Guided Attention Network for Object Detection and Counting on Drones

Yuanqiang Cai[1], Dawei Du[2], Libo Zhang[3,†], Longyin Wen[4], Weiqiang Wang[1], Yanjun Wu[3], Siwei Lyu[2]
[1]University of Chinese Academy of Sciences, Beijing, China
[2]University at Albany, State University of New York, Albany, NY, USA
[3]Institute of Software Chinese Academy of Sciences, Beijing, China
[4]JD Finance America Corporation, Mountain View, CA, USA

ABSTRACT

Object detection and counting are related but challenging problems, especially for drone based scenes with small objects and cluttered background. In this paper, we propose a new Guided Attention network (GAnet) to deal with both object detection and counting tasks based on the feature pyramid. Different from the previous methods relying on unsupervised attention modules, we fuse different scales of feature maps by using the proposed weakly-supervised Background Attention (BA) between the background and objects for more semantic feature representation. Then, the Foreground Attention (FA) module is developed to consider both global and local appearance of the object to facilitate accurate localization. Moreover, the new data argumentation strategy is designed to train a robust model in the drone based scenes with various illumination conditions. Extensive experiments on three challenging benchmarks (*i.e.*, UAVDT, CARPK and PUCPR+) show the state-of-the-art detection and counting performance of the proposed method compared with existing methods. Code can be found at https://isrc.iscas.ac.cn/gitlab/research/ganet.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision problems**; *Image segmentation*; *Object detection*; *Object recognition*.

KEYWORDS

guided attention network; weakly-supervised background attention; foreground attention; data augmentation

ACM Reference Format:
Yuanqiang Cai, Dawei Du, Libo Zhang, Longyin Wen, Weiqiang Wang, Yanjun Wu, Siwei Lyu. 2020. Guided Attention Network for Object Detection and Counting on Drones. In *Proceedings of the 28th ACM International Conference on Multimedia (MM'20), October 12–16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394171.3413816

†Corresponding author (libo@iscas.ac.cn).

MM '20, October 12–16, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-7988-5/20/10. . . $15.00
https://doi.org/10.1145/3394171.3413816

1 INTRODUCTION

Object detection and counting are fundamental techniques in many applications, such as scene understanding, traffic monitoring and sports video, to name a few. However, these tasks become even more challenging in drone based scenes because of various factors such as small objects, scale variation and background clutter. With the development of deep learning, much progress has been achieved recently. Recent methods deal with the crowd counting solution by convolutional neural networks based object detectors [7, 22, 32, 33]. To further improve the detection and counting accuracy, the deep frameworks focus on discriminative feature representation of the objects.

First of all, the feature pyramid is widely applied in deep learning because it has rich semantics at all levels, *e.g.*, U-Net [35], TDM [36] and FPN [23]. Inspired by a human visual system, the attention modules play an important role in object detection, resulting in better performance. Therefore, the researchers use various attention modules to better exploit multi-scale feature representation. In [15], the channel-wise feature responses are recalibrated adaptively by explicitly modelling interdependencies between channels. Recently, Wang *et al.* [41] improve the efficiency of SENet using a local cross-channel interaction strategy without dimension reduction. Except channel attention, Woo *et al.* [43] refine intermediate features by both channel and spatial attentions. In [42], the non-local operation is proposed to capture long-range dependencies by calculating the correlation matrix between each spatial point in the feature map. To reduce large amount of computational complexity in the non-local blocks, Cao *et al.* [1] develop a lightweight global context (GC) block. However, all the above methods are unsupervised attention modules, and consider little about the background discriminative information in feature maps.

Based on the fused feature maps, the object is represented by proposals in anchor based methods [4, 25, 33] or keypoints in anchor-free methods [19, 44, 52]. Anchor based methods exploit the global appearance information of the object, relying on pre-defined anchors. It is not flexible to design different kinds of anchors because of large scale variation

in drone based scenes. Anchor-free methods employ corner points, center points or target part points to capture local object appearance without anchors. However, local appearance representation does not contain object's structure information, which is less discriminative in cluttered background, especially for small objects.

In addition, the diversity of training data is essential in deep learning. Especially in the drone based scenes, the number of difficult samples is very limited. It is difficult for traditional data argumentation such as rescale, horizontal flip, rotation and cropping to train a robust model to deal with unconstrained drone based scenarios.

To address these issues, in this paper, we propose an anchor-free Guided Attention network (GAnet) with both background and foreground attentions for drone based scenes. To learn discriminative information in cluttered background, we develop the background attention module to enforce different channels of feature maps focus on different semantic information. Then, we fuse can the multi-scale features in the network with the weakly-supervision of classification between background and foreground images. Moreover, to capture the structure of small objects, we employ the foreground attention module to capture both global and local appearance representation of the objects. Specifically, we rely on the corner feature maps of objects to extract more context information. Due to limited training data, we develop a new data argumentation strategy to reduce the influence of different illumination conditions on the images for the drone based scenes, *e.g.*, sunny, night, cloudy and foggy scenes. We conduct the experiment on three challenging datasets (*i.e.*, UAVDT [6], CARPK [14] and PUCPR+ [14]) to show the effectiveness of the proposed method. The main contributions of this paper are summarized as follows.

