Density Map Guided Object Detection in Aerial Images

Changlin Li[1], Taojiannan Yang[1], Sijie Zhu[1], Chen Chen[1], Shanyue Guan[2]
[1]University of North Carolina at Charlotte [2]East Carolina University
{cli33, tyang30, szhu3, chen.chen}@uncc.edu, guans18@ecu.edu

Abstract

Object detection in high-resolution aerial images is a challenging task because of 1) the large variation in object size, and 2) non-uniform distribution of objects. A common solution is to divide the large aerial image into small (uniform) crops and then apply object detection on each small crop. In this paper, we investigate the image cropping strategy to address these challenges. Specifically, we propose a Density-Map guided object detection Network (DMNet), which is inspired from the observation that the object density map of an image presents how objects distribute in terms of the pixel intensity of the map. As pixel intensity varies, it is able to tell whether a region has objects or not, which in turn provides guidance for cropping images statistically. DMNet has three key components: a density map generation module, an image cropping module and an object detector. DMNet generates a density map and learns scale information based on density intensities to form cropping regions. Extensive experiments show that DMNet achieves state-of-the-art performance on two popular aerial image datasets, i.e. *VisionDrone [30] and UAVDT [4].*

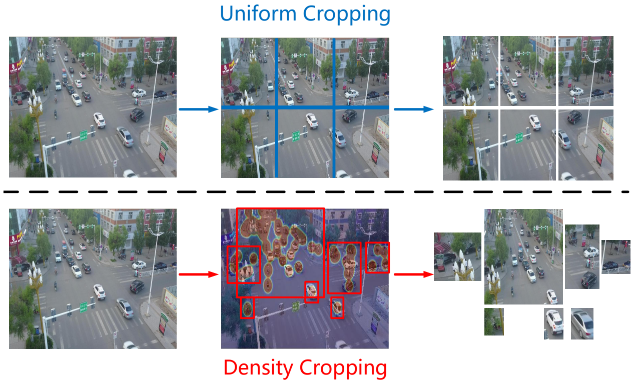


Figure 1: Visualization of density cropping vs. uniform cropping. Top row provides an example of uniform cropping. Bottom row gives a comparable example of density cropping. Uniform crops have more background pixels and fail to accommodate the bounding box resolution of different categories compared with density crops. The first column shows the input aerial image. The second column shows the proposal regions for cropping. The third column shows the cropping results. Blue and red rectangles indicate candidate regions for cropping.

1. Introduction

Object detection is a fundamental problem in computer vision, which is critical for surveillance applications, *e.g.*, face detection and pedestrian detection. Deep learning based architectures have now become the standard pipelines for general object detection (*e.g.*, Faster RCNN [17], RetinaNet [11], SSD [14]). Although these methods achieve good performance on natural image datasets (*e.g.*, MS COCO dataset [13] and Pascal VOC [5] dataset), they are not able to generate satisfactory results on specialized images, *e.g.*, aerial and medical images.

Due to the special view point and large field of view, aerial image has become an important source for practical applications, *e.g.*, surveillance. Aerial images are usually collected by drones, airplane or satellite from top view [23], therefore their visual appearance can be significantly different from natural images like ImageNet [18]. These characteristics give rise to several special challenges for aerial image object detection: (1) Due to variation of the photoing angle, object scale variance exists in aerial image dataset. (2) The number of objects is highly imbalanced across different categories in most of the cases. (3) Occlusion (between objects) and truncation (objects appear on the boundary) are common in aerial images. (4) Small objects account for a larger percentage compared with natural image datasets.

Early works [9, 2] on aerial image object detection simply leverage the general object detection architecture and focus on improving the detection of small objects. [9] introduces the upsampling module after feature extraction to increase spatial resolution. [2] generates fine-grained feature representations to help map small objects to its larger correspondences. The improved small object detection may achieve reasonable results on popular datasets [30, 31, 23], they are far from satisfactory for practical applications.

To address the scale variation problem, another promising direction is to crop the original image into small crops/chips before applying the object detection, such as uniform cropping [15] and random cropping. For most of the cases, these simple cropping strategies help improve the detection accuracy of small objects, since the resolutions of small crops become higher when they are resized to the size of the original image. However, they are not able to leverage the semantic information for cropping, thus resulting in a majority of crops with only background. In addition, large objects may be cut into two or more different crops by these strategies.

Following the idea of image cropping, how to find reasonable crops turn out to be critical for aerial image object detection. Apparently, cropping based on the distribution of objects would generate better crops than uniform or random strategy. And how to generate the distribution of objects has been studied in a similar task [24], crowd counting, which shares the same challenge of scale and viewpoint variation. In dense crowd scenes, bounding box based detection may not be applicable for small objects. Recent state-of-the-art methods leverage the power of density map for estimating the distribution of people in the scene, and achieve promising performance. This inspires us to explore the power of object density map in generating crops for aerial image object detection.

In this paper, we propose a density map based aerial image detection framework – DMNet. It utilizes object density map to indicate the presence of objects as well as the object density within a region. The distribution of objects enables our cropping module to generate better image crops for further object detection as shown in Fig. 1. For example, a proper density threshold can filter out most of the background area and reduce the number of objects in each crop, which makes it possible to recognize extremely small objects by upsampling the image crops.

Fig. 2 shows the framework of the proposed DMNet. First, we introduce a density map generation network to generate the density map for each aerial image. Second, we assign a window with average object scale and slide the window over the density map without overlapping. The density map intensity indicates the probability of object presence in one position. Therefore, at each window position, the sum of all (density) pixel intensities within the window is computed, which can be considered as the likelihood of objects in this window. Then, a density threshold is applied to filter out windows with low overall intensity values. That is we assign “0” to the window whose intensity sum value is below the threshold (*i.e.*, the pixels in this window all have 0 value), and “1” to the opposite. Third, we merge the candidate windows assigned with “1” into regions via connected component to generate image crops. Variations of pixel intensity in different regions implicitly provide the context information (*e.g.*, background between neighboring objects) to generate valid crops accordingly. Finally, we use the cropped images to train the object detector.

Compared with existing approaches, DMNet has the following advantages: (1) It offers a simple design to crop image based on the distribution of objects with the help of object density map. (2) It is able to alleviate object truncation and preserve more contextual information than the uniform cropping strategy. (3) Compared with [26], which also develops a non-uniform cropping scheme, DMNet only needs to train a simple density generation network instead of training two sub-networks (*i.e.* a cluster proposal sub-network (CPNet) and a scale estimation sub-network (ScaleNet)).

In summary, the paper has the following contributions.

- We are the first to introduce density map into aerial image object detection, where density map based cropping method is proposed to utilize spatial and context information between objects for improved detection performance.
- We propose an effective algorithm to generate image crops without the need of training additional deep neural networks, as an alternative to [26].
- Extensive experiments suggest that the proposed method achieves the state-of-the-art performance on representative aerial image datasets, including VisionDrone [30] and UAVDT [4].

The rest of the paper is organized as follows. Section 2 discusses related work for object detection. Section 3 presents the methodology in detail. Section 4 provides experimental results on two datasets and extensive ablation studies. Finally, Section 5 concludes the paper.

2. Related work

2.1. General object detection

General object detection targets primarily on natural images. Proposal-based detectors introduce the concept of anchors with multiple stages. Fast R-CNN [6] generates proposals using selective search and then extracts features and classifies objects accordingly based on those proposals. Faster R-CNN [17] generates proposals by the region proposal network (RPN) which significantly accelerates the inference speed. Mask R-CNN [7] extends Faster R-CNN to perform detection and instance segmentation tasks simultaneously. On the other hand, YOLO3 [16], SSD [14] and RetinaNet [11] are examples of single stage detectors. Single stage detectors skip proposal stage and detect directly on sampling regions. They improve detection speed at the

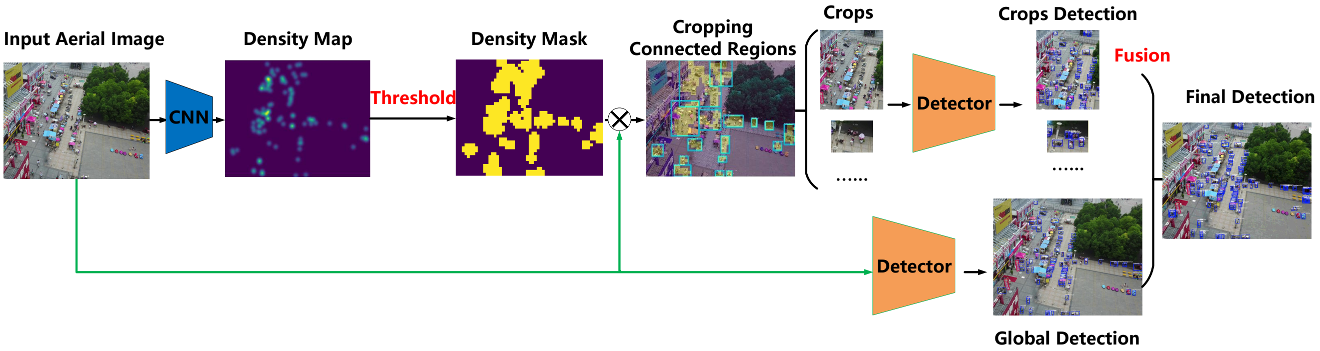


Figure 2: Overview for the DMNet framework. First, DMNet learns features of aerial images and predicts density map via the density generation module. Then it utilizes a sliding window (Section 3.2) on the density map to obtain density mask and applies connected component algorithm to generate proposal regions for cropping. The generated image crops and the original global aerial image are feed into the same object detector for object detection. Finally, detection results from the global image and crops are fused to generate the final detection. More details are presented in Section 3.

cost of accuracy drop. Some object detection tasks may suffer from data imbalance issue. To solve the issue, RetinaNet [11] introduces focal loss, which is a variation of cross entropy loss. It places more weights on hard examples than easy examples to guide detector to pay more attention to hard-to-learn objects.

2.2. Object detection in aerial images

Aerial image object detection faces more challenges compared with general object detection. First, small objects account for a higher percentage in aerial image dataset, which requires more attention on small objects [30]. Second, the object scale varies per image, per category due to the change of camera viewpoint. Third, data imbalance issue exists in aerial image dataset since some categories (such as tricycle and awning-tricycle in VisionDrone [30] dataset) rarely show in real world. Finally, aerial images may have object occlusion issue during photoing. Many research works have been developed to address these challenges.

[15] suggests that tiling helps improve detection performance of small objects. To counter the scale variation caused by the change of viewpoint, in [20], a detection network is proposed to increase the receptive field for high-level semantic features and to refine spatial information for multi-scale object detection. [26] proposes a cluster network to crop regions of dense objects and leverages a ScaleNet to adjust generated shape of crops. The final detection result is fused from cropped images and the original image to improve overall performance. [27] pays attention to learn regions with low scores from a detector and gains performance by better scoring those low score regions. To solve data imbalance issue, [27] introduces IOU-sampling method and a balanced L1 loss. Moreover, [19, 28] discuss challenges and insights for object detection in Very High Resolution (VHR) remote sensing imagery.

2.3. Density map estimation

Density map is commonly used in crowd counting literature. Crowd counting requires to estimate the head counts for a given scene where a large number of people present. Due to the high density of objects, general object detectors fail to detect and count the number of people correctly. Since density map can reflect the head location and offer spatial distribution, it turns out to be a better solution since an integral of density map can approximate head counts. Such method provides higher accuracy and thus is widely used in counting tasks.

To improve the performance of density map based counting, [29] proposes geometry adaptive and fixed kernels with Gaussian convolution to generate density map. [10] further improves the quality of density map by introducing a VGG16-based dilated convolutional neural network. [25] observes that the large difference in object scales leads to a great variation in density map. A scale preservation and adaption network is thus introduced to balance the pixel difference in generated density maps for robust counting performance. [21] captures the pixel-level similarity in original images and implements the locally linear embedding algorithm to estimate density maps while persevering the geometry property. [22] further improves the quality of generated density maps by introducing a sparsity constraint which is motivated by manifold learning.

3. Density Map guided detection Network (DMNet)

3.1. Overview

As shown in Fig. 2, DMNet consists of three components, which are density map generation module, image cropping module and fusion detection module. In detail, we first train a density map generation network to predict density map for each aerial image. Afterwards, we apply a sliding window on the generated density map to gather the sum of pixel intensities and compare its value with a density threshold to form a density mask. We connect the windows whose pixel intensity is above the density threshold to generate image crops. The final detection result will be fused from detection on the image crops and the original image.

3.2. Density map generation

3.2.1 Density map generation network

Density map is of great significance in the context of crowd counting. [29] proposes the Multi-column CNN (MCNN) to learn density map for crowd counting task. Due to the variation of head size per image, single column with fixed receptive field may not capture enough features. Therefore three columns are introduced to enhance feature extraction. In aerial image object detection, the general categories can be broadly divided to three sub-categories by scale (small, medium and large). To capture the balanced feature patterns in all scales, we adopt MCNN [29] in our approach to generate object density map for image cropping.

The loss function for training density map generation network is based on the pixel-wise mean absolute error, which is given as below:

$$L(\Theta) = \frac{1}{2N} * \sum_{i=1}^{N} \|D(X_i;\Theta) - D_i\|^2. \qquad (1)$$

Θ is the parameters of density map generation module. N is the total number of images in the training set. X_i is the input image and D_i is the ground truth density map for image X_i. $D(X_i;\Theta)$ stands for the generated density map by the density generation network.

As MCNN [29] introduces two pooling layers, the output feature map will shrink by $4\times$ for both height and width. To preserve the original resolution, we upsample the generated density map by $4\times$ with cubic interpolation to restore the original resolution. For the case where the image height or width is not the multiplier of four, we directly resize the image to its original resolution.

As reported in [1], it is also a working solution to add the same number of upsampling layers to restore the resolution. However, only a slight difference (approximately 0.02 in terms of mean absolute error in evaluation) is observed for this approach in our experiment. However, the size of feature maps is largely increased during training, which may cause memory issue for images with large resolution. Therefore, we do not introduce upsampling layers in our density map generation network.

3.2.2 Ground truth object density map

To generate the ground truth object density maps for aerial images in the training stage, we follow the similar idea as proposed in [29] and [10] for crowd counting, where two methods, geometry-adaptive and geometry-fixed kernel, are developed. Both methods follow the similar concepts. We use Gaussian kernel (normalized to 1 in general) to blur each object annotation to generate ground truth density maps. The key to distinguish adaptive kernel from fixed kernel is the spread parameter σ. It is fixed in fixed kernel but is computed by the K-Nearest-Neighbor (KNN) method for adaptive kernel. The formula for geometry-adaptive kernel is defined in Eq. 2 [29],

$$F(x) = \sum_{i=1}^{N} \delta(x - x_i) \times G_{\sigma_i}(x), with\ \sigma_i = \beta \bar{d}_i, \qquad (2)$$

where x_i is the target of interest. $G_{\sigma_i}(x)$ is the Gaussian kernel, which convolves with $\delta(x - x_i)$ to generate ground truth density map. $\bar{d}_i$ is the average distance of K nearest targets. In our implementation, we prefer the fixed kernel as we consider the following assumptions for geometry-adaptive kernel are violated. (1) The objects are neither in single class nor evenly distributed per image, resulting in no guarantee for accurate estimation of geometric distortion. (2) It is not reasonable to assume the object size is related to the average distance of two neighboring objects, since objects in aerial images are not so densely distributed as in crowd counting. Based on these considerations, we choose geometry-fixed kernel accordingly.

3.2.3 Improving ground truth with class-wise kernel

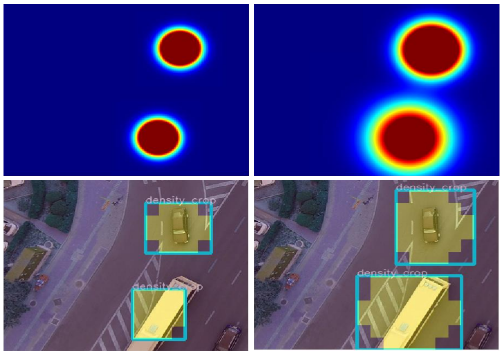
In fixed kernel method, the standard deviation of Gaussian filters is constant for all objects, regardless of the shape of the exact object. This leads to possible truncation when cropping large objects (such as buses). One example is provided at the top-right of Fig. 3.

To resolve the possible truncation issue, we propose the class-wise density map ground truth generation method. To start, exploratory data analysis is performed on the training set to analyze the average scale for each target category. Then we compute σ by estimating the average scale for each object category.

Assuming that the average height and width for a category is H_i and W_i, where i is the current object category, we estimate σ by applying Eq. 3:

$$\sigma_i = \frac{1}{2}\sqrt{H_i^2 + W_i^2}. \qquad (3)$$

We record those σ values for each category and apply them to Eq. 2 to generate density maps. In this case, we are able to accommodate the scale of medium and large objects



Fixed Kernel

Class-wise Kernel

Figure 3: Visual comparison between fixed kernel and class-wise kernel. Left top is the density map for fixed σ. Left bottom is its corresponding cropping results. As can be observed, the bus is not fully covered by the light blue rectangle, which results in truncation. To resolve this issue, we replace the fixed σ with the average scale of bus category (right top). Then the light blue rectangle (right bottom) is able to fully cover the bus. Light blue rectangle represents the candidate region to crop.

in a more suitable manner. A comparison between fixed kernel and our proposed class-wise kernel for ground truth density map generation is provided in Fig. 3.

3.3. Image cropping based on density map

3.3.1 Density mask generation

The core of DMNet is to properly crop images from the contextual information provided by density maps. As observed from the density mask provided in Fig. 1, the regions with more objects (labeled in yellow color) have higher pixel intensities compared with those with fewer objects. By placing a threshold within a region, we can estimate the object counts and filter out pixels in the region with no or limited objects accordingly.

We introduce a sliding window on a density map, where the size of the window is the average size of the objects in the training set. We slide the window with the step of window size (*i.e.*, non-overlapping). Then we sum all pixel values in the current window and compare the sum with the density threshold. If the sum value is below the threshold, then the pixels in this window will all have 0 value, and "1" for the opposite case. This leads to a density mask with 0 and 1 values. The detailed implementation is illustrated in Algorithm 1.

The density threshold is introduced to control the noise from predicted density map. In the meanwhile, it dynamically adjusts the number of objects finally collected per density crop. By increasing the threshold, the boundary will be irregular and pixels on the boundary will be more likely to be filtered out at a higher threshold. This leads to more crops with some only have a few objects. Fig. 4 provides a visualization to graphically explain how different density thresholds may affect the cropping boundary.

Algorithm 1 Density mask generation

```
**Input:** Aerial image $Img$. Density map $Den$. Sliding window size $W_h, W_w$. Density threshold $TH$.
**Output:** Density mask $M$.
▷ Initialization.
$I_h, I_w = Img.height, Img.width.$
$M =$ zeros $(I_h, I_w)$
▷ Generate density mask
**for** $h$ in $range(0, I_h, W_h)$ **do**
  **for** $w$ in $range(0, I_w, W_w)$ **do**
    $S =$ sum $(Den[h : h + W_h, w : w + W_w])$
    **if** $S > TH$ **then**
      $M[h : h + W_h, width : width + W_w] = 1$
    **end if**
  **end for**
**end for**
**return** $M$
```

3.3.2 Generating density crops from density mask

The generated density mask indicates the presence of objects. We generate image crops based on the density mask. First, we select all the pixels whose corresponding density mask value is "1". Second, we merge the eight-neighbor connected pixels into a large candidate region. Finally, we use the candidate region's circumscribed rectangle to crop the original image. We filter out the crops whose resolution is below the density threshold. The reasons are: (1) some of the predicted density maps are not in high quality and contain noise that spreads over the whole map given a low density threshold. Thus, it is likely to obtain some random single windows as the single crop. Keeping such crops is not desired. (2) Object detectors cannot perform well on low resolution crops, as crops become really blurry after resizing to the original input size.

3.4. Object detection on density crops

After obtaining image crops from the density map, the next step is to detect objects and fuse results from both density crops and the whole image. Any existing modern detectors can be of the choice. We first run separate detection on original validation set and density crops. Then we collect the predicted bounding boxes from density crops detection and add them back to the detection results of original images to fuse them together. Finally, we apply non maximum suppression (NMS) to all bounding boxes and calculate the final results. The threshold of NMS is 0.5 which follows

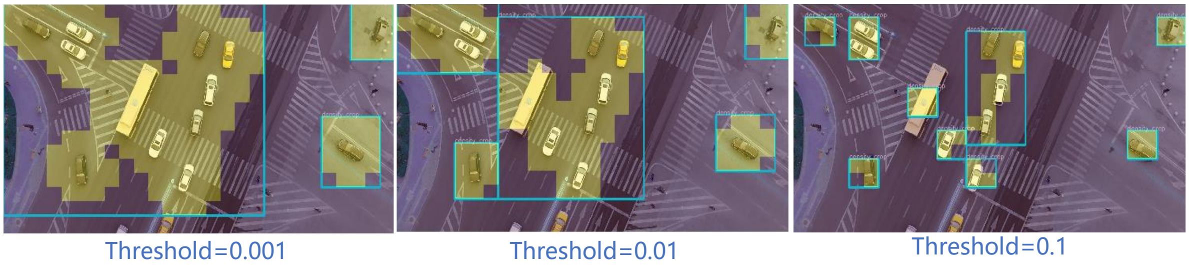


Figure 4: Visualization of density mask under different thresholds. As the threshold increases, the yellow region shrinks and one large region breaks into disconnected sub-regions. Yellow region is the candidate crop region and the light blue bounding box indicates the full region to crop.

the setting in [26]. Note that in our fusion design, we do not remove bounding boxes from original detection result. From our visualization analysis, we observe that the original detection results contain large objects that are correctly detected. Removing those detection will result in a drop in AP_{large}, which does not fully show the performance of the detector. Thus we keep those detected bounding boxes during evaluation.

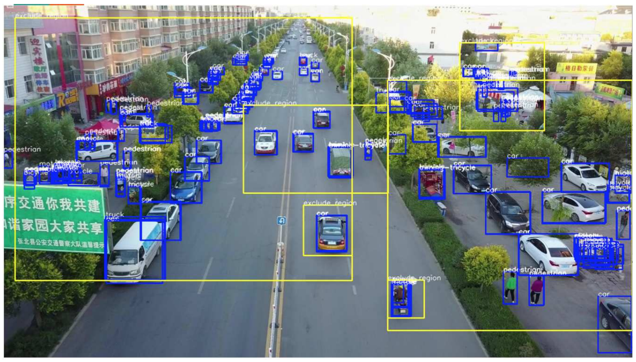


Figure 5: A visual example of the final detection result. The yellow rectangles represent regions of density crops. The blue rectangles represent ground-truth bounding boxes. The bounding boxes from both density crops and the whole images in inference stage are kept and labeled on the plot, as well as their corresponding categories. NMS is applied after obtaining the fusion bounding boxes. Thus we do not show it in this figure.

4. Experiments

4.1. Implementation details

Our implementation is based on the MMDetection toolbox [3]. The MCNN [29] is selected as the baseline network for density map generation. For object detector, we use Faster R-CNN with Feature Pyramid Network (FPN). Unless specified, we use the default configurations for all the experiments. We use ImageNet [18] pre-trained weights to train the detector. The density threshold is set to 0.08 in both training and testing phases for VisionDrone dataset and 0.03 for UAVDT dataset. The minimal threshold for filtering bounding boxes is set to 70 × 70, which follows the similar setting in [26].

The density map generation module is trained for 80 epochs using the SGD optimizer. The initial learning rate is 10^{-6}. The momentum is 0.95 and the weight decay is 0.0005. We only use one GPU to train the density map generation network and no data argumentation is used.

For the object detector, we set the input size to 600 × 1,000 on both datasets. We follow the similar setup in [26] to train and test on the datasets. The detector is trained for 42 epochs on 2 GPUs, each with a batch size of 2. The initial learning rate is 0.005. We decay the learning rate by the factor of 10 at 25 and 35 epochs. The threshold for non-max suppression in fusion detection is 0.7. The maximum allowed number for bounding boxes after fusion detection is 500. Unless specified, we use MCNN to generate density map and Faster R-CNN with FPN to detect objects for all the experiments.

4.2. Datasets

To show the effectiveness of the proposed method, we evaluate the performance of DMNet on two popular datasets, VisionDrone 2018 [30] and UAVDT [4].

VisionDrone. VisionDrone is a widely used dataset for aerial image detection. It includes 10,209 aerial images in total. In detail, there are 6,471 training images, 548 validation images and 3,190 testing images. Ten categories are provided for evaluation purpose with abundant annotations. The image scale is about 2,000 × 1,500 pixels. Due to the fact that we have no access to the test data and the evaluation server, we cannot evaluate our method on the test set. As an alternative, we use the validation set to evaluate the performance, which is also the choice of existing works [26, 27].

UAVDT. UAVDT has a rich amount of images (23,258 training images and 15,069 test images) for aerial image

Table 1: Quantitative result on VisionDrone dataset. “Test data” represents the type of data used. “Original” is for the original validation data. “Cluster” and “Density” denote cluster crops [26] and our density crops respectively. “#img” is the number of images that send to the detector. In the experiment, we select Average precision (AP) as the primary metric to measure the overall performance.

Method	Backbone	Test data	#Image	AP	AP_{50}	AP_{75}	AP_{small}	AP_{mid}	AP_{large}
DetecNet+CPNet+ScaleNet [26]	ResNet 50	Original+cluster	2716	26.7	50.6	24.7	17.6	38.9	51.4
DetecNet+CPNet+ScaleNet [26]	ResNet 101	Original+cluster	2716	26.7	50.4	25.2	17.2	39.3	54.9
DetecNet+CPNet+ScaleNet [26]	ResNeXt 101	Original+cluster	2716	28.4	**53.2**	26.4	19.1	40.8	54.4
DMNet	ResNet 50	Original+density	2736	28.2	47.6	28.9	19.9	39.6	55.8
DMNet	ResNet 101	Original+density	2736	28.5	48.1	29.4	20.0	39.7	**57.1**
DMNet	ResNeXt 101	Original+density	2736	**29.4**	49.3	**30.6**	**21.6**	**41**	56.9

Table 2: Quantitative result for UAVDT dataset.

Method	Backbone	#Image	AP	AP_{50}	AP_{75}	AP_{small}	AP_{mid}	AP_{large}
R-FCN [6]	ResNet 50	15096	7.0	17.5	3.9	4.4	14.7	12.1
SSD [14]	N/A	15096	9.3	21.4	6.7	7.1	17.1	12.0
RON [8]	N/A	15096	5.0	15.9	1.7	2.9	12.7	11.2
FRCNN [17]	VGG	15096	5.8	17.4	2.5	3.8	12.3	9.4
FRCNN [17]+FPN [12]	ResNet 50	15096	11.0	23.4	8.4	8.1	20.2	26.5
ClusDet [26]	ResNet 50	25427	13.7	26.5	12.5	9.1	25.1	31.2
DMNet	ResNet 50	32764	**14.7**	24.6	**16.3**	**9.3**	**26.2**	**35.2**

object detection. It has three categories, namely car, truck and bus. Those (except car) all have a larger size compared with categories in VisionDrone. The resolution for UAVDT is about 1,024 × 540 pixels.

4.3. Evaluation metric

We follow the same evaluation metric as proposed in MS COCO [13]. Six evaluation metrics are employed, namely AP (average precision), AP_{50}, AP_{75}, AP_{small}, AP_{medium} and AP_{large}. The AP is the average precision under multiple IoU thresholds, ranging from 0.50 to 0.95 with a step size of 0.05. Since AP considers all thresholds, we use AP to measure and compare the performance between the proposed method and other competing approaches. Meanwhile, as the number of generated image crops will affect the inference speed, we also record image counts in the table for a fair comparison. We denote “#img” for the total number of images (including both original images and density crops) we used in the validation set.

4.4. Quantitative result

In this section, we evaluate the proposed DMNet on VisionDrone and UVADT datasets. Table 1 shows the results on VisionDrone. We can see that DMNet consistently outperforms ClusDet [26] by 1-2 points on three different backbone networks. Specifically, DMNet achieves the state-of-the-art performance of 29.4 AP with the ResNetXt101 backbone. This clearly exceeds all previous methods. Moreover, the result of AP_{75} improves nearly 4 points compared with ClusDet [26], indicating the robustness of DMNet at higher IoU thresholds. We also observe more than 2 points improvements on AP_{small} under different backbones, which suggests that the proposed density map crops significantly help the detection for small scale objects.

Table 2 shows the results of different methods on UVADT. It can be seen that general object detectors fail to achieve a comparable result as discussed in Sec 1. Similar to the results in VisionDrone, DMNet substantially outperforms ClusDet and achieves the state-of-the-art performance of 14.7 AP on UVADT. Particularly, DMNet consistently improves the accuracy on small scale, medium scale and large scale objects. This validate the effectiveness of our generated crops based on density maps.

Inference speed. Here we report the inference speed of the proposed DMNet. We conduct the experiment on one GTX 1080 Ti GPU per task. The inference speed on three backbones (ResNet 50, ResNet 101 and ResNeXt 101) is 0.29 s/img, 0.36 s/img and 0.61 s/img, respectively.

4.5. Ablation study

In this section, we design a series of ablation studies to analyze the contribution of each component in the proposed DMNet. In all experiments, we use MCNN [29] as the density generation backbone and Faster RCNN [17] as the detector. The input image size is 600 × 1000.

Density threshold. The density threshold is an important factor as it controls how to generate density crops. In this experiment, we remove thresholding by keeping all windows whose pixel intensity is larger than 0. From Table 3 we can clearly see that AP drops drastically without thresholding. From the previous result analysis, we examine the generated crops and find most of them are large and cover many objects, which makes it difficult to detect small objects. Since no threshold is applied, more background pixels are cropped, which further affect the performance of detector.

Table 3: Ablation study on VisionDrone Dataset.

Method	AP	AP$_{small}$	AP$_{mid}$	AP$_{large}$
FRCNN [17]+FPN [12]	21.4	11.7	33.9	54.7
DMNet without thresholding	22.6	11.8	37.5	58.5
Uniform cropping without fusion	24.5	19.1	31.9	22.4
DMNet without fusion	25.9	19.4	38.1	41.6
DMNet with all components	28.2	19.9	39.6	55.8

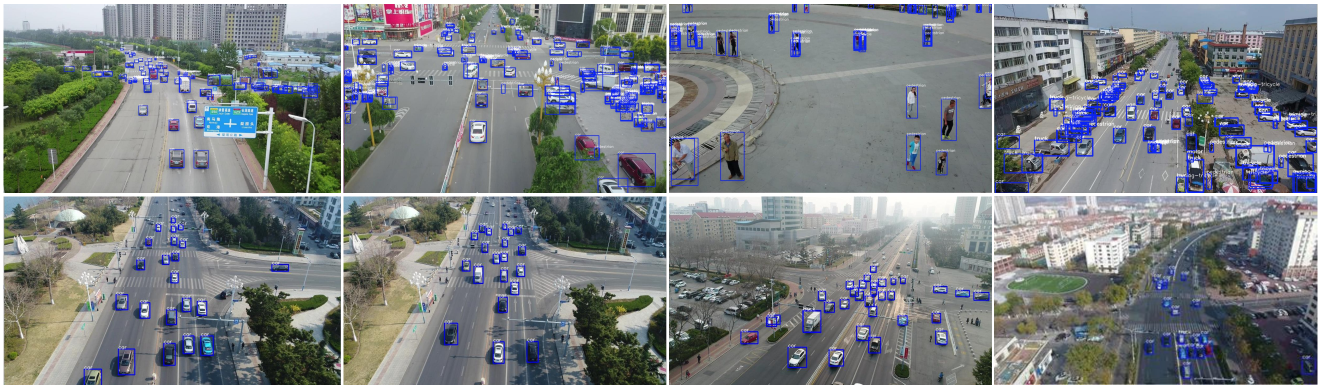


Figure 6: Visualization of our DMNet detection results on VisionDrone (first row) and UAVDT (second row).

Comparison with uniform crops. As discussed in Sec 1, aerial images contain a majority of small scale objects. DMNet is able to effectively crop small objects from the whole image and significantly improve AP$_{small}$ as stated in Table 1. But one can also get small objects by uniform cropping with a very small window size. In this experiment, we replace our density crops with 3×4 uniform cropping, where the size of each uniform crop is small to benefit small object detection. As shown in Table 3, this method fails to beat DMNet, although it improves nearly 3 points on AP compared with the baseline. The reason is that although small uniform crops are able to help small object detection, they also increase the risk of cutting off large objects. We can see that the AP$_{small}$ is comparable with DMNet while there is a large drop in AP$_{medium}$ and AP$_{large}$. This demonstrate the superiority of our DMNet since it is able to better accommodate object scales and thus achieves better performance.

Contribution of density crop detection. Directly detecting objects on image crops instead of the original image can give better performance as reported in [26]. However, how it contributes to the final fusion detection remains unclear. Therefore, we additionally report performance of DMNet with only detection on images crops (*i.e.*, without fusing the results of detection on the original whole images). The results are provided in Table 3. We can conclude that density crop detection primarily contributes to AP$_{small}$ and AP$_{mid}$ as the large performance improvements have been observed on those two categories. Meanwhile, detection on the original image contributes more on the AP$_{large}$ category, compared with density crop detection.

5. Conclusion

In this paper, we propose the density map guided detection network (DMNet) to address the challenges in aerial image object detection. Density map provides spatial distribution and collects window-based pixel intensity to implicitly form the boundary of a potential cropping region, which benefits the following image cropping process. The proposed DMNet achieves state-of-the-art performance on two popular aerial image detection datasets under different backbone networks. Extensive ablation studies are conducted to analyze the contribution of each component in DMNet. Our proposed density map based image cropping strategy provides a promising direction to improve the detection accuracy in high resolution aerial images.

6. Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant CNS-1910844. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] Reza Bahmanyar, Elenora Vig, and Peter Reinartz. MRCNet: Crowd Counting and Density Map Estimation in Aerial and Ground Imagery. *arXiv e-prints*, page arXiv:1909.12743, Sept. 2019.

[2] Yancheng Bai, Yongqiang Zhang, Mingli Ding, and Bernard Ghanem. SOD-MTGAN: Small Object Detection via Multi-Task Generative Adversarial Network. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 210–226, Cham, 2018. Springer International Publishing.

[3] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[4] Dawei Du, Yuankai Qi, Hongyang Yu, Yifan Yang, Kaiwen Duan, Guorong Li, Weigang Zhang, Qingming Huang, and Qi Tian. The Unmanned Aerial Vehicle Benchmark: Object Detection and Tracking. *arXiv e-prints*, page arXiv:1804.00518, Mar. 2018.

[5] Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision*, 88(2):303–338, June 2010.

[6] Ross Girshick. Fast R-CNN. *arXiv e-prints*, page arXiv:1504.08083, Apr. 2015.

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. *arXiv e-prints*, page arXiv:1703.06870, Mar. 2017.

[8] Tao Kong, Fuchun Sun, Anbang Yao, Huaping Liu, Ming Lu, and Yurong Chen. RON: Reverse Connection with Objectness Prior Networks for Object Detection. *arXiv e-prints*, page arXiv:1707.01691, July 2017.

[9] Jianan Li, Xiaodan Liang, Yunchao Wei, Tingfa Xu, Jiashi Feng, and Shuicheng Yan. Perceptual Generative Adversarial Networks for Small Object Detection. *arXiv e-prints*, page arXiv:1706.05274, Jun 2017.

[10] Yuhong Li, Xiaofan Zhang, and Deming Chen. CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes. *arXiv e-prints*, page arXiv:1802.10062, Feb. 2018.

[11] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, Oct 2017.

[12] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. *arXiv e-prints*, page arXiv:1612.03144, Dec. 2016.

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common Objects in Context. *arXiv e-prints*, page arXiv:1405.0312, May 2014.

[14] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng Yang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016.

[15] F. Ozge Unel, Burak O. Ozkalayci, and Cevahir Cigla. The power of tiling for small object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[16] Joseph Redmon and Ali Farhadi. YOLOv3: An Incremental Improvement. *arXiv e-prints*, page arXiv:1804.02767, Apr. 2018.

[17] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 91–99, Cambridge, MA, USA, 2015. MIT Press.

[18] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[19] C. Wang, X. Bai, S. Wang, J. Zhou, and P. Ren. Multiscale visual attention networks for object detection in vhr remote sensing images. *IEEE Geoscience and Remote Sensing Letters*, 16(2):310–314, 2019.

[20] Haoran Wang, Zexin Wang, Meixia Jia, Aijin Li, Tuo Feng, Wenhua Zhang, and Licheng Jiao. Spatial attention for multi-scale feature refinement for object detection. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[21] Y. Wang and Y. Zou. Fast visual object counting via example-based density estimation. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3653–3657, Sep. 2016.

[22] Y. Wang, Y. X. Zou, J. Chen, X. Huang, and C. Cai. Example-based visual object counting with a sparsity constraint. In *2016 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, July 2016.

[23] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3974–3983, 2018.

[24] Haipeng Xiong, Hao Lu, Chengxin Liu, Liang Liu, Zhiguo Cao, and Chunhua Shen. From Open Set to Closed Set: Counting Objects by Spatial Divide-and-Conquer. *arXiv e-prints*, page arXiv:1908.06473, Aug. 2019.

[25] Chenfeng Xu, Kai Qiu, Jianlong Fu, Song Bai, Yongchao Xu, and Xiang Bai. Learn to Scale: Generating Multipolar Normalized Density Maps for Crowd Counting. *arXiv e-prints*, page arXiv:1907.12428, July 2019.

[26] Fan Yang, Heng Fan, Peng Chu, Erik Blasch, and Haibin Ling. Clustered object detection in aerial images. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[27] Junyi Zhang, Junying Huang, Xuankun Chen, and Dongyu Zhang. How to fully exploit the abilities of aerial image detectors. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[28] Shuo Zhang, Guanghui He, Hai-Bao Chen, Naifeng Jing, and Qin Wang. Scale adaptive proposal network for object detection in remote sensing images. *IEEE Geoscience and Remote Sensing Letters*, 16(6):864–868, 2019.

[29] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma. Single-image crowd counting via multi-column convolutional neural network. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 589–597, June 2016.

[30] Pengfei Zhu, Longyin Wen, Xiao Bian, Haibin Ling, and Qinghua Hu. Vision Meets Drones: A Challenge. *arXiv e-prints*, page arXiv:1804.07437, Apr 2018.

[31] Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Haibin Ling, Qinghua Hu, Qinqin Nie, Hao Cheng, Chenfeng Liu, Xiaoyu Liu, et al. Visdrone-det2018: The vision meets drone object detection in image challenge results. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.