Coarse-grained Density Map Guided Object Detection in Aerial Images

Chengzhen Duan Zhiwei Wei Chi Zhang Siying Qu Hongpeng Wang *
School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen)
{19S051024,18S151541}@stu.hit.edu.cn; wanghp@hit.edu.cn

Abstract

Object detection in aerial images is challenging for at least two reasons: (1) most objects are small scale relative to high resolution aerial images; and (2) the object position distribution is nonuniform, making the detection inefficient. In this paper, a novel network, the coarse-grained density map network (CDMNet), is proposed to address these problems. Specifically, we format density maps into coarse-grained form and design a lightweight dual task density estimation network. The coarse-grained density map can not only describe the distribution of objects, but also cluster objects, quantify scale and reduce computing. In addition, we propose a cluster region generation algorithm guided by density maps to crop input images into multiple subregions, denoted clusters, where the objects are adjusted in a reasonable scale. Besides, we improved mosaic data augmentation to relieve foreground-background and category imbalance problems during detector training. Evaluated on two popular aerial datasets, VisDrone[29] and UAVDT[6], CDMNet has achieved significant accuracy improvement compared with previous state-of-the-art methods.

1. Introduction

Object detection is a fundamental problem in computer vision, and is widely applied in many fields, such as disaster search and traffic surveillance. Recently, object detection models based on deep learning have achieved great success. (e.g. Faster RCNN [21], YOLO [20], SSD [18]) on natural image datasets (e.g., MS COCO [17], Pascal VOC [7]). However, they always generate inferior detection results in aerial images.

Aerial images are usually captured by drones, airplanes or satellites from a top view, therefore they are different from images taken on the ground. There are several special challenges for aerial image detection. (1) Most objects are small scale relative to high resolution aerial images (e.g. 2000x1500 in VisDrone). Detectors are hard to distinguish small objects from the surrounding background when inference is performed in the case of limited resolution. (2) The distribution of object positions is nonuniform in aerial images. For example, objects always appear on the streets but rarely in other regions, such as the sky. It is inefficient and meaningless to detect regions without objects.

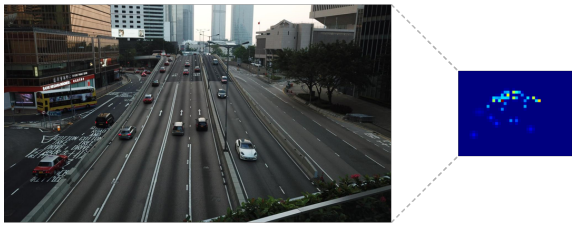

Figure 1. The density map estimation model inputs a low resolution image (640 × 480) and outputs a high down sampling rate coarse-grained density map. The advantages of the coarse-grained format include: (1) clustering the objects; (2) quantifying scale and reducing geometric distortion for density prediction; (3) reducing model complexity and saving computing.

Some methods are proposed based on image cropping strategies [13, 19, 26, 27] to solve the above problems. These methods first crop images through specific schemes, dropping out many background pixels, then leverage object detection on each cropped block, and finally merge the detection results. The reason why the cropping strategy is effective is that the area proportion of objects in the image is increased by the cropping method, thereby upsampling small objects. The methods based on image cropping have become mainstream solutions for aerial image detection. However, there still exist some problems waiting to be solved, including low efficiency method generating too many crops, unreasonable object scale in the cropped block, and foreground-background and category imbalance.

Inspired by the phenomenon of objects gathering in local regions in aerial images, we proposed a coarse-grained density map guidance network (CDMNet). Specifically, a concept of coarse-grained density map is proposed to cluster objects and describe the object distribution of aerial images as shown in Fig. 1. We designed a lightweight dual

*corresponding author.

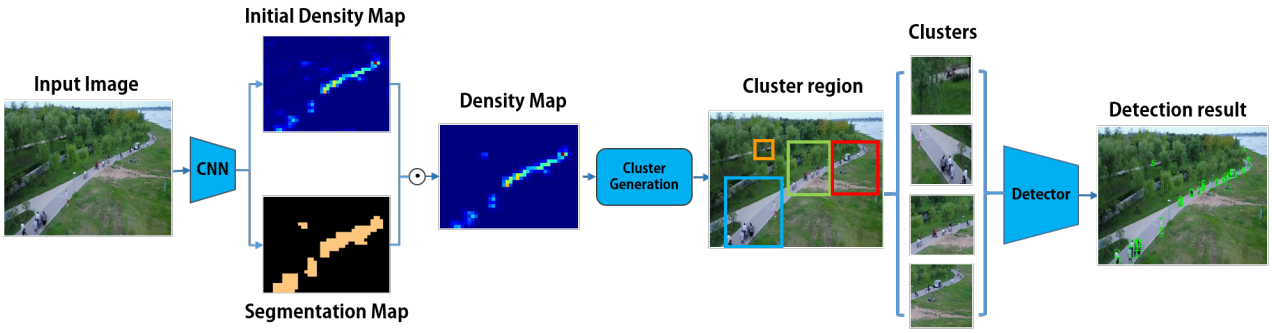

Figure 2. An Overview of the CDMNet framework. CDMNet mainly consists of three components: (1) coarse-grained density estimation subnet; (2) cluster region generation module; (3) object detection network. The coarse-grained density estimation subnet predicts the density map and segmentation mask of the image. $\odot$ represents the and operation. The cluster region generation module generates initial clusters by density connected regions and adjusts clusters based on the object's relative scale information. Finally, all cluster regions are fed into the detector, and the outputs are merged by non-maximum suppression (NMS) into the final detection result.

task network to efficiently generate coarse-grained density maps. Furthermore, we propose an object cluster region generation algorithm guided by density maps. We leverage density connected regions and extract object scale information from density maps to adjust the proportion of the object area in cluster regions. In the detector training stage, we improved the mosaic data augmentation method of YOLOv4 [1] to alleviate the problems of foreground-background and category imbalance.

For each image, the process of our method can be divided into three stages. First, the coarse-grained density map of the image is predicted by the density estimation model. Second, we generate initial cluster regions through density connected regions, then estimate the proportion of the object area in cluster regions and adjust cluster regions by splitting or enlarging operations. Finally, all cluster regions are detected and merged through non-maximum suppression (NMS).

Compared with DMNet [13] exploiting density maps to describe object distribution, our work focuses on extracting object scale information from density maps. The method of density map generation is different in some ways, including an extra segmentation branch helping to locate objects, an object-wise scheme to adjust the gaussian kernel in density map ground truth generation and coarse-grained density maps to cluster objects, quantify scale and save computing.

In summary, the paper has the following contributions:

(1) A coarse-grained density map concept and a lightweight dual head density estimation network are proposed. The coarse-grained density map can cluster objects, quantify scale and save computing cost. An extra segmentation branch in the network helps to locate the object on density maps more accurately.

(2) A cluster region generation algorithm guided by coarse-grained density maps is proposed. We explored the physical meaning of the elements in density maps to estimate coarse scale information, and proposed a cluster region adjustment algorithm to normalize objects into a reasonable scale range.

(3) The mosaic data augmentation method of YOLOv4 [1] is improved to focus on rare appeared and hare detected objects, which alleviates the problems of foreground-background and category imbalance.

(4) Evaluated on two aerial datasets: VisDrone [29] and UAVDT [6], CDMNet achieves state-of-the-art performance only by leveraging about 10 % pixels of the original dataset for testing.

2. Related work

2.1. Object detection in natural scenes

The mainstream detectors based on deep learning can be divided into region-based detectors and region-free detectors. R-CNN [9] is the earliest detection model based on candidate regions, which leverages the selective search algorithm to extract candidate regions where objects may exist. Faster R-CNN [21] proposed a region proposal network (RPN) to replace the selective search algorithm, and assumes one object in each proposal. Region-based detectors have achieved great success in detection accuracy, but the efficiency is not satisfactory. Region-free detectors abandon candidate region generation and directly perform feature extraction on the image to predict category probability and bounding box coordinates, thereby greatly improving the detection efficiency. The representative models include YOLO [20], SSD [18], RetinaNet [17] and FCOS [24].

2.2. Object detection in aerial images

Aerial image detection algorithms usually employ cropping strategies. Detectors first crop high resolution images

into several subregions, denoted as cluster regions, and detect them. The final results are fused by the detection of cluster regions and original images. In [19], the authors split images uniformly and show the power of cropping strategies in small object detection. [8] proposed a coarse-to-fine method. Rough detection is performed through R-Net, and then the reinforcement learning network Q-net generates some fixed potential regions for fine detection. ClusDet [26] and DMNet [13] are both three-stage detectors. In the first two stages, ClusDet first uses CPNet to generate the object cluster regions, then ScaleNet predicts scale information to adjust the cluster region. DMNet first predicts the density map and then introduces sliding windows to determine whether the region contains objects according to the density value in windows. DREN [27] strengthens the detection of difficult regions. [25] exploits a clustering algorithm to generate initial difficult regions based on detection results on the whole images.

2.3. Data augmentation

Data augmentation is one of the easiest approaches to improving model performance, including random image flipping, rotating, and cropping. Perceptual GAN [14] leveraged a confrontation generation network to generate super-resolution images of small objects, thereby improving the detection ability of small objects. RRNet [2] uses semantic segmentation to predict the road region, and pastes the object on the road region to increase the diversity of the object. YOLOv4 [1] proposed the mosaic augmentation method. Mosaic refers to the combination of four subregions from different images after flipping, scaling, and color gamut changing, which increases the background complexity of the generated images.

Different from random combining subregions of the original mosaic method, we discard some subregions where objects are in extreme scales and prefer to stitch the subregions which contain rare appeared and difficult detected objects into new images.

3. Method

3.1. Density map estimation

3.1.1 Coarse-grained density estimation network

Density estimation is widely studied in crowd counting tasks. For crowd counting algorithms based on deep learning (e.g., MCNN [28], CSRNet [15]), the interested objects of images are presented in the form of density maps. Unlike the density estimation of crowd counting attention on fine-grained information, object detection based on clusters focuses on coarse information of object distribution and object numbers in local regions. The density estimation module can suffer from more lower resolution aerial images as input and more higher down sampling rate (16×) density maps as output. We denote this type of density map as the coarse-grained density map. Formulating the density map into the coarse-grained format not only clusters the objects, but also reduces the geometric distortion of the objects, which effectively improves the accuracy of density estimation. In addition, removing the upsampling layer in the density map estimation model can reduce the time cost.

Our proposed density estimation network structure is shown in Fig.2. The backbone uses MobileNetv2 [22], which leverages deep separable convolution to reduce model time costs. In order to distinguish the background and the foreground more accurately, two heads follow the backbone: one predicting the density map, and the other predicting the segmentation map. Every element in the density maps is able to map a definite 16×16 region of the input image. The value of density map elements has a physical meaning that the number of objects distributed in the corresponding region of the element. The segmentation maps are responsible for generating background and foreground masks. In the inference stage, we only retain the value of density maps in the foreground mask to guide cluster region generation. Unlike the sliding windows and threshold filter method in DMNet[13], an extra segmentation branch can help to binarize density maps, cutting noise and error density value prediction, which is parameter free and robust.

The loss function of coarse-grained density estimation network includes the density map loss and the segmentation loss as $L = L_{des} + L_{seg}$. The loss of density map is based on pixel-wise mean absolute error, which is given as below:

$$L_{des} = \frac{1}{2N}\sum_{i=1}^{N}||D(X_i) - D_{gt}(X_i)||^2 \tag{1}$$

where N is the number of images in training batches. X_i is the input image and D_{gt} is the ground truth density map. D stands for the generated density map by the density subnet.

Segmentation loss L_{seg} adopts binary cross entropy loss. In the training stage, we reweigh the loss of positive and negative samples to obtain better masks.

$$L_{seg} = -\frac{1}{N}\sum_{i=1}^{N}\lambda_1 p_i \log(\hat{p_i}) + \lambda_2(1-p_i)\log(1-\hat{p_i})) \tag{2}$$

where N is the pixel number of the mask. λ_1 and λ_2 are the weights of positive samples and negative samples. p_i is the value of pixel i in the ground truth mask. $\hat{p_i}$ is the predicted logic of the pixel i. We will discuss the set of λ_1 and λ_2 in section 4.4.

3.1.2 Ground truth map generation

The coarse-grained density estimation model is supervised training. In order to generate ground truth density maps,

we follow the similar method of generating density maps in MCNN [28]. First, we map objects to their corresponding positions on density maps. Assuming the coordinate of the object center point is (x_c, y_c) in input images, the corresponding position of density maps is calculated as Eq. 3.

$$x_i, y_i = \frac{x_c}{s}, \frac{y_c}{s} \tag{3}$$

where s is the down sampling factor of density maps relative to input images. We use a gaussian kernel which is normalized to 1 to describe the object distribution on density maps. The ground truth density maps can be generated by accumulating the gaussian weight of all objects as the Eq. 4.

$$D(x,y) = \sum_{i=1}^{N} G_{\sigma_i}(x - x_i, y - y_i), \quad \sigma_i = (\beta h_i, \beta w_i) \tag{4}$$

where N is the object number of images. G is the Gaussian kernel. h_i and w_i are the height and width of the object i on density maps. We configured β to be 0.15 in our experiment. We adopt an object-wise scheme to adjust the variance of gaussian kernel rather than the class-wise scheme used in DMNet[13]. We consider that the object-wise scheme reflects true instance scale information in density maps, which will create a positive compact in the next stage of scale estimation.

The ground truth segmentation map can be generated by simply binarization of density maps as the Eq. 5.

$$S(x,y) = \begin{cases} 1 & D(x,y) > 0 \\ 0 & D(x,y) = 0 \end{cases} \tag{5}$$

3.2. Cluster region generation

The generation method of cluster regions is critical for object detection in the next stage. We need to consider these issues. (1) All interested objects ought to be contained in cluster regions. (2) The number of cluster regions should be as fewer as possible to promote detection efficiency. (3) Objects should be in reasonable scale relative to cluster regions to improve detection accuracy. In the paper, we leverage object position and scale information from the density map to guide cluster region generation.

Specifically, density maps will be updated by bitwise and operation with foreground mask to remove some error prediction, then the minimum bounding boxes of density connected regions are leveraged to generate some initial cluster regions as shown in Fig. 3(a). In order to avoid generating too many cluster regions, we use the closing operation in morphology to eliminate some narrow discontinuities. As shown in Fig. 3(b), this operation effectively reduces the number of cluster regions. Furthermore, we extract the object scale information from density maps to finely adjust

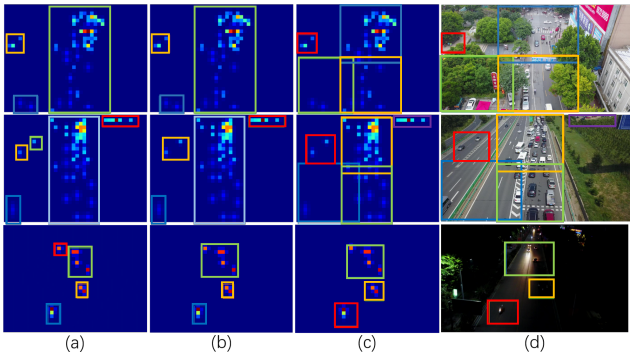

Figure 3. (a) the cluster regions obtained through the density connected regions; (b) the cluster regions obtained by morphological closing operation and connected regions (c) the cluster regions based on b with scale adjustment; (d) cluster regions on the original images. After the above steps, the number of cluster regions is reduced, and objects are normalized in a reasonable scale.

cluster regions, reducing the extreme object scale in the cluster regions as shown in Fig. 3(c).

We take advantage of two features, the summation of density values and the number of elements whose density value is not equal to zero, to roughly estimate object scales in cluster regions as Eq. 6. The summation of density values represents the number of objects. The density map element whose density value is not equal to zero means that its corresponding region in the input image is covered by objects.

$$Obj_{avg} = \frac{\sum_{i,j} I(D_{i,j}) \times s}{\sum_{i,j} D_{i,j}}, \quad I(X) = \begin{cases} 1 & X > 0 \\ 0 & X = 0 \end{cases} \tag{6}$$

where $D_{i,j}$ is the density value of elements whose coordinates are i, j on the density map. I is indicator function. s is the down sampling factor of the density map relative to input images.

Furthermore, we chose LightGBM [11] with two extra features, the area of cluster regions and the area of original images, to predict the proportion factor between object area and cluster area. The factor represents relative scale of objects in clusters. The desired ratio of object area relative to cluster after adjustment is denoted as α. The zoom factor of the cluster region w can be calculated as the Eq. 7.

$$P = \frac{S_{obj}}{S_{chip}}, \quad w = \frac{P}{\alpha} \tag{7}$$

where S_{obj} is the average area of objects, S_{chip} is the area of clusters. P is the area ratio factor which is predicted by the model in the inference stage. Then the cluster region is

Algorithm 1 Enlarge and Split the Cluster Region

Input :
B : the vertical bounding box of the cluster area
W : the predicted value of the regression model
β γ : the threshold of zoom factor
H_* W_* C_*: the hight/width/center of *

Output :
$B^{'}$: the adjusted box

if $W < \beta$ **then**
  $B^{'} = splitFourPart(B)$
else if $W < \gamma$ **then**
  $B^{'} = splitTwoPart(B)$
else if $W > 1$ **then**
  $A = W \times Area(B)$
  $C_{B'} = C_B$
  if $H_B > \sqrt{A}$ **then**
    $H_{B'} = H_B, W_{B'} = A/H_B$
  else if $W_B > \sqrt{A}$ **then**
    $W_{B'} = W_B, H_{B'} = A/W_B$
  else
    $H_{B'} = \sqrt{A}, W_{B'} = \sqrt{A}$
  end if
end if
return $B^{'}$;

adjusted according to the zoom factor to reduce the extreme object scale in the cluster region.

The detailed implementation is illustrated in Algorithm 1. For each cluster region, we compare the zoom factor with the threshold. If W is smaller than 1, we divide the cluster region into four parts or two parts (along the long side) equally. If W is greater than 1, we enlarge cluster regions, making it close to a square. β and γ are configured as 0.3 and 0.6 in our experiment. When W is 0.5, clusters area requires to become half. We set γ litter than 0.5 due to existing overlap when splitting clusters into two parts. The reason is the same as setting β.

3.3. Object detection

In the cluster region generation stage, most of the background pixels are discarded and the background elements in training samples are insufficient, resulting in many false-positive results. SNIPER [23] solves this problem by adding regions with high false detection rates to the training set, and improves the model's ability to discriminate complex backgrounds. Although it can solve the problem, the operation is complicated and time-consuming.

We refer to the mosaic data augmentation method in YOLOv4 [1] to disrupt the semantic features of the image through image splicing, and enhance the model's detection ability for objects under complex backgrounds. As shown in Fig. 4, four sliding windows of different scales

Figure 4. The improved mosaic method includes: (1) generating subregions through sliding windows of different scales; (2) discarding some subregions where objects are in extreme scale; and (3) stitching subregions containing more objects and covering the rare appeared and difficult detected categories into mosaic images.

are employed to generate some candidate subregions. In the combination stage, different from the random choosing subregions of the original mosaic method, we discard some subregions where objects are in extreme scales and prefer to stitch the subregions which contain more objects and cover rare appeared and difficult detected objects into images. The improved mosaic augmentation method not only enhances the complexity of the background, but also strengthens some objects from the rare and difficult categories, alleviating the problems of foreground-background and category imbalance. We simply regard categories which detection performance is below average performance as difficult categories.

4. Experiments

4.1. Implementation details

We implement the coarse-grained density estimation model on Pytorch. The backbone is MobileNetv2 [22] pretrained on ImageNet [5]. The density and segmentation head is implemented with a 3×3 convolutional layer, followed by a SElayer[10] and a 1×1 convolutional layer to predict maps. The output channel of the segmentation head is 2, which represents the probability of the background and foreground. The output channal is 1 in density head. The model is trained for 50 epoches using Stochastic Gradient Descent (SGD) with a weight decay of 0.0005 and momentum of 0.9. We set the base learning rate to 0.01 and decay by a factor of 10 at epoch 35 and 45. The input resolution of the density estimation model is 640×480. We all trained

Table 1. The ablation study on VisDrone [29] dataset. "Original" means the original verification set. "Cluster" represents the cropped cluster region, and different models adopt different cropping strategies. The #img is the number of images detected by the detector. Small, medium and large are represented by 's', 'm', and 'l' respectively. ⋆ represents mosaic expanded training sets.

Methods	backbone	test data	$\#img$	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
FRCNN[21]+FPN[16]	ResNet50	Original	548	21.4	40.7	19.9	11.7	33.9	54.7
FRCNN[21]+FPN[16]	ResNet101	Original	548	21.4	40.7	20.3	11.6	33.9	54.9
FRCNN[21]+FPN[16]	ResNeXt101	Original	548	21.8	41.8	20.1	11.9	34.8	55.5
ClusDet[26]	ResNet50	Original+cluster	2,716	26.7	50.6	24.7	17.6	38.9	51.4
ClusDet[26]	ResNet101	Original+cluster	2,716	26.7	50.4	25.2	17.2	39.3	54.9
ClusDet[26]	ResNeXt101	Original+cluster	2,716	28.4	53.2	26.4	19.1	40.8	54.4
DMNet[13]	ResNet50	Original+cluster	2736	28.2	47.6	28.9	19.9	39.6	55.8
DMNet[13]	ResNet101	Original+cluster	2736	28.5	48.1	29.4	20.0	39.7	**57.1**
DMNet[13]	ResNeXt101	Original+cluster	2736	29.4	49.3	30.6	21.6	41.0	56.9
CDMNet	ResNet50	cluster	2170	29.2	49.5	29.8	20.8	40.7	41.6
CDMNet	ResNet101	cluster	2170	29.7	50.0	30.9	21.2	41.8	42.9
CDMNet	ResNeXt101	cluster	2170	30.7	51.3	32.0	22.2	42.4	44.7
CDMNet⋆	ResNeXt101	cluster	2170	**31.9**	**52.9**	**33.2**	**23.8**	**43.4**	45.1

Table 2. Quantitative result for UAVDT [6] dataset. ⋆ represents mosaic expanded training sets.

Methods	backbone	$\#img$	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
R-FCN[4]	ResNet50	15096	7.0	17.5	3.9	4.4	14.7	12.1
SSD[18]	N/A	15096	9.3	21.4	6.7	7.1	17.1	12.0
RON[12]	N/A	15096	5.0	15.9	1.7	2.9	12.7	11.2
FRCNN[21]	VGG	15096	5.8	17.4	2.5	3.8	12.3	9.4
FRCNN[21]+FPN[16]	ResNet50	15096	11.0	23.4	8.4	8.1	20.2	26.5
ClusDet[26]	ResNet50	25427	13.7	26.5	12.5	9.1	25.1	31.2
DMNet[13]	ResNet50	32764	14.7	24.6	16.3	9.3	26.2	**35.2**
CDMNet	ResNet50	37522	16.8	29.1	18.5	11.9	29.0	15.7
CDMNet⋆	ResNet50	37522	**20.7**	**35.5**	**22.4**	**13.9**	**33.5**	19.8

on two RTX 2080Ti GPUs. For cluster regions generation, we configure the desired factor α as 0.032 and 0.006 on VisDrone [29] and UAVDT [6] respectively. For object detector, Faster-RCNN[9] implemented based on MMDetection [3] is used as the base detector. The input resolution is set to 1000×600 pixels on two datasets. The detector uses SGD to train 12 and 6 epoches on VisDrone [29] and UAVDT [6] respectively. The initial learning rate is 0.01, decaying weight at 8 and 11 epoches on VisDrone[29], 4 and 5 epoches on UAVDT[6]. The maximum detection number is set to 500.

4.2. Datasets and evaluation metrics

We evaluate our approach on two public aerial image datasets: VisDrone [29] and UAVDT [6]. Next, we briefly introduce the datasets and the evaluation metric below:

VisDrone. The dataset contains 10,209 images (6,471 for training, 548 for validation and 3,190 for testing) captured by drone platforms in different places at different heights. The image resolution is between 640×950 and 1080×1920. There are over 380k annotated object instances from 10 categories in the dataset. The proportion of small objects is 89%. As VisDrone has all the characteristics of aerial images, it is one of the best benchmarks for verifying our method of detection performance. The same as existing works [26, 27, 13], we evaluate the performance of our method on the validation set.

UAVDT. The dataset comes from 50 videos taken by drones (23258 for training, 15069 for testing). The resolution of the image is 540×1080 pixels and includes 3 categories: cars, trucks and buses. The object scale distribution is similar to VisDrone.

In the density estimation task, we use mIOU and MAE to measure the performance of the segmentation task and the density estimation task, respectively. At the same time, we use a rough object recall rate to evaluate the impact of the density estimation task on the subsequent detection stage. The recall is calculated by the ratio of the number of objects covered by the density connection region to the total number of objects. In object detection tasks, based on COCO style [17] Average Precision (AP) metrics, we use $AP, AP_{50}, AP_{75}, AP_{small}, AP_{medium}$ and AP_{large} as the metrics to measure precision.

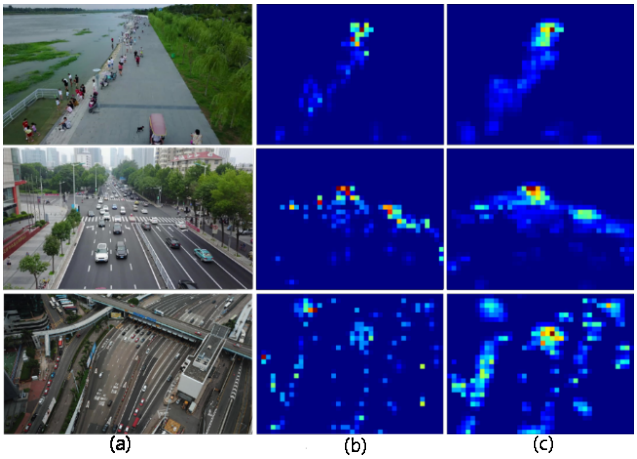

Figure 5. Visualization of density maps: (a) the input images. (b) the ground truth of density maps. (c) the predicted density maps.

4.3. Quantitative result

VisDrone. The performance comparison is shown in Table 1. It can be observed that CDMNet consistently exceeds previous methods on three different backbone networks. Specifically, CDMNet achieves the state-of-the-art performance of 30.7 AP with the ResNetXt101 backbone network. It is noted that CDMNet has significant advantages on AP_s and AP_m under different backbones. CDMNet improves about 1 point performance of the small and middle scale objects respectively compared with DMNet[13]. However, compared with the other methods, CDMNet has relatively low performance in the detection of large scale objects. The reason is that CDMNet does not add detection results of the original whole image.

UAVDT. Table 2 shows the experimental results on the UAVDT dataset. Compared with the previous method, CDMNet has significant performance improvement. Specifically, compared to DMNet [13], CDMNet improves AP_s and AP_m by 2.6 points and 2.8 points respectively. This validates the effectiveness of scale adjustment on cluster regions. In addition, mosaic augmentation has a great increment on detection performance. We conjecture that the mosaic helps to solve the background similarity problem in UAVDT.

4.4. Ablation study

Table 3. Binarization method comparison.

method	mIOU	Recall	#img	AP
threshold filter	61.94	**99.40**	2651	28.7
segmentation branch	**80.13**	98.97	**2170**	**29.2**

Effect of Segmentation Branch. We compare different binarization methods in Table 3 on the VisDrone validation dataset. The first row exploits the slid window method and filters regions where the density value is below a certain threshold. The second row exploits and operation for the density map and segmentation map. It is noted that the recall of two methods is high, which means almost all interested objects are described by coarse density maps. However, the AP and mIOU have a large margin (28.7 vs 29.2, 61.94 vs 80.13), which means the segmentation branch can help to locate objects, cutting out some noise and error density prediction, generating more accurate density maps. In addition, the accurate density map reduces the number of cluster regions, which improves detection efficiency.

Table 4. Different segmentation loss weight comparison.

weight	mIOU	MAE	Recall	AP
1:1	**82.4**	**13.8**	97.95	28.5
7:1	80.13	14.9	**98.97**	**29.2**

Table 5. Different ground truth scheme comparison.

method	Recall	#img	AP
class-wise	**99.10**	2457	28.9
object-wise	98.97	**2170**	**29.2**

Effect of Segmentation Loss Weight. There are some imbalance problems in the segmentation subtask. As shown in Table 4, recall rate can be improved by strengthening the loss weight of positive samples. The loss weight of 7:1 is calculated by the logarithmic function of the ratio between object pixel number and background pixel number. It can be observed by increasing the loss of positive samples, recall and detection performance can be improved.

Effect of Object-wise Scheme. There are two different schemes for setting the gaussian variance in density map ground truth generation. As shown in Table 5, the AP of object-wise is slightly higher than that of class-wise. The reason is compared with the class-wise scheme, the object-wise scheme really reflects the scale of instances in density maps. It is helpful for scale adjustment to normalize object scale and create fewer clusters (2457 vs 2170).

Effect of Density Map Estimation. We visualize the ground truth density map and estimated density map as shown in Fig. 5. It can be observed that the contour of the object distribution in the predicted density map is basically the same as that in the real density map. For the density value, the obvious problem is that the predicted value is more dense than the real value in some regions. The main reason is that the ground truth label ignores some objects with small scale or fuzzy pixels, but images still contain these objects, resulting in abnormal density values predicted by the network.

Effect of Cluster Region Generation. In the inference stage, we try different ways to generate cluster re-

Table 6. The effect of cluster region generation on the detection performance of the VisDrone validation dataset. "Original" represents the original image. "Tiling" representsthe image is cropped into 6 pieces of equal size. "CR" represents generating clipping blocks directly according to the object connected region of the density map. "CO" represents using closed operations to process the density map. "SA" represents the scale adjustment in cropped blocks.

Original	Tiling	CR	CO	SA	$\#img$	AP	AP_s	AP_m	AP_l
✓					548	19.4	9.4	33.7	**54.0**
	✓				3288	28.2	**22.1**	37.1	37.4
		✓			1895	21.8	13.2	34.0	34.4
		✓	✓		1295	21.6	12.3	34.6	39.8
		✓		✓	2535	**29.3**	21.0	40.4	39.9
		✓	✓	✓	2170	29.2	20.8	40.6	41.6
✓		✓	✓	✓	2718	**29.3**	20.5	**40.9**	51.8

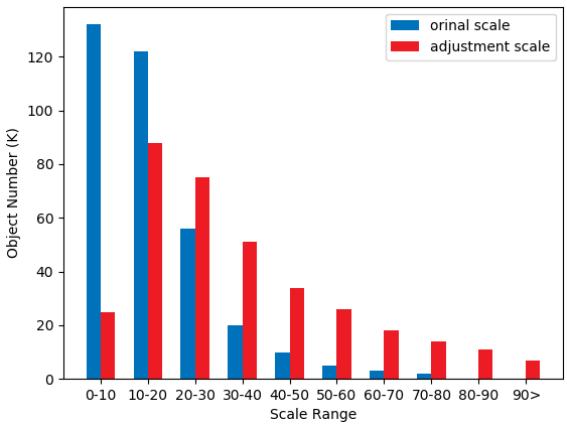

Figure 6. The variation of object scale distribution after employing scale adjustment operation. All scales are measured at 640 x 480 resolution.

gions, and the experimental results are shown in Table 6. The performance is poor when directly detecting original images. This is because most objects are small compared to the original images, and the detection performance of small objects is very poor (AP_s=9.4). We use a uniform cropping strategy to obtain a higher accuracy (AP=28.2), but the number of detection images increases, and the performance of large scale objects decreases significantly due to the truncation of large objects.

Compared to direct cropping density connected regions, adding closed operation can effectively reduce the number of cluster regions (1895 vs 1295) with only sacrifice 0.2 AP. Remarkably, the scale adjustment effectively improves the detection accuracy from 21.8 to 29.3. The reason is the method balances the proportion of object area relative to cluster region, normalizing object scale in a reasonable range. We count the scale distribution of the object before and after the scale adjustment operation as shown in Fig. 6. Most small scale objects are normalized to other scales.

However, the detection accuracy of large objects is low.

Table 7. Vanilla and improved mosaic methods comparison.

method	AP	AP_s	AP_m	AP_l
vanilla mosaic	30.0	22.1	41.2	**43.8**
improved mosaic	**30.6**	**22.8**	**41.7**	36.8

The reason is that large objects are hard to appear in the cluster region as a whole. It can be noticed that after we add the original image for detection, the detection performance of large objects increases remarkably, but the AP only increases by 0.1. The reason is that the VisDrone[29] dataset contains a small number of large objects.

Effect of Mosaic. We compare the improved mosaic method and vanilla mosaic augmentation in Tbale 7. The improved method has overall increase in small and middle scale objects. During the training of VisDrone [29] and UAVDT [6], we added 10000 mosaic images. It can be seen from table 1 and table 2, the detection accuracy has been greatly improved by using the improved mosaic augmentation method. It is worth noting mosaic augmentation increases large margin performance on UAVDT. The reason is that the UAVDT training set comes from 30 videos, and images have similar scene information because they come from a series of adjacent video frames with small differences. The mosaic method, combining subregions and creating complicated images, is suitable to relieve the foreground-background imbalance problem.

5. Conclusion

We propose a coarse-grain density map network (CDMNet) to solve small object and nonuniform distribution problems in aerial image detection. The coarse-grained density map is predicted by the lightweight estimation network. Then cluster regions are generated based on density maps. Finally, the detector, which trained by the improved mosaic augmentation method, detects all cluster regions and merges them. Experiments show that our method achieves significant accuracy improvements on two aerial datasets.

References

[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 2, 3, 5

[2] Changrui Chen, Yu Zhang, Qingxuan Lv, Shuo Wei, Xiaorui Wang, Xin Sun, and Junyu Dong. Rrnet: A hybrid detector for object detection in drone-captured images. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019. 3

[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. 6

[4] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016. 6

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[6] 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. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 370–386, 2018. 1, 2, 6, 8

[7] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 1

[8] Mingfei Gao, Ruichi Yu, Ang Li, Vlad I Morariu, and Larry S Davis. Dynamic zoom-in network for fast object detection in large images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6926–6935, 2018. 3

[9] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 2, 6

[10] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 5

[11] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems*, pages 3146–3154, 2017. 4

[12] Tao Kong, Fuchun Sun, Anbang Yao, Huaping Liu, Ming Lu, and Yurong Chen. Ron: Reverse connection with objectness prior networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5936–5944, 2017. 6

[13] Changlin Li, Taojiannan Yang, Sijie Zhu, Chen Chen, and Shanyue Guan. Density map guided object detection in aerial images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 190–191, 2020. 1, 2, 3, 4, 6, 7

[14] Jianan Li, Xiaodan Liang, Yunchao Wei, Tingfa Xu, Jiashi Feng, and Shuicheng Yan. Perceptual generative adversarial networks for small object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1222–1230, 2017. 3

[15] Yuhong Li, Xiaofan Zhang, and Deming Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1091–1100, 2018. 3

[16] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 6

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1, 2, 6

[18] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 1, 2, 6

[19] F Ozge Unel, Burak O Ozkalayci, and Cevahir Cigla. The power of tiling for small object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 1, 3

[20] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1, 2

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1, 2, 6

[22] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 3, 5

[23] Bharat Singh, Mahyar Najibi, and Larry S Davis. Sniper: Efficient multi-scale training. In *Advances in neural information processing systems*, pages 9310–9320, 2018. 5

[24] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 9627–9636, 2019. 2

[25] Yi Wang, Youlong Yang, and Xi Zhao. Object detection using clustering algorithm adaptive searching regions in aerial images. In *European Conference on Computer Vision*, pages 651–664. Springer, 2020. 3

[26] Fan Yang, Heng Fan, Peng Chu, Erik Blasch, and Haibin Ling. Clustered object detection in aerial images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8311–8320, 2019. 1, 3, 6

[27] Junyi Zhang, Junying Huang, Xuankun Chen, and Dongyu Zhang. How to fully exploit the abilities of aerial image detectors. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019. 1, 3, 6

[28] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 589–597, 2016. 3, 4

[29] 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)*, 2018. 1, 2, 6, 8