- We present an anchor-free guided attention network for object detection and counting on drones, including both the foreground and background attention blocks to extract the discriminative feature representation.
- A new data augmentation strategy is designed to ease the influence of various illumination conditions on the drone based scene images and boost up the performance in drone based scenes.
- Extensive experiments on several datasets demonstrate the favorable detection and counting performance of the proposed method against the state-of-the-arts.

2 RELATED WORK

2.1 Object detection algorithms

Object detection requires algorithms to produce a series of bounding boxes with categories, which can be roughly divided into two categories, *i.e.*, anchor-based approach and anchor-free approach. The anchor-based approach uses the anchor boxes to generate object proposals, and then determines the accurate object regions and the corresponding class labels using convolutional networks. For example, Faster R-CNN [34] designs the region proposal network to generate proposals and uses Fast R-CNN [10] to produce accurate bounding boxes and class labels of objects. FPN [23] uses multi-scale, pyramidal hierarchy of deep convolutional networks to construct feature pyramids for object detection. Considering the efficiency, SSD [25], RetinaNet [24], and RefineDet [48] omit the proposal generation step and tile multi-scale anchors at different layers, which run very fast and produce competitive detection accuracy. Recently, the anchor-free approach attracts much attention of researchers, including CornerNet [19], CenterNet [52], FCOS [40], RepPoint [44], which generally produces the bounding boxes of objects by learning the features of several object key-points. The anchor-free approach has shown great potential to surpass the anchor-based approach in terms of both accuracy and efficiency.

2.2 Object counting algorithms

Object counting methods aim to predict the total number of objects in different categories existing in images, such as pedestrian counting [20, 26, 46, 50], vehicle counting [12, 49], goods counting [11, 21] and general object counting [2, 3, 18]. In [2], the regression-based common object counting with image-level and instance-level supervision is investigated. The image-level counting strategy directly estimate the global count of objects without providing their location information [20, 26, 46, 50]. The instance-level counting strategy predict an accurate number of objects with their location information (*e.g.*, center point or bounding box) [3, 11, 12, 21, 43, 49].

The object is represented by proposals with global appearance information in anchor based methods, it is not flexible to design different kinds of anchors because of large scale variation in drone based scenes. However, the object is represented by key-points with local salience information in anchor-free methods, the local salience representation does not contain object's structure information, which is less discriminative in cluttered background, especially for small objects. Our algorithm focus on combining the advantages of both global and local information via a new Guided Attention network (GAnet) to predict object locations and counts.

3 GUIDED ATTENTION NETWORK

In this section, we introduce the novel anchor-free deep learning network for object detection and counting in drone images, the Guided Attention network (GAnet), which is illustrated in Figure 1. Specifically, GAnet consists of three parts, *i.e.*, the backbone, multi-scale feature fusion, and output predictor. We will first describe each part in detail, and then loss function and data argumentation strategy.

3.1 Backbone Network

Since diverse scales of objects are taken into consideration in feature representation, we choose the feature maps from four side-outputs of the backbone network (*e.g.*, VGG-16 [37] and ResNet-50 [13]). Four side outputs correspond to *pool1*, *pool2*, *pool3*, and *pool4*, each of which is the output of four convolution blocks with different scales, respectively. The feature maps from four *pooling* layers are $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$ the size of the input image. They are marked with light blue regions

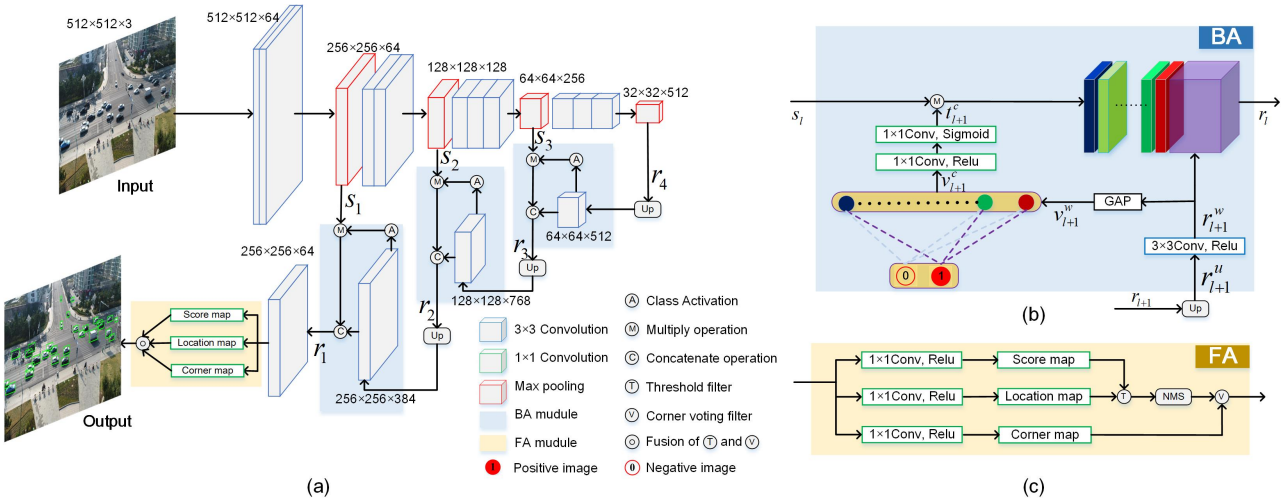

Figure 1: (a) The architecture of GAnet. (b) The background attention module. (c) The foreground attention module. In (a), s_1, s_2, and s_3 denote *pool1*, *pool2*, and *pool3* low-level features, respectively; r_1, r_2, and r_3 denote the corresponding high-level features. In (b), s_l denotes the low-level features with rich texture details, r_{l+1} and r_l denote the high-level features with strong semantic information.

in Figure 1(a). The backbone network is pre-trained by the ImageNet dataset [17].

3.2 Multi-Scale Feature Fusion

As discussed in [23], the feature pyramid has strong semantics at all scales, resulting in significant improvement as a generic feature extractor. Specifically, we fuse the side-outputs of the backbone network from top to down, *e.g.*, feature maps from *pool4* to *pool1* of VGG-16. Meanwhile, the receptive fields of the stacked feature maps can adaptively match the scale of objects. To consider background discriminative information in the feature pyramid, we introduce the Background Attention (BA) module in multi-scale feature fusion.

3.2.1 Background Attention. As shown in Figure 1(b), the BA modules are stacked from the deepest to the shallowest convolutional layer. Meanwhile, the cross-entropy loss function is used to enforce different channels of feature maps focus on either foreground and background in every stage. Then, the attention module weights the pooling features with the same scale via the class-activated tensor. Finally, the weighted pooling features and the up-sampled features are concatenated as the base feature maps in the next BA.

We denote the l-th pooling features as s_l, and the input and output of l-th BA as r_{l+1} and r_l. Specifically, r_{l+1} is used to learn the class-related weights for activating the class-related feature maps in s_l. For the deepest BA module, the input is regarded as the *pool4* feature maps (see r_4 in Figure 1(a)). Note that the size of output r_l in this architecture is the same as the pooling features s_l rather than the size of input r_{l+1}. Therefore, the bilinear interpolation is introduced to up-sample r_{l+1} to r^u_{l+1}. As the up-sampling operation is a linear transformation, one 3×3 convolutional layer w^u_l is used as soft-adding to improve the scale adaptability. Instead of concatenating the up-sampled r_{l+1} and the activated s_l directly, the 1×1 and 3×3 convolutional layers w^c_l is used to generate r_l. In summary, the l-th BA is formulated as

$$r_l = w^c_l \cdot (f(s_l, r^w_{l+1}) + r^w_{l+1}), \tag{1}$$

where w^c_l denotes the convolutional weights of the concatenation layer. $r^w_{l+1} = w^u_l * r^u_{l+1}$ and w^u_l are the convolutional weights of up-sampled r^u_{l+1}. w^c_l has two elements, *i.e.*, one for r^w_{l+1} and the other for $f(s_l, r^w_{l+1})$. $f(s_l, r^w_{l+1})$ is a class activation function with two parameters, *i.e.*, the pooling features s_l and the weighted up-sampled features r^w_{l+1}. It is defined as

$$f(s_l, r^w_{l+1}) = s_l \otimes g^c(r^w_{l+1}), \tag{2}$$

where $\otimes$ is the multiply operation between the features s_l and the weight tensor $g^c(r^w_{l+1})$. $g^c(r^w_{l+1})$ is obtained by three steps. First, r^w_{l+1} is compressed into a one-dimensional vector v^w_{l+1} by the Global Average Pooling (GAP) [51]. Second, v^w_{l+1} is activated and converted to the vector with class-related information v^c_{l+1} via determining whether the input image contains the objects. Third, v^c_{l+1} is transformed into a weight tensor with class-related information $t^c_{l+1} = g^c(r^w_{l+1})$ via two 1×1 convolutional layers.

3.2.2 Supervision for Background Attention. To learn class-related feature maps, we use both the images with and without objects in the training stage. We denote them as positive and negative images respectively. Specifically, we use positive images with objects to activate the channels of feature maps to represent the pixels of object region, and negative images without overlapping of objects to activate the channels of feature maps to describe the background region. As shown in Figure 2, we generate positive and negative images with

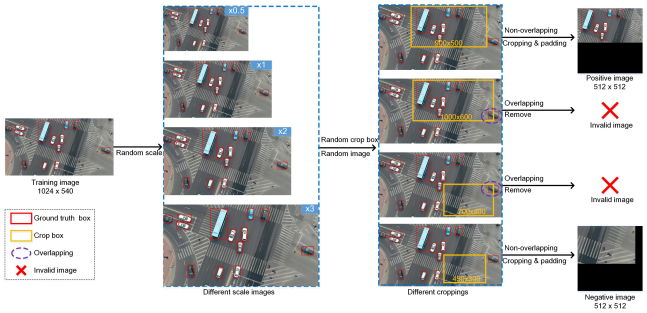

Figure 2: Generation of positive and negative samples.

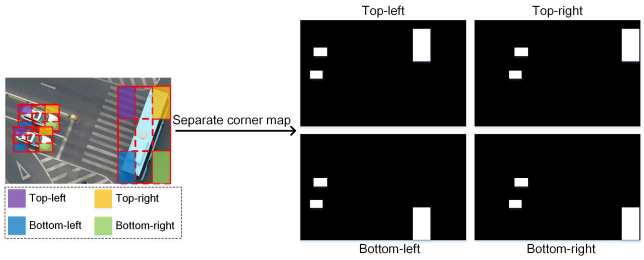

Figure 3: Illustration of four corner maps for foreground attention.

the size of 512×512 by randomly cropping and padding the rescaled training images (from 0.5x to 3x scale).

3.3 Output Predictor

Based on multi-scale feature fusion, we predict the scales and locations of objects using both score and location maps (see Figure 1(c)), which are defined as follows:

- The score map corresponds to confidence score of the object region. Similar to the confidence map in FCN [27], each pixel of the score map is a scalar between 0 to 1 representing the confidence belonging to an object region.
- The location map describes the location of object by using four distance channels $G = (l, t, r, b)$. The channels denote the distances from the current pixel i to the *left*, *top*, *right*, and *bottom* edges of the bounding box respectively. Then we can directly predict the object box by four distance channels. Specifically, for each point in the score map, four distance channels predict the distances to the above four edges of the bounding box.

Foreground Attention. In general, based on both score and location maps, we can estimate the bounding boxes of the objects in the image. However, the estimated bounding boxes only rely on the global appearance of the object. That is, little local appearance of the object is taken into consideration, resulting in less discriminative foreground representation. To improve localization accuracy, we introduce the Foreground Attention (FA) module to consider both global and local appearance representation of the objects.

In practice, we use four corner maps (*top-left*, *top-right*, *bottom-left* and *bottom-right*) to denote different corner positions within the object region, as shown in Figure 3. Similar to score map, each pixel of the corner map is also a scalar between 0 to 1 representing the confidence belonging to a corresponding position in the object region. The corner is set as 1/9 the size of the whole object. Specifically, as illustrated in Figure 1(c), we first use a threshold filter to remove the candidate bounding boxes with low confidence pixels, *i.e.*, $c_i < \mu$. c_i is the confidence value of pixel i in the predicted score map, and μ denotes the confidence threshold. Then, the locality-aware NMS operation [53] is applied to remove redundant candidate bounding boxes and choose the top ones with higher confidence. Finally, a corner voting filter is designed to determine whether the selected bounding boxes should be retained. Specifically, we calculate the number of reliable corners $\mathbb{N}(b_k)$ in the k-th candidate bounding box b_k by

$$\mathbb{N}(b_k) = \sum_{s=1}^{4} \mathbb{I}(\tau(\mathcal{C}_s) > \varepsilon), \tag{3}$$

where $\tau(\mathcal{C}_s)$ denotes the average confidence of the corner region $\mathcal{C}_s$. ε indicates the threshold of mean confidence $\tau(\mathcal{C}_s)$ to determine the reliable corner. $\mathbb{I}(\cdot) = 1$ if its argument is true, and 0 otherwise. We only keep the bounding box b_k if the number of reliable corners is larger than the threshold κ, *i.e.*, $\mathbb{N}(b_k) > \kappa$.

3.4 Loss function

To train the proposed network, We optimize the location map and score map, as well as both foreground and background attentions simultaneously. The overall loss function is defined as

$$\mathcal{L} = \mathcal{L}_{\text{loc}} + \lambda_{\text{sco}}\mathcal{L}_{\text{sco}} + \lambda_{\text{FA}}\mathcal{L}_{\text{FA}} + \lambda_{\text{BA}}\mathcal{L}_{\text{BA}}, \tag{4}$$

where $\mathcal{L}_{\text{loc}}$, $\mathcal{L}_{\text{sco}}$, $\mathcal{L}_{\text{FA}}$, and $\mathcal{L}_{\text{BA}}$ are loss terms for the location map, score map, foreground attention, and background attention, respectively. The parameter λ_{sco}, λ_{FA}, and λ_{BA} are used to balance these terms. In the following, we explain these loss terms in detail.

Location Map Loss. To achieve scale-invariance, the IoU loss [45] is adopted to evaluate the difference between the predicted bounding box and the ground truth of bounding box. The loss of location map is defined as:

$$\mathcal{L}_{\text{loc}} = \text{IoU}(G, G^*), \tag{5}$$

where $G = (l, t, r, b)$ and $G^* = (l^*, t^*, r^*, b^*)$ are the estimated and ground-truth bounding box of the object. The function $\text{IoU}(\cdot)$ calculates the intersection-over-union (IoU) score between G and G^*.

Score Map Loss. Similar to image segmentation [47], we use the Dice loss to deal with the imbalance problem of positive and negative pixels in the score map. It calculates the errors between the predicated score map and ground-truth map,

i.e.,

$$\mathcal{L}_{\text{sco}} = 1 - \frac{2 \cdot \sum_{i=1}^{N}(c_i c_i^*)}{\sum_{i=1}^{N}(c_i) + \sum_{i=1}^{N}(c_i^*)}, \tag{6}$$

where the sums run over the all N pixels of the score map. c_i^* and c_i are the confidence values of pixel i in the ground-truth and predicted maps respectively.

Background Attention Loss. Similar to classification algorithms, we use the cross-entropy loss $\mathcal{L}_{\text{BA}}$ to guide background attention based on the binary classification, *i.e.*,

$$\mathcal{L}_{\text{BA}} = \begin{cases} -\log(p) & \text{if} \quad y = 1, \\ -\log(1-p) & \text{otherwise}, \end{cases} \tag{7}$$

where $y \in \{\pm 1\}$ denotes the ground-truth category (*i.e.*, foreground or background), $p \in [0, 1]$ is the estimated probability for the category with label $y = 1$.

Foreground Attention Loss. Similar to the score map, to deal with the imbalance problem of positive and negative pixels in the feature maps, we use the Dice loss to guide the foreground attention for the four corner maps.

3.5 Data Augmentation for Drones

Data augmentation is important in deep network training based on limited training data. Since the data is captured from a very high altitude by the drone, it is susceptible to the influence of different illumination conditions, *e.g.*, sunny, night, cloudy and foggy. Therefore, we develop a new data augmentation strategy for drones.

As discussed above, sunny or night scenes correspond to the brightness of the image, therefore we synthesize these scenes via changing the whole contrast of the image (denoted as *BNoise*). On the other hand, since convincing representations of clouds and water can be created in pixel-level [29], we use Perlin noise [30] to imitate cloudy and foggy scenes (denoted as *PNoise*). Inspired by the image blending algorithm [39], the data augmentation model is defined as

$$\Phi(i) = \alpha I(i) + \beta M^*(i) + \gamma, \tag{8}$$

where $\Phi(i)$ is the transformed value of the pixel i in image. α and β denote the weight of the pixel of original image $I(p)$ and noise map $M^*(i)$ respectively. The asterisk $*$ denotes different kinds of noise maps, *i.e.*, BNoise $M^b(i)$ and PNoise $M^p(i)$. We have $\alpha = 1 - \beta$ to control the contrast of the image. The perturbation factor γ is used to revise the brightness. We set different factors α and γ for each image in the training phase.

As shown in Figure 4(a), we employ white and black maps to synthesize sunny or night images. On the other hand, we use Perlin noise [30] to generate noise maps in Figure 4(b), and then revise the brightness via disturbance factor γ to synthesize cloudy and foggy images. For each training image, we first resize it using random scale factors (x0.5, x1, x2 and x3). Then, we introduce both noise maps into the image to imitate the challenging scenes (*i.e.*, sunny, night, cloudy, and foggy). Finally, we select positive and negative images by random cropping on the blending images, and transform the selected images to 512×512 size via zooming and padding.

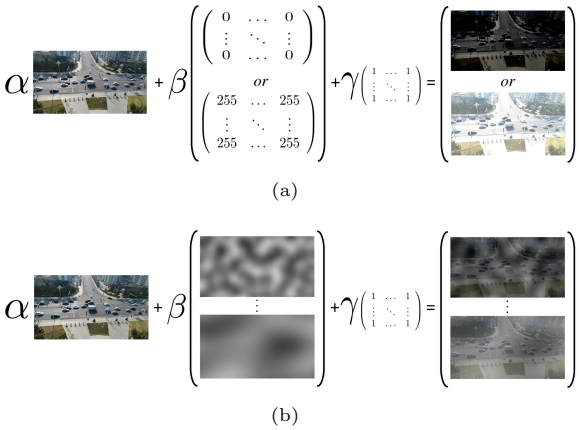

Figure 4: Illustration of data augmentation including (a) BNoise (Brightness noises to imitate sunny or night scenes) and (b) PNoise (Perlin noises to imitate cloudy and foggy scenes).

4 EXPERIMENT

We evaluate our method on three datasets: UAVDT [6], CARPK [14], and PUCPR+ dataset [14]. In this section, we first describe implementation details. Then, we compare our GAnet with the state-of-the-art methods. More visual examples are shown in Figure 5. In addition, the ablation study is carried out to evaluate the effectiveness of each component in our network.

4.1 Implementation Details

Due to the shortage of computational resources, we train GAnet using the VGG-16 and ResNet-50 backbone with the input size 512×512. All the experiments are carried out on the machine with NVIDIA Titan Xp GPU and Intel(R) Xeon(R) E5-1603v4@2.80GHz CPU.

For fair evaluation, we generate the same top 200 detection bounding boxes for the UAVDT and CARPK datasets and 400 detection bounding boxes for the PUCPR+ dataset based on the detection confidence. Note that the detection confidence is calculated by summarizing the value of each pixel in the score map. To output the count of objects in each image, we calculate the number of detection with the detection confidence larger than 0.5.

We train our method using the Adam Optimizer. An exponential decay learning rate is used in the training phrase, *i.e.*, its initial value is 0.0001 and decays every $10,000$ iterations with the decay rate 0.94. The batch size is set as 10. In the loss function (4), we set the balancing factors as $\lambda_{sco} = 0.01$, $\lambda_{\text{FA}} = 0.0025$, $\lambda_{\text{BA}} = 0.001$ empirically. In the FA module, the confidence threshold μ is set as 0.8, and the threshold ε in (3) is set as 0.3 empirically. The NMS operation is conducted with a threshold 0.2. In the data argumentation model (8), we set the balancing weights as $\alpha = \{0.1, 0.3, 0.5, 0.7, 0.8, 0.9, 1.0\}$ and $\gamma = [-20, 20]$.

(a) UAVDT (b) CARPK (c) PUCPR+

Figure 5: Visual examples of GAnet with VGG16 backbone. The ground-truth and predicted detection bounding boxes are highlighted in red and green rectangles, respectively. The blue mask in the top-right corner indicates the comparison between the ground-truth (GT) and estimated detection (DT) counts.

Table 1: Comparison on the UAVDT dataset.

Method	Backbone	MAE↓	RMSE↓	AP@0.7[%]↑
YOLO9000 [31]	DarkNet-19	12.59	16.73	7.6
YOLOv3 [32]	DarkNet-53	11.58	21.50	20.3
RON [16]	VGG-16	-	-	21.6
Faster R-CNN [33]	VGG-16	-	-	22.3
SSD [25]	VGG-16	-	-	33.6
CADNet [7]	VGG-16	-	-	43.6
Ours	VGG-16	**5.10**	**8.10**	**46.8**
SA+CF+CRT [22]	ResNet-101	7.67	10.95	27.8
R-FCN	ResNet-50	-	-	34.4
Ours	ResNet-50	**5.09**	**8.16**	**47.2**

Table 2: Comparison on the CARPK dataset.

Method	MAE↓	RMSE↓	AP@0.5[%]↑	AP@0.7[%]↑
One-Look Reg [28]	59.46	66.84	-	-
IEP [38]	51.83	-	-	-
Faster R-CNN [33]	47.45	57.39	-	-
YOLO9000 [31]	38.59	43.18	20.9	3.7
SSD [25]	37.33	42.32	68.7	25.9
LPN [14]	23.80	36.79	-	-
RetinaNet [24]	16.62	22.30	-	-
YOLOv3 [32]	7.92	11.08	85.3	47.0
IoUNet [11]	6.77	8.52	-	-
SA+CF+CRT [22]	5.42	7.38	89.8	61.4
Ours (VGG-16)	4.80	6.94	**90.2**	73.6
Ours (ResNet-50)	**4.61**	**6.55**	90.1	**74.9**

Metrics. To evaluate detection algorithms on the UAVDT dataset [6], we compute the Average Precision (AP@0.7) score based on [8, 9]. That is, the hit/miss threshold of the overlap between detection and ground-truth bounding boxes is set to 0.7. In terms of CARPK [14] and PUCPR+ [14], we report the detection score under two hit/miss thresholds, *i.e.*, AP@0.5 and AP@0.7. To evaluate the counting results, similar to [14], we use two object counting metrics including Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

4.2 Quantitative Evaluation

Evaluation on UAVDT. The UAVDT dataset [6] consists of 100 video sequences with approximate $80,000$ frames, which are collected from various scenes. Moreover, the objects are annotated by bounding boxes as well as several attributes (*e.g.*, weather condition, flying altitude, and camera view). Note that we only use the subset of UAVDT dataset for object detection in our experiment.

As presented in Table 1, we can conclude that our GAnet performs the best among all the compared detection methods in terms of both the VGG-16 and ResNet-50 backbones. Specifically, GAnet with VGG-16 backbone achieves 5.10 MAE and 8.10 RMSE; while GAnet with ResNet-50 backbone achieves better 5.09 MAE and 8.16 RMSE. Besides, GAnet surpasses YOLO9000 [31], YOLOv3 [32], RON [16], Faster R-CNN [33], SSD [25], CADNet [7], R-FCN [4] and SA+CF+CRT [22] by 39.2%, 26.3% 25.2%, 24.5%, 13.2%, 3.2%, 12.8%, and 19.4% AP scores, respectively. Moreover, our method achieves better counting accuracy than SA+CF+CRT [22] with the more complex ResNet-101 backbone, *i.e.*, 5.09 vs. 7.67 in MAE score and 8.16 vs. 10.95 in RMSE score. It demonstrates that the effectiveness of our method in object detection in drone based scenes.

Evaluation on CARPK. The CARPK dataset [14] provides the largest-scale drone view parking lot dataset in unconstrained scenes, which is collected in various scenes for

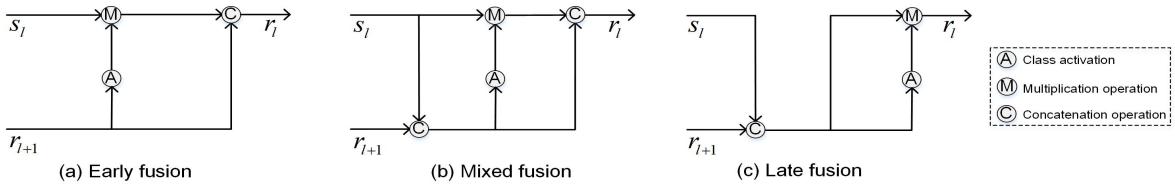

Figure 6: Different fusion strategies of multi-scale feature maps. s_l denotes the low-level features with rich texture details, r_{l+1} and r_l denote the high-level features with strong semantic information.

Table 3: Comparison on the PUCPR+ dataset.

Method	MAE↓	RMSE↓	AP@0.5[%]↑	AP@0.7[%]↑
SSD [25]	119.24	132.22	32.6	7.1
Faster R-CNN [33]	111.40	149.35	-	-
YOLO9000 [31]	97.96	133.25	12.3	4.5
RetinaNet [24]	24.58	33.12	-	-
LPN [14]	23.80	36.79	-	-
One-Look Reg [28]	21.88	36.73	-	-
IEP [38]	15.17	-	-	-
IoUNet [11]	7.16	12.00	-	-
YOLOv3 [32]	5.24	7.14	**95.0**	45.4
SA+CF+CRT [22]	3.92	5.06	92.9	55.4
Ours (VGG-16)	3.68	5.47	91.3	**67.0**
Ours (ResNet-50)	**3.28**	**4.96**	91.4	65.5

Table 4: Influence of data augmentation.

Method	AP	AP_{day}	AP_{night}	AP_{fog}
GAnet	0.3908	0.4779	0.5513	0.1509
GAnet+BNoise	0.4034	0.4928	**0.5686**	0.1579
GAnet+PNoise	0.4063	0.4798	0.5263	**0.2118**
GAnet+BPNoise	**0.4181**	**0.4940**	0.5581	0.2027

4 different parking lots. It contains approximately 90,000 cars in total.

We compare our method with state-of-the-art algorithms in Table 2. The results show that our approach achieves the best MAE, RMSE and AP scores. Specifically, GAnet with VGG-16 backbone achieves 4.80 MAE and 6.94 RMSE; while GAnet with ResNet-50 backbone achieves better 4.61 MAE and 6.55 RMSE. Both of them obtain the AP@0.5 score more than 90.0%. It is worth mentioning that we obtain much better AP@0.7 score than the second best SA+CF+CRT [22] (*i.e.*, 74.9% vs. 61.4%). This is attributed to the proposed attention modules to locate the objects more accurately.

Evaluation on PUCPR+. The PUCPR+ dataset [14] is the subset of PKLot [5], which is annotated with nearly 17,000 cars in total. It shares the similar high altitude attribute to drone based scenes, but the camera sensors are fixed and set in the same place.

As presented in Table 3, our method performs the best in terms of MAE and RMSE. Specifically, GAnet with VGG-16 backbone achieves 3.68 MAE and 5.47 RMSE; while GAnet with ResNet-50 backbone achieves better 3.28 MAE and 5.47 RMSE. YOLOv3 [32] achieves the best AP score at 0.5 hit/miss threshold, but inferior AP@0.7 score than that of our method. We speculate that YOLOv3 lack of global appearance representation of objects to achieve accurate localization. SA+CF+CRT [22] performs slightly better than our method in AP@0.5 score, but much worse in AP@0.7 score. It indicates the effectiveness of our method.

Table 5: Influence of background attention.

Method	AP	AP_{front}	AP_{side}	AP_{bird}
GAnet+BPNoise	0.4181	0.4618	0.5219	0.2533
GAnet+BPNoise+LF	0.4457	0.4667	0.5301	0.3294
GAnet+BPNoise+MF	0.4530	0.4699	**0.5338**	0.3495
GAnet+BPNoise+EF	**0.4576**	**0.4719**	0.5309	**0.3640**
GAnet+BPNoise+FPN	0.3985	0.4378	0.4943	0.2480
GAnet+BPNoise+GC	0.4343	0.4681	**0.5374**	0.2919
GAnet+BPNoise+SE	0.4442	**0.4723**	0.5347	0.3142
GAnet+BPNoise+BA	**0.4576**	0.4719	0.5309	**0.3640**

4.3 Ablation Study

We select the UAVDT dataset [6] to conduct the ablation experiment because it provides various attributes in terms of altitude, illumination and camera-view for comprehensive evaluation.

Effectiveness of Data Augmentation. As discussed above, the data augmentation strategy is used to increase the difficult samples affected by various illumination attributes in the UAVDT dataset [6] such as *daylight*, *night* and *fog*. We compare different variants of GAnet with different data augmentation, denoted as GAnet+BNoise, GAnet+PNoise and GAnet+PBNoise. Notably, BNoise denotes the brightness noise, PNoise denotes the Perlin noise, and BPNoise denotes both. As shown in Table 4, GAnet+BNoise outperforms GAnet slightly, which shows the effectiveness of brightness noise in daylight and night scenarios. GAnet+PNoise achieves better AP score in terms of foggy scenes compared to GAnet (21.18% AP_{fog} vs. 15.09% AP_{fog}), which demonstrates that Perlin noise can simulate the foggy scenes effectively. If we use the above two data augmentation strategies, the performance will increase by 2.73% AP score.

Table 6: Comparison of variants of GAnet on the UAVDT dataset.

Method	AP	AP_{day}	AP_{night}	AP_{fog}	AP_{low}	AP_{med}	AP_{high}	AP_{front}	AP_{side}	AP_{bird}
GAnet	0.3908	0.4779	0.5513	0.1509	0.5505	0.4616	0.1227	0.4478	0.5111	0.1981
GAnet+BPNoise	0.4181	0.4940	0.5581	0.2027	0.5565	0.4867	0.1665	0.4618	0.5219	0.2533
GAnet+FA	0.4207	0.5006	**0.5878**	0.1890	**0.5935**	0.4834	0.1431	0.4595	0.5462	0.2439
GAnet+BA	0.4353	0.5041	0.5743	0.2401	0.5908	0.4812	0.1996	0.4655	0.5451	0.2947
GAnet+FA+BA	0.4519	0.5079	0.5781	0.2686	0.5763	0.4955	0.2514	0.4667	0.5407	0.3434
GAnet+BPNoise+FA	0.4411	**0.5272**	0.5819	0.2139	0.5900	0.5146	0.1751	0.4805	**0.5618**	0.2715
GAnet+BPNoise+BA	0.4576	0.5049	0.5779	0.3068	0.5815	0.4923	**0.2695**	0.4719	0.5309	**0.3640**
GAnet+BPNoise+FA+BA	**0.4679**	0.5240	0.5841	**0.3084**	0.5820	**0.5206**	0.2624	**0.4852**	0.5435	0.3603

Table 7: Influence of foreground attention.

Method	κ	AP	AP_{low}	AP_{med}	AP_{high}
GAnet+BPNoise	-	0.4181	0.5565	0.4867	0.1656
GAnet+BPNoise+FA	0	0.4271	0.5729	0.4978	0.1698
	1	**0.4411**	**0.5900**	**0.5146**	**0.1751**
	2	0.4391	0.5869	0.5130	0.1736
	3	0.4372	0.5817	0.5128	0.1718
	4	0.4347	0.5764	0.5116	0.1699

Effectiveness of Background Attention. Different from the previous unsupervised attention modules, our Background Attention (BA) is guided based on discrimination between the background and objects. Firstly, we study different fusion strategies of the proposed BA module in Figure 6, *i.e.*, early fusion (EF), mixed fusion (MF) and late fusion (LF). The results presented in Table 5 show the early fusion strategy (*i.e.*, GAnet+BPNoise+EF) achieves the best performance. Secondly, we also compare the BA module with several previous channel-wise attention modules including SE block [15] and GC block [1]. For fair comparison, we use the same early fusion strategy in Figure 6(a). Compared to the baseline FPN fusion strategy using lateral connection [23], all the attention modules can improve the performance by learning the weights of different channels of feature maps. However, our BA module can learn additional discriminative information of background, resulting in the best AP score in the drone based scenes under different camera views.

Effectiveness of Foreground Attention. We enumerate the threshold for Foreground Attention (FA) κ in (3), *i.e.*, $\kappa = \{0, 1, 2, 3, 4\}$, to study its influence on the accuracy. As shown in Table 7, we can conclude that GAnet with the FA module achieves the best AP score 44.11% when the threshold $\kappa = 1$. If we remove the FA module, the detection performance will decrease to 41.81%. It shows the effectiveness of the FA module.

Variants of GAnet. In Table 6, we show the impact of different conditions and compare various variants of GAnet that combine several components in the network. Using data argumentation strategy GAnet can improve the performance considerably in all the attributes. FA facilitates GAnet to extract the local and global salient features of the object, and BA can distinguish difficult background effectively and extract multi-scale features of objects, especially small objects. Either BA or FA can improve the performance by 3% $\sim$ 4%. Combining BA and FA, the GAnet+BA+FA method achieves the AP score of 45.19%, which surpasses GAnet, GAnet+FA, and GAnet+BA by 6.11% (*i.e.*, 45.19% vs. 39.08%), 3.12% (*i.e.*, 45.19% vs. 42.07%), and 1.66% (*i.e.*, 45.19% vs. 43.53%) respectively. Moreover, the proposed method with all proposed modules (*i.e.*, GAnet+BPNoise+FA+BA) can boost the performance by approximate 8% improvement in AP score compared to the baseline GAnet method.

5 CONCLUSION

In the paper, we propose a novel guided attention network to deal with object detection and counting in drone based scenes. Specifically, we develop both background and foreground attention modules to not only learn background discriminative representation but also consider local appearance of the object, resulting in better accuracy. Moreover, we propose a new data argumentation strategy in drone based scenes. Extensive experiments on three challenging datasets demonstrate that our method can improve the localization and counting accuracy considerably with different backbone. We plan to expand our method to multi-class object detection and counting for future work.

6 ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (No. 61807033, No. 61976201), the Key Research Program of Frontier Sciences, CAS (No. ZDBS-LY-JSC03), the NSFC Key Projects of International (Regional) Cooperation and Exchanges (No. 61860206004), and the Ningbo 2025 Key Project of Science and Technology Innovation (No. 2018B10071). Libo Zhang was supported by Youth Innovation Promotion Association, CAS (2020111), and Outstanding Youth Scientist Project of ISCAS.

REFERENCES

[1] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. 2019. GCNet: Non-local Networks Meet Squeeze-Excitation Networks and Beyond. *CoRR* abs/1904.11492 (2019).
[2] Prithvijit Chattopadhyay, Ramakrishna Vedantam, Ramprasaath R. Selvaraju, Dhruv Batra, and Devi Parikh. 2017.

Counting Everyday Objects in Everyday Scenes. In *CVPR*. 4428–4437.
[3] Hisham Cholakkal, Guolei Sun, Fahad Shahbaz Khan, and Ling Shao. 2019. Object Counting and Instance Segmentation with Image-level Supervision. *CoRR* abs/1903.02494 (2019).
[4] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. 2016. R-FCN: Object Detection via Region-based Fully Convolutional Networks. In *NeurIPS*. 379–387.
[5] Paulo Ricardo Lisboa de Almeida, Luiz S. Oliveira, Alceu S. Britto Jr., Eunelson Jose da Silva Junior, and Alessandro L. Koerich. 2015. PKLot - A robust dataset for parking lot classification. *Expert Syst. Appl.* 42, 11 (2015), 4937–4949.
[6] Dawei Du, Yuankai Qi, Hongyang Yu, Yi-Fan Yang, Kaiwen Duan, Guorong Li, Weigang Zhang, Qingming Huang, and Qi Tian. 2018. The Unmