OBJECT DETECTION IN AERIAL IMAGE

by

Changlin Li

A thesis submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Master of Science in
Computer science

Charlotte

2020

Approved by:

_____

Dr.Chen Chen

_____

Dr.Pu Wang

_____

Dr.Qiong Cheng

ABSTRACT

CHANGLIN LI. Object detection in aerial image. (Under the direction of DR.CHEN CHEN)

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 [1] and UAVDT [2].

ACKNOWLEDGEMENTS

Research work is challenging with surprises happen. Sometimes you may run into a dead end and may not even realize it. So the guidance and instructions from the professors are highly important to me.

Thus, I want to say thank you to my thesis advisor, Dr. Chen Chen. He devotes to my research and is willing to guide me into my current research topic, aerial image detection, leading me to my first publication. In the meanwhile, my academic advisor, Dr.Pu Wang helps me in my master research, solving my academic issues I encountered whenever possible. I highly appreciate his help.

Dr. Qiong Cheng offered me parallel computing course in UNC Charlotte in my first semester. She encourages me to challenge myself and believes that I can make progress towards the end from the beginning of my search. As I finally make it, I want to show my gratitude to Dr. Qiong for her encouragements.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1 Motivation

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 [4], RetinaNet [5], SSD [6]). Although these methods achieve good performance on natural image datasets (*e.g*, MS COCO dataset [7] and Pascal VOC [8] 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 [9], therefore their visual appearance can be significantly different from natural images like ImageNet [10]. 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. Exploratory data analysis results from VisionDrone 2018 training set [1] confirms the occurrence of those challenges. Please check Fig. 1.2 for further reference.

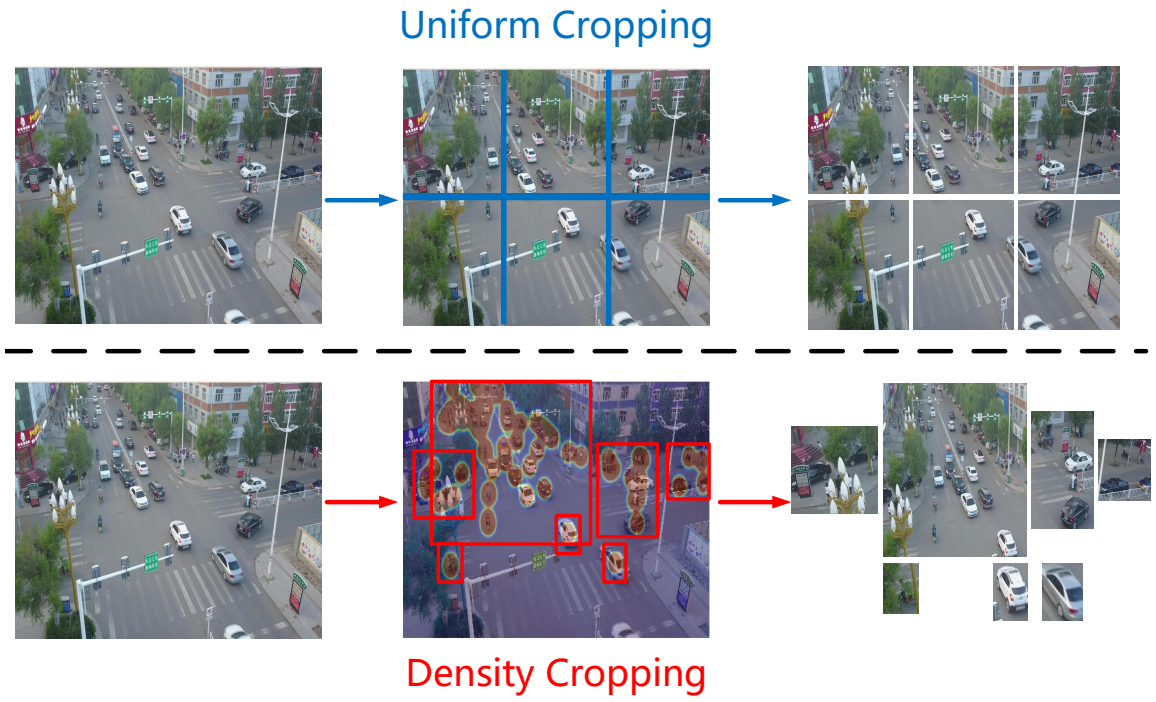

Figure 1.1: Visualization of density cropping vs. uniform cropping. Top row provides an example of uniform cropping. Bottom row gives a comparable example of density cropping. Uniform crops have more background pixels and fail to accommodate the bounding box resolution of different categories compared 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.

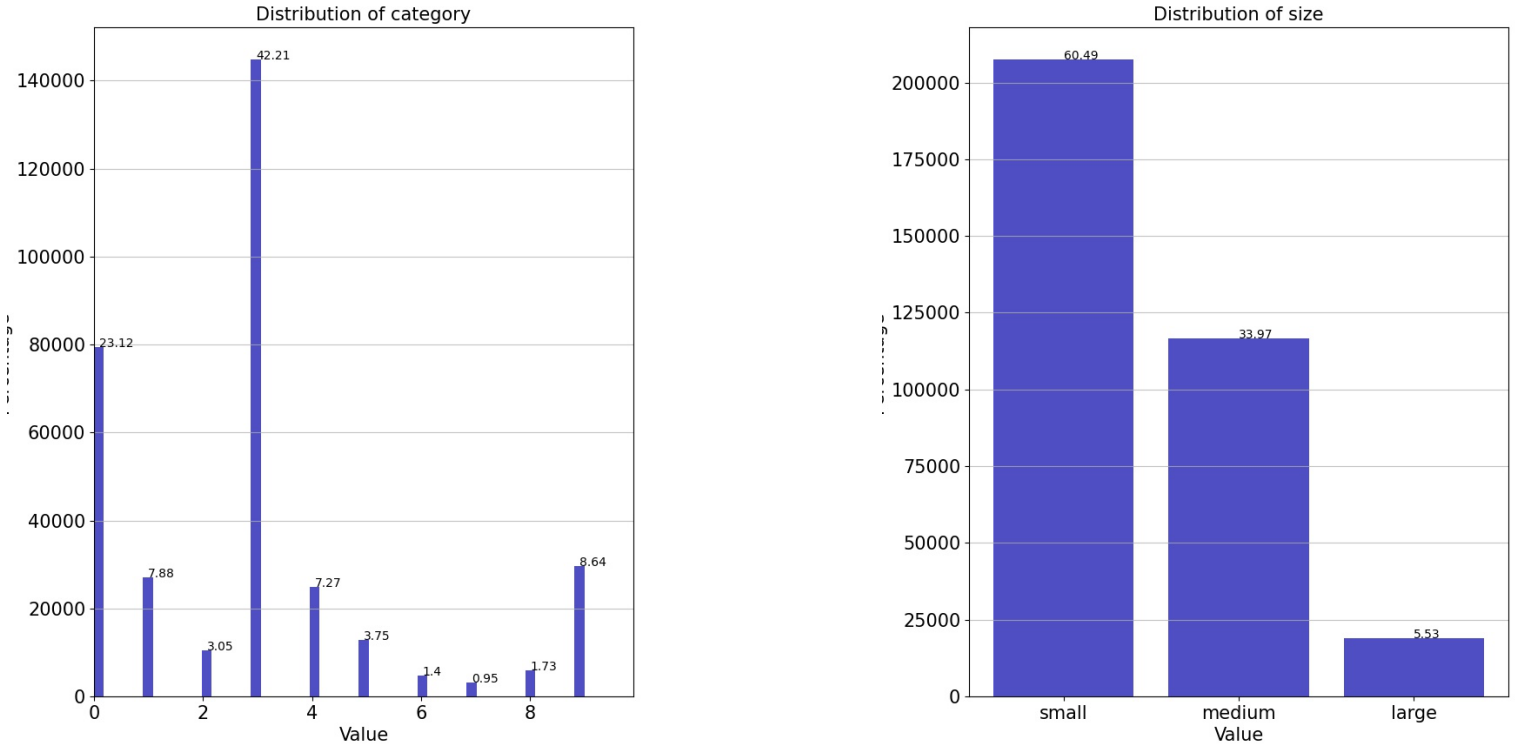

Figure 1.2: Visualization of distribution for scales and categories in Visiondrone 2018 training set

Early works [11, 12] on aerial image object detection simply leverage the general object detection architecture and focus on improving the detection of small objects. [11] introduces the upsampling module after feature extraction to increase spatial resolution. [12] 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 [1, 13, 9], they are far from satisfactory for practical applications.

To address the scale variation problem, another promising research direction is to crop the original image into small crops/chips before applying the object detection, such as uniform cropping [14] 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 [15], 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.

1.2 Methodology

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. 1.1. 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. 1.3 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 [3], 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)).

1.3 Contribution

In summary, the DMNet 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 [3].
- Extensive experiments suggest that the proposed method achieves the state-of-the-art performance on representative aerial image datasets, including VisionDrone [1] and UAVDT [2].

1.4 Organization

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

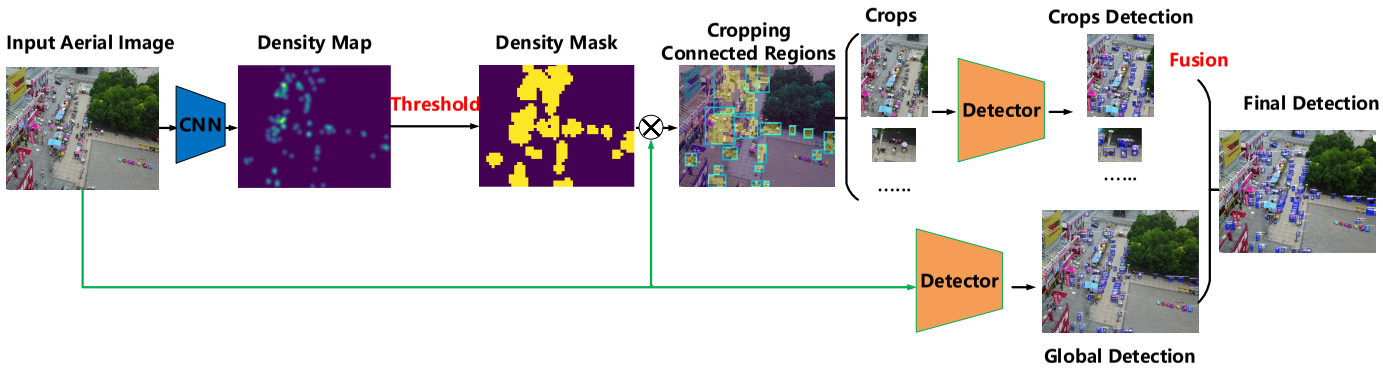

Figure 1.3: Overview for the DMNet framework. First, DMNet learns features of aerial images and predicts density map via the density generation module. Then it utilizes a sliding window (Section 3.2) on the density map to obtain density mask and applies connected component algorithm to generate proposal regions for cropping. The generated image crops and the original global aerial image are fed 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.

CHAPTER 2: LITERATURE REVIEW

2.1 General object detection

General object detection targets primarily on natural images. Proposal-based detectors introduce the concept of anchors with two stages. Fast R-CNN [16] generates proposals using selective search and then extracts features and classifies objects accordingly based on those proposals. Faster R-CNN [4] achieves similar functionality by introducing the region proposal network (RPN), which significantly accelerates the inference speed. Mask R-CNN [17] extends Faster R-CNN to perform detection and instance segmentation tasks simultaneously with innovations of ROI align layers.

On the other hand, YOLO3 [18], SSD [6] and RetinaNet [5] are examples of single stage detectors. Single stage detectors skip proposal stage and detect directly on the sampled feature map. They improve detection speed at the cost of accuracy drop. Some object detection tasks may suffer from the data imbalance issue. To solve the issue, RetinaNet [5] 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.

Recently, anchor-free detectors receives attention for its ability to achieve state-of-art detection performance, while skipping the step to manually find suitable anchors for customer dataset. Some of the examples for anchor-free detectors are CornerNet [19] and CenterNet [20].

CornerNet [19] saves the trouble to design suitable anchors by replacing anchor detection mechanism with object pairs (left-top and right-bottom). Two sub-networks are designed to regress the coordination of those object pairs. CornerNet also Introduces corner pooling to help extract feature of corner information.

CenterNet [20] improves CornerNet by adding information of object center to detector, which reduce the false positive rate of detection. Instead of use two key points, three key points (upper-left, center and lower-bottom) are introduced to better localize the objects.

Foveabox [21] introduces feature pyramid network as backbone for feature extraction purpose. It simulates "Fovea" of human eyes, where the center of vision field shares highest visual activities. FoveaBox detects targets by their scale and divides scale into multiple bins for different levels of feature map. Focal loss is implemented to solve the imbalance of positive-negative samples.

2.2 Object detection in aerial images

Aerial image object detection addresses more challenges compared with general object detection task. And many research works have been developed to address these challenges.

- Small objects account for a higher percentage in aerial image dataset, which requires detectors to pay more attention on small objects [1].
- The object scale varies per image, per category due to the change of camera viewpoint.
- Data imbalance issue exists in aerial image dataset since some categories (such as tricycle and awning-tricycle in VisionDrone [1] dataset) rarely show in real world.
- Aerial images may have object occlusion issue during photoing.

[14] suggests that tiling helps improve detection performance of small objects. [22] modifies yolo3 detector to make it suitable for aerial image detection. To counter the scale variation caused by the change of viewpoint, in [23], 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. [3] proposes a cluster network to crop regions of dense objects and leverages a scale network to adjust generated shape of crops. The final detection result is fused from both cropped images and the original image to improve overall performance. [24] 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, [24] introduces IOU-sampling method and a balanced L1 loss, which shares the similar design with [25]. Moreover, [26, 27] discusses 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 crowds present. Due to the high density of objects, general object detectors fail to detect and count the number of crowds correctly.

In this section density map related literature will be reviewed. Since density map can reflect the head locations and offer spatial distribution, it turns out to be a better solution as 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, [28] proposes geometry adaptive and fixed kernels with Gaussian convolution to generate density map. [29] further improves the quality of density map by introducing a VGG16-based dilated convolutional neural network. [30] 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. [31] captures the pixel-level similarity in original images and implements the locally linear embedding algorithm to estimate density maps while persevering the geometry property. [32] further improves the quality of generated

density maps by introducing a sparsity constraint which is motivated by manifold learning.

2.4 Data imbalance in detection

Data imbalance turns out to be a general issue in object detection. And aerial image detection also suffers from this. Many attempts have been made to mitigate the negative effect of data imbalance in object detection.

1st solution of OpenImage 2019 [33] implements expert model to solve the issues for data imbalance. To summarize, the dataset can be further divided into two groups, with limited and abundant samples. So we can train two detectors to detect different categories and merge them together. As the samples in each categories are not so imbalanced, detectors can learn better and thus are able to generalize well.

Copy-paste argumentation [34] solves data imbalance issues by directly copy objects of minor classes to new images and thus upsamples the amount of bounding boxes of minor classes. The edges of copied objects are smoothed with filters to ensure that share edges will not interrupt training process.

Yolov4 [35] introduces mosaic argumentation, which can be treated as a variation of copy-paste argumentation. Mosaic argumentation generates new training data by utilizing available images and bounding boxes, based on contextual information. It not only can upsample bounding boxes of minor classes, but also can argument background images with bounding boxes of different categories, which combines the advantages of both image-level upsampling and bounding boxes level upsampling.

Patch level argumentation [36] borrows similar idea from copy-paste argumentation [34] to address data imbalance issue. To summarize, random sampling is introduced to sample bounding boxes for each category. Then those sampled bounding boxes will be pasted to each aerial image to balance the distribution of dataset. The performance of the proposed algorithm ranks top 3 in Visiondrone 2019 challenge [37].

RRNet [38] also attempts to copy bounding boxes to solve data imbalance issue.

The position to paste is carefully selected to make sure the resulting images will not conflict with real world scenario. To achieve this, pre-train segmentation model is applied to segment out road, sky, building and etc. Then objects will be pasted accordingly.

CHAPTER 3: DENSITY MAP GUIDED DETECTION NETWORK

3.1 Overview

As shown in Fig. 1.3, 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 pixels in terms of intensities and compare its value with a density threshold to form a density mask. We connect the windows whose pixel intensities are above the density threshold to generate image crops. The final detection result will be fused based on detection from both the image crops and the original image.

3.2 Density map generation network

Density map is of great significance in the context of crowd counting. [28] proposes the Multi-column convolutional neural network (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 [28] 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 square error, which is given as below:

$$L(\Theta) = \frac{1}{2N} * \sum_{i=1}^{N} \|D(X_i; \Theta) - D_i\|^2. \tag{3.1}$$

where 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 [28] 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.

For DMNet we need to ensure same resolution for both aerial image and generated density map. As reported in [39], 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 when compare this method with its alternative (Namely, not add upsampling layers, which is proposed). However, the size of feature maps is greatly 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.

3.3 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 [28] and [29] 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 a constant value 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. 3.2 [28],

$$F(x) = \sum_{i=1}^{N} \delta(x - x_i) \times G_{\sigma_i}(x), with\ \sigma_i = \beta \bar{d}_i, \tag{3.2}$$

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.

3.4 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. 3.1.

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. The statistics of σ will be saved for further reference.

Assuming that the average height and width for a category is H_i and W_i, where i

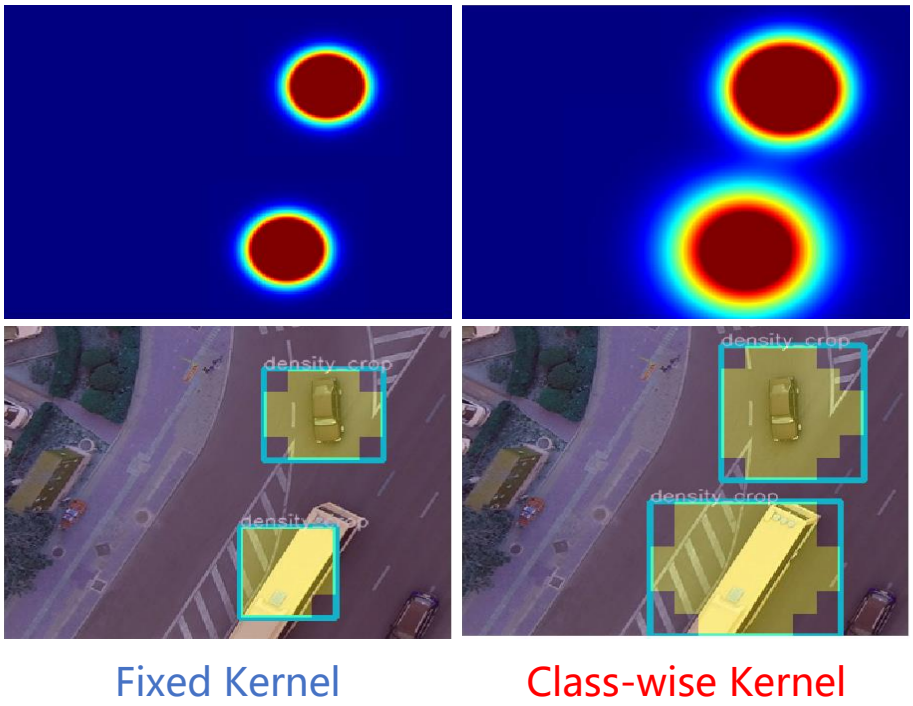

Figure 3.1: Visual comparison between fixed kernel and class-wise kernel. Left top is the density map for fixed σ. Left bottom is its corresponding cropping results. As can be observed, the bus is not fully covered by the light blue rectangle, which results in truncation. To resolve this issue, we replace 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.

is the current object category, we estimate σ by applying Eq. 3.3:

$$\sigma_i = \frac{1}{2}\sqrt{H_i^2 + W_i^2}. \tag{3.3}$$

We record those σ values for each category and apply them to Eq. 3.2 to generate density maps. In this case, we are able to accommodate the scale of medium and large objects 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. 3.1.

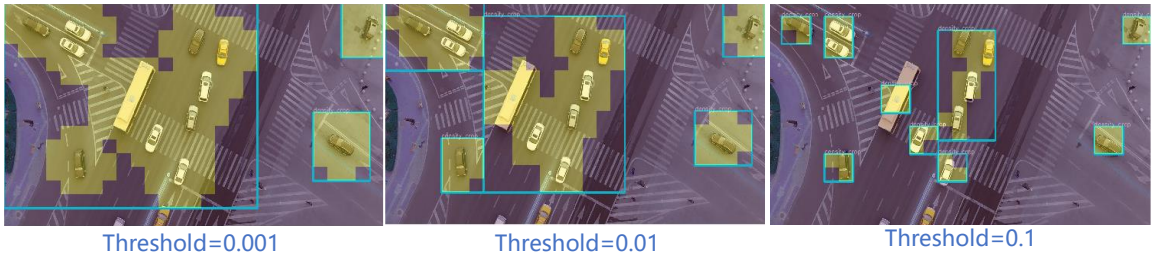

Figure 3.2: Visualization of density mask under different thresholds. As the threshold increases, the yellow region shrinks and one large region breaks into disconnected sub-regions. Yellow region is the candidate crop region and the light blue bounding box indicates the full region to crop.

3.5 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. 1.1, 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 intensities in the current window and compare the sum with the density threshold. If the sum 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 binary(Only 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 under a higher threshold. This leads to more crops with some only have a few objects. Fig. 3.2 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

3.6 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 are not desired. (2) The performance of detector may drop on low resolution crops as compared with higher resolution counterparts, as crops become really blurry after resizing to the original input size.

3.7 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 the setting in [3].

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.

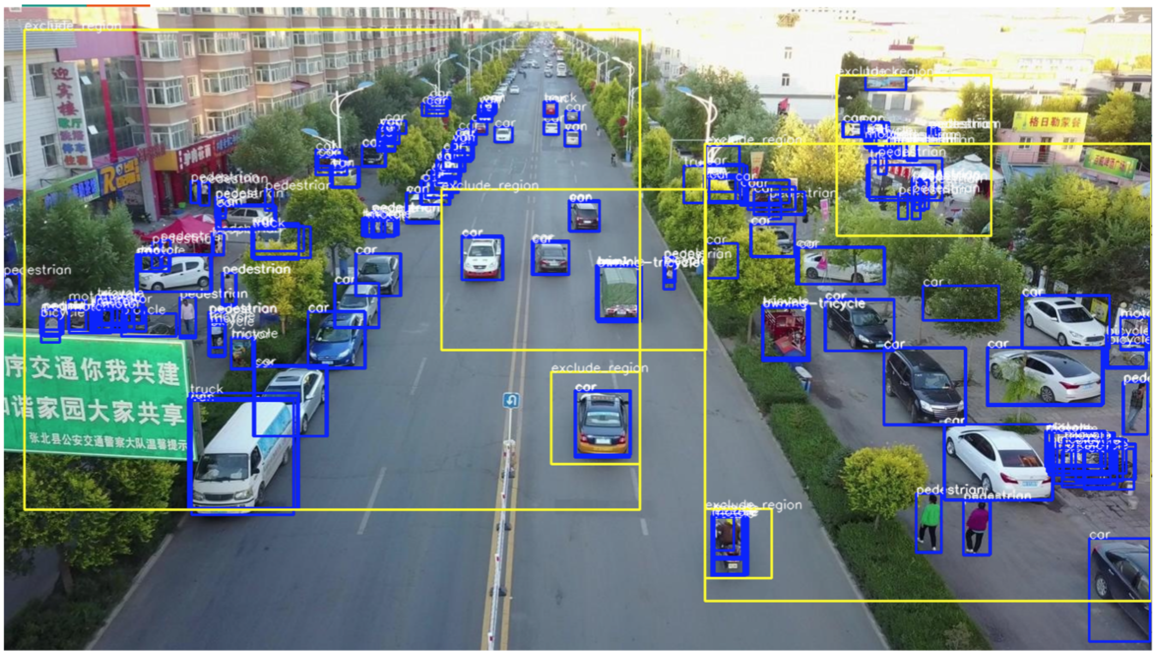

Figure 3.3: A visual example of the final detection result. The yellow rectangles represent regions of density crops. The blue rectangles represent ground-truth bounding boxes. The bounding boxes from both density crops and the whole images in inference stage are kept and labeled on the plot, as well as their corresponding categories. NMS is applied after obtaining the fusion bounding boxes. Thus we do not show it in this figure.

CHAPTER 4: EXPERIMENTS

4.1 Implementation details

Our implementation is based on the MMDetection toolbox [40]. The MCNN [28] 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 [10] 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 [3].

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 $600 \times 1{,}000$ on both datasets. We follow the similar setup in [3] 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 aerial image datasets, VisionDrone dataset (year of 2018) [1] and UAVDT dataset [2].

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 $\times$ 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 solution, we use the validation set to evaluate the performance, which is also the choice of existing works [3, 24].

UAVDT. UAVDT has a rich amount of images (23,258 training images and 15,069 test images) for aerial image 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 $\times$ 540 pixels.

4.3 Evaluation metric

For density map generation, pixel wise based evaluation metrics, such as mean absolute error and mean square error, turns out to be the fair choices for general evaluation purpose.

The equation for mean absolute error is defined in Eq. 4.1 and mean square error is defined in Eq. 4.2

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|x_i - \hat{x_i}| \tag{4.1}$$

$$MSE = \frac{1}{N}\sqrt{\sum_{i=1}^{N}(x_i - \hat{x_i})^2} \tag{4.2}$$

Where x_i is the ground truth of the pixel and $\hat{x_i}$ is the predicted value of the

Table 4.1: Quantitative result for UAVDT dataset.

Method	Backbone	#Image	AP	AP_{50}	AP_{75}	AP_{small}	AP_{mid}	AP_{large}
R-FCN [16]	ResNet 50	15096	7.0	17.5	3.9	4.4	14.7	12.1
SSD [6]	N/A	15096	9.3	21.4	6.7	7.1	17.1	12.0
RON [41]	N/A	15096	5.0	15.9	1.7	2.9	12.7	11.2
FRCNN [4]	VGG	15096	5.8	17.4	2.5	3.8	12.3	9.4
FRCNN [4]+FPN [42]	ResNet 50	15096	11.0	23.4	8.4	8.1	20.2	26.5
ClusDet [3]	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**

corresponding pixels.

For object detection, We follow the same evaluation metric as proposed in MS COCO [7]. 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 it as the primary metric to measure and compare the performance between the proposed method and other competing approaches.

In the 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 UAVDT datasets. Table 4.2 shows the results on VisionDrone. We can see that DMNet consistently outperforms ClusDet [3] 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 [3], 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 4.1 shows the results of different methods on UAVDT. 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 UAVDT. Particularly, DMNet consistently improves the accuracy on small scale, medium scale and large scale objects. This validates the effectiveness of our generated crops.

Inference speed. Here we report the inference speed for 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.

Table 4.2: 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 [3] 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 [3]	ResNet 50	Original+cluster	2716	26.7	50.6	24.7	17.6	38.9	51.4
DetecNet+CPNet+ScaleNet [3]	ResNet 101	Original+cluster	2716	26.7	50.4	25.2	17.2	39.3	54.9
DetecNet+CPNet+ScaleNet [3]	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

Figure 4.1: Visualization of our DMNet detection results on VisionDrone (first row) and UAVDT (second row).

Table 4.3: Ablation study on VisionDrone Dataset.

Method	AP	AP_{small}	AP_{mid}	AP_{large}
FRCNN [4]+FPN [42]	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

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 [28] as the density generation backbone and Faster RCNN [4] 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 intensities are larger than 0. From Table 4.3 we can clearly see that AP drops drastically without thresholding. From the post result analysis, we examine the generated crops and find most of them are large and cover many objects, which makes it difficult to generate small density crops. Since no threshold is applied, more background pixels are reserved, which further affects the performance of detector.

Comparison with uniform crops. As discussed in Chapter 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 4.2. 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 4.3, this method fails to beat DMNet, although it improves nearly 3 points on

Table 4.4: Performance of DMNet with strong backbone.

Detector	backbone	AP
Foveabox [21]	Res50	29.5
HRNet [43]	Cascade R-CNN(HRNetV2p-W32)	32.3
HRNet [43]	Cascade R-CNN(HRNetV2p-W40)	32.5
Libra-Rcnn [25]	Res50	30.9
GCNet [44]	Res101Xt	32.3

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 [3]. However, how it contributes to the final fusion detection remains unclear. Therefore, we additionally report performance of DMNet with only detection on image crops ($i.e$, without fusing the results of detection on the original whole images). The results are provided in Table 4.3. 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.

How much improvements can stronger detectors achieve. DMNet already achieves state of art detection performance on aerial image dataset. However, as the overall design is in a "plug-in manner", it is interesting to see how better will stronger detectors bring. A series of experiments are conducted to provide insights for this question.

As can be observed from the table 4.4, indeed the stronger model can contribute to more performance boost. As the backbone becomes stronger or model becomes

more sophisticated, the performance further goes up, which suggests the potential of DMNet.

CHAPTER 5: CONCLUSION

5.1 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.

5.2 Future research direction

Although DMNet achieves great detection performance, it still has large room to improve. And here I list some research directions for reference purpose.

- Replace Density map generation network with stronger backbone to see whether performance boosts. Better performance should be observed as the quality of density map greatly affects the performance of density crops, which in turn affects the performance of detectors.

- Data imbalance issue in aerial image detection. As being discussed in [1][13], one of the challenges is data imbalance issue in aerial image dataset. For DMNet, it has not considered to solve this issue from algorithm level. So this should be considered in future research.

- Further optimization for DMNet from the perspective of network architecture. It is highly desirable to combine density map generation and further detection together.

REFERENCES

[1] P. Zhu, L. Wen, X. Bian, H. Ling, and Q. Hu, "Vision Meets Drones: A Challenge," *arXiv e-prints*, p. arXiv:1804.07437, Apr 2018.

[2] 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," *arXiv e-prints*, p. arXiv:1804.00518, Mar. 2018.

[3] F. Yang, H. Fan, P. Chu, E. Blasch, and H. Ling, "Clustered object detection in aerial images," in *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[4] S. Ren, K. He, R. Girshick, and J. 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*, NIPSâ15, (Cambridge, MA, USA), p. 91â99, MIT Press, 2015.

[5] T. Lin, P. Goyal, R. Girshick, K. He, and P. DollÃ¡r, "Focal loss for dense object detection," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, Oct 2017.

[6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. 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.

[7] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft COCO: Common Objects in Context," *arXiv e-prints*, p. arXiv:1405.0312, May 2014.

[8] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *Int. J. Comput. Vision*, vol. 88, p. 303â338, June 2010.

[9] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3974–3983, 2018.

[10] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[11] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, "Perceptual Generative Adversarial Networks for Small Object Detection," *arXiv e-prints*, p. arXiv:1706.05274, Jun 2017.

[12] Y. Bai, Y. Zhang, M. Ding, and B. Ghanem, "SOD-MTGAN: Small Object Detection via Multi-Task Generative Adversarial Network," in *Computer Vision – ECCV 2018* (V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, eds.), (Cham), pp. 210–226, Springer International Publishing, 2018.

[13] P. Zhu, L. Wen, D. Du, X. Bian, H. Ling, Q. Hu, Q. Nie, H. Cheng, C. Liu, X. 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)*, pp. 0–0, 2018.

[14] F. Ozge Unel, B. O. Ozkalayci, and C. Cigla, "The power of tiling for small object detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[15] H. Xiong, H. Lu, C. Liu, L. Liu, Z. Cao, and C. Shen, "From Open Set to Closed Set: Counting Objects by Spatial Divide-and-Conquer," *arXiv e-prints*, p. arXiv:1908.06473, Aug. 2019.

[16] R. Girshick, "Fast R-CNN," *arXiv e-prints*, p. arXiv:1504.08083, Apr. 2015.

[17] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *arXiv e-prints*, p. arXiv:1703.06870, Mar. 2017.

[18] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv e-prints*, p. arXiv:1804.02767, Apr. 2018.

[19] H. Law and J. Deng, "CornerNet: Detecting Objects as Paired Keypoints," *arXiv e-prints*, p. arXiv:1808.01244, Aug. 2018.

[20] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "CenterNet: Keypoint Triplets for Object Detection," *arXiv e-prints*, p. arXiv:1904.08189, Apr. 2019.

[21] T. Kong, F. Sun, H. Liu, Y. Jiang, L. Li, and J. Shi, "FoveaBox: Beyound Anchor-Based Object Detection," *IEEE Transactions on Image Processing*, vol. 29, pp. 7389–7398, Jan. 2020.

[22] P. Zhang, Y. Zhong, and X. Li, "Slimyolov3: Narrower, faster and better for real-time uav applications," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[23] H. Wang, Z. Wang, M. Jia, A. Li, T. Feng, W. Zhang, and L. Jiao, "Spatial attention for multi-scale feature refinement for object detection," in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[24] J. Zhang, J. Huang, X. Chen, and D. Zhang, "How to fully exploit the abilities of aerial image detectors," in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[25] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra R-CNN: Towards Balanced Learning for Object Detection," *arXiv e-prints*, p. arXiv:1904.02701, Apr. 2019.

[26] 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*, vol. 16, no. 2, pp. 310–314, 2019.

[27] S. Zhang, G. He, H.-B. Chen, N. Jing, and Q. Wang, "Scale adaptive proposal network for object detection in remote sensing images," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 6, pp. 864–868, 2019.

[28] 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)*, pp. 589–597, June 2016.

[29] Y. Li, X. Zhang, and D. Chen, "CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes," *arXiv e-prints*, p. arXiv:1802.10062, Feb. 2018.

[30] C. Xu, K. Qiu, J. Fu, S. Bai, Y. Xu, and X. Bai, "Learn to Scale: Generating Multipolar Normalized Density Maps for Crowd Counting," *arXiv e-prints*, p. arXiv:1907.12428, July 2019.

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

[32] 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)*, pp. 1–6, July 2016.

[33] Y. Liu, G. Song, Y. Zang, Y. Gao, E. Xie, J. Yan, C. C. Loy, and X. Wang, "1st place solutions for openimage2019 – object detection and instance segmentation," 2020.

[34] D. Dwibedi, I. Misra, and M. Hebert, "Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection," *arXiv e-prints*, p. arXiv:1708.01642, Aug. 2017.

[35] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," 2020.

[36] S. Hong, S. Kang, and D. Cho, "Patch-level augmentation for object detection in aerial images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[37] D. Du, P. Zhu, L. Wen, X. Bian, H. Lin, Q. Hu, T. Peng, J. Zheng, X. Wang, Y. Zhang, L. Bo, H. Shi, R. Zhu, A. Kumar, A. Li, A. Zinollayev, A. Askergaliyev, A. Schumann, B. Mao, B. Lee, C. Liu, C. Chen, C. Pan, C. Huo, D. Yu, D. Cong, D. Zeng, D. Reddy Pailla, D. Li, D. Wang, D. Cho, D. Zhang, F. Bai, G. Jose, G. Gao, G. Liu, H. Xiong, H. Qi, H. Wang, H. Qiu, H. Li, H. Lu, I. Kim, J. Kim, J. Shen, J. Lee, J. Ge, J. Xu, J. Zhou, J. Meier, J. Won Choi, J. Hu, J. Zhang, J. Huang, K. Huang, K. Wang, L. Sommer, L. Jin, and L. Zhang, "Visdrone-det2019: The vision meets drone object detection in image challenge results," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[38] C. Chen, Y. Zhang, Q. Lv, S. Wei, X. Wang, X. Sun, and J. Dong, "Rrnet: A hybrid detector for object detection in drone-captured images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[39] R. Bahmanyar, E. Vig, and P. Reinartz, "MRCNet: Crowd Counting and Density Map Estimation in Aerial and Ground Imagery," *arXiv e-prints*, p. arXiv:1909.12743, Sept. 2019.

[40] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, *et al.*, "Mmdetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.

[41] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, and Y. Chen, "RON: Reverse Connection with Objectness Prior Networks for Object Detection," *arXiv e-prints*, p. arXiv:1707.01691, July 2017.

[42] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," *arXiv e-prints*, p. arXiv:1612.03144, Dec. 2016.

[43] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, "Deep High-Resolution Representation Learning for Visual Recognition," *arXiv e-prints*, p. arXiv:1908.07919, Aug. 2019.

[44] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "GCNet: Non-local Networks Meet Squeeze-Excitation Networks and Beyond," *arXiv e-prints*, p. arXiv:1904.11492, Apr. 2019.