen and annotated. Considering that ClusDet is based on the cluster characteristic of the objects in aerial image, some categories in the dataset are not

Table 2: The detection performance on VisDrone validation dataset. The ⋆ denotes the multi-scale inference and bounding box voting are utilized in test phase.

Methods	backbone	AP	AP_{50}	AP_{75}
RetinaNet[21]+FPN[20]	ResNet50	13.9	23.0	14.9
RetinaNet[21]+FPN[20]	ResNet101	14.1	23.4	14.9
RetinaNet[21]+FPN[20]	ResNeXt101	14.4	24.1	15.5
FRCNN[27]+FPN[20]	ResNet50	21.4	40.7	19.9
FRCNN[27]+FPN[20]	ResNet101	21.4	40.7	20.3
FRCNN[27]+FPN[20]	ResNeXt101	21.8	41.8	20.1
FRCNN[27]+FPN[20] ⋆	ResNeXt101	28.7	51.8	27.7
FRCNN[27]+FPN[20]+EIP	ResNet50	21.1	44.0	18.1
FRCNN[27]+FPN[20]+EIP	ResNet101	23.5	46.1	21.1
FRCNN[27]+FPN[20]+EIP	ResNeXt101	24.4	47.8	21.8
FRCNN[27]+FPN[20]+EIP⋆	ResNeXt101	25.7	48.4	24.1
ClusDet	ResNet50	26.7	50.6	24.7
ClusDet	ResNet101	26.7	50.4	25.2
ClusDet	ResNeXt101	28.4	53.2	26.4
ClusDet ⋆	ResNeXt101	**32.4**	**56.2**	**31.6**

suitable for ClusDet, e.g., roundabout, bridge. Thus, we only choose the images with movable objects in the dataset to evaluate our method, i.e., plane, ship, large vehicle, small vehicle, and helicopter, Thus, the training and validation data contain 920 images and 285 images, respectively.

4.3. Compared Methods

We compare our ClusDet with evenly image partition (EIP) method on all datasets. On some datasets if the EIP is not provided, we implement it according to the property of the datasets. In addition, we also compare our method with representative state-of-the-art methods on all datasets.

4.4. Evaluation Metric

Following the evaluation protocol on the COCO [22] dataset, we use AP, AP_{50}, and AP_{75} as the metrics to measure the precision. Specifically, AP is computed by averaging over all categories. AP_{50} and AP_{75} are computed at the single IoU threshold 0.5 and 0.75 over all categories. The efficiency is measured by the number of images needed to be processed by the detector and the average time to process a global image and its chips in inference stage. In specific, the number of images refer to the summation of global images and cropped chips. In the subsequent experiments, the number of images is denoted as $\#img$.

4.5. Ablation Study

To validate the contributions of the cluster detection and scale estimation to detection improvement, we conduct extensive experiments on VisDrone [37].

In the following experiments, the input size of detector in the test phase is set to $600 \times 1,000$ pixels. To validate if the proposed method can gain consistent improvement in performance under different backbone networks, we conduct experiments with three backbone networks: ResNet-50 [15], ResNet-101 [15], and ResNeXt-101 [34].

Effect of EIP. The experimental results are listed in Table 1. We note that FRCNN [27] performs inferior compared to that in COCO [22] (AP=36.7). This is because the relative scale of object to image in VisDrone [37] is much smaller than that in COCO [22]. By applying EIP to the image, the performance of detectors are increased significantly, especially on small objects (AP_s). However, the number of images needed to be processed increases 6 times (3,288 vs 548). In addition, we note that although the overall performance AP is improved by applying EIP, the performance of large scale objects (AP_l) is decreased. This is because the EIP truncates the large objects into pieces, which results in many false positives.

Effect of Cluster Detection. From Table 1, we note that the DetecNet+CPNet processes much less amount of images (1,945 vs 3,288) but achieves better performance than FRCNN [27] plus EIP. This demonstrates that the CPNet not only selects the clustered regions to save computation resource but also implicitly encodes the prior context information to improve the performance. In addition, we note that compared to EIP, the CPNet dose not reduce the performance of large objects (AP_l), this can be attributed to the CPNet, which introduces the spatial distribution information of the object into the ClusDet network so as to avoid truncating the large object.

Effect of Scale Estimation. After integrating ScaleNet into CPNet and DetecNet, we note that the number of processed image increases to 2,716, this is because the PP module partitions some cluster chips into pieces. This mitigates the small scale problem when performing detection, such that the performance (AP) is improved to 26.7 on ResNet50 [15] backbone network. In addition, we see that the ScaleNet improves the detection performance on all types of backbone networks. Particularly, the metric AP_{50} is boosted by 2-3 points. In addition, the AP_s is increased by 1.6 points even on very strong backbone, ResNeXt101 [15]. This demonstrate that the ScaleNet does alleviate the scale problem to certain extent.

The Effect of Hyperparameter TopN. To fairly investigate the effect of TopN, we only change the setting in test phase, which avoids the influence by the amount of training data. From Fig. 7, we see that after $TopN = 4$, the number of processed images gradually increases, yet the AP dose not change too much and just fluctuates around $AP = 27$. This means that a lot of cluster regions are repetitively computed when TopN is set to a high value. This observation also indicates that the cluster merge operation is critical to decrease the computation cost.

Table 3: The detection performance of the baselines and proposed method on the UAVDT [8] dataset.

Methods	backbone	#img	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
R-FCN[5]	ResNet50	15,069	7.0	17.5	3.9	4.4	14.7	12.1
SSD[23]	N/A	15,069	9.3	21.4	6.7	7.1	17.1	12.0
RON[16]	N/A	15,069	5.0	15.9	1.7	2.9	12.7	11.2
FRCNN[27]	VGG	15,069	5.8	17.4	2.5	3.8	12.3	9.4
FRCNN[27]+FPN[20]	ResNet50	15,069	11.0	23.4	8.4	8.1	20.2	26.5
FRCNN[27]+FPN[20]+EIP	ResNet50	60,276	6.6	16.8	3.4	5.2	13.0	17.2
ClusDet	ResNet50	25,427	**13.7**	**26.5**	**12.5**	**9.1**	**25.1**	**31.2**

Table 4: The detection performance of the baselines and proposed method on DOTA [33] dataset.

Methods	backbone	#img	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
RetinaNet[21]+FPN[20]+EIP	ResNet50	2,838	24.9	41.5	27.4	9.9	32.7	30.1
RetinaNet[21]+FPN[20]+EIP	ResNet101	2,838	27.1	44.4	30.1	10.6	34.8	33.7
RetinaNet[21]+FPN[20]+EIP	ResNeXt101	2,838	27.4	44.7	29.8	10.5	35.8	32.8
FRCNN[27]+FPN[20]+EIP	ResNet50	2,838	31.0	50.7	32.9	16.2	37.9	37.2
FRCNN[27]+FPN[20]+EIP	ResNet101	2,838	31.5	50.4	36.6	16.0	38.5	38.1
ClusDet	ResNet50	1,055	**32.2**	47.6	**39.2**	**16.6**	32.0	**50.0**
ClusDet	ResNet101	1,055	31.6	47.8	38.2	15.9	31.7	49.3
ClusDet	ResNeXt101	1,055	31.4	47.1	37.4	17.3	32.0	45.4

4.6. Quantitative Results

VisDrone The detection performance of the proposed method and representative detectors, i.e., Faster RCNN [27] and RetinaNet [21], is shown in Table 2. We note that our method outperforms the state-of-the-art methods by a large margin over various backbone settings. Besides, we observe that when testing the model using multi-scale setting (denoted by $\star$), the performance is significantly boosted, except for the methods using EIP. This is because in multi-scale test, the cropped chips are resized to extremely large scale such that detectors output many false positives on background or local regions of objects.

UAVDT The experimental results on the UAVDT [8] dataset are displayed in Table 3. The performance of the compared methods, except for FRCNN [27]+FPN [20], is computed using the experimental results provided in [8]. From the Table 3, we observe that applying EIP on test data dose not improve the performance. On the contrary, it dramatically decreases the performance (11.0 vs 6.1). The reason of this phenomenon is that the objects, i.e. vehicles, in the UAVDT always appear in the center of the image, while the EIP operation divides the objects into pieces such that the detector cannot correctly estimate the objects scale. Compared to FRCNN [27]+FPN [20] (FFPN), our ClusDet is superior to the FFPN and FFPN+EIP. The performance improvement mainly benefits from the different image crop operation. In our method, the image is cropped based on the clusters information, which is less likely to truncate numerous objects. The performance of detectors on UAVDT [8] is much lower than that on VisDrone [38], which is caused by the extremely unbalanced data.

DOTA On the DOTA[33] dataset, our ClusDet achieves similar performance with state-of-the-art methods but processes dramatically less image chips. This is because the CPNet significantly reduces the number of chips for fine detection. Although our method does not outperform the state-of-the-art methods in term of the overall performance at low IoU (AP_{50}), it obtains higher AP_{75} value, which indicates that our method can more precisely estimate the object scale. Besides, we observe that the performance does not change too much when more complex backbone networks are adopted. This can be attributed to the limited training images. Without a large amount of data, the complex model cannot achieve its superiority.

5. Conclusion

We present a Clustered object Detection (ClusDet) network to unify object clustering and detection in an end-to-end framework. We show that ClusDet can successfully predict the clustered regions in images to significantly reduce the number of chips for detection so as to improve the efficiency. Moreover, we propose a cluster-based object scale estimation network to effectively detect the small object. In addition, we experimentally demonstrate that the proposed ClusDet network implicitly models the prior context information to improve the detection precision. By extensive experiments, we show that our method obtains state-of-the-art performance on three public datasets.

Acknowledgement. We sincerely appreciate anonymous reviewers for their helpful comments in improving the draft. This work is supported in part by US NSF Grants 1814745, 1407156 and 1350521.

References

[1] B. Alexe, N. Heess, Y. W. Teh, and V. Ferrari. Searching for objects driven by context. In *NIPS*. 2012. 3

[2] N. Audebert, B. Le Saux, and S. Lefèvre. Segment-before-detect: Vehicle detection and classification through semantic segmentation of aerial images. *Remote Sensing*, 9(4):368, 2017. 3

[3] Z. Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018. 3

[4] X. S. Chen, H. He, and L. S. Davis. Object detection in 20 questions. In *WACV*, 2016. 3

[5] J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In *NIPS*, 2016. 8

[6] Z. Deng, H. Sun, S. Zhou, J. Zhao, and H. Zou. Toward fast and accurate vehicle detection in aerial images using coupled region-based convolutional neural networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(8):3652–3664, 2017. 3

[7] J. Ding, N. Xue, Y. Long, G.-S. Xia, and Q. Lu. Learning roi transformer for detecting oriented objects in aerial images. In *CVPR*, 2019. 3

[8] D. Du, Y. Qi, H. Yu, Y. Yang, K. Duan, G. Li, W. Zhang, Q. Huang, and Q. Tian. The unmanned aerial vehicle benchmark: object detection and tracking. In *ECCV*, 2018. 2, 6, 8

[9] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, 2015. 3

[10] M. Gao, R. Yu, A. Li, V. I. Morariu, and L. S. Davis. Dynamic zoom-in network for fast object detection in large images. In *CVPR*, 2018. 2, 3, 4

[11] R. Girshick. Fast r-cnn. In *ICCV*, 2015. 3, 5

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 3

[13] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He. Detectron. https://github.com/facebookresearch/detectron, 2018. 5

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, 2017. 3

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5, 7

[16] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, and Y. Chen. Ron: Reverse connection with objectness prior networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5936–5944, 2017. 8

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 3

[18] R. LaLonde, D. Zhang, and M. Shah. Clusternet: Detecting small objects in large scenes by exploiting spatio-temporal information. In *CVPR*, 2018. 2, 3, 4

[19] P. Liang, G. Teodoro, H. Ling, E. Blasch, G. Chen, and L. Bai. Multiple kernel learning for vehicle detection in wide area motion imagery. In *Proc. of the International Conference on Information Fusion (FUSION)*, 2012. 3

[20] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 5, 6, 7, 8

[21] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 3, 7, 8

[22] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1, 3, 6, 7

[23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 1, 3, 8

[24] Y. Lu, T. Javidi, and S. Lazebnik. Adaptive object detection using adjacency and zoom prediction. In *CVPR*, 2016. 2, 3, 4

[25] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 1, 3

[26] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. In *CVPR*, 2017. 3

[27] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 1, 3, 4, 5, 6, 7, 8

[28] I. Ševo and A. Avramović. Convolutional neural network based automatic object detection on aerial images. *GRSL*, 13(5):740–744, 2016. 3

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 5

[30] B. Singh and L. S. Davis. An analysis of scale invariance in object detection snip. In *CVPR*, 2018. 2, 3

[31] L. W. Sommer, T. Schuchert, and J. Beyerer. Fast deep vehicle detection in aerial images. In *WACV*, 2017. 3

[32] K. Sun, Y. Zhao, B. Jiang, T. Cheng, B. Xiao, D. Liu, Y. Mu, X. Wang, W. Liu, and J. Wang. High-resolution representations for labeling pixels and regions. *CoRR*, abs/1904.04514, 2019. 3

[33] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang. Dota: A large-scale dataset for object detection in aerial images. In *CVPR*, 2018. 2, 6, 8

[34] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. 5, 7

[35] S. Zhang, G. He, H.-B. Chen, N. Jing, and Q. Wang. Scale adaptive proposal network for object detection in remote sensing images. *GRSL*, 2019. 3

[36] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li. Single-shot refinement neural network for object detection. In *CVPR*, 2018. 3

[37] P. Zhu, L. Wen, X. Bian, H. Ling, and Q. Hu. Vision meets drones: a challenge. *arXiv:1804.07437*, 2018. 1, 2, 3, 6, 7

[38] P. Zhu et al. Visdrone-det2018: The vision meets drone object detection in image challenge results. In *ECCVW*, 2018. 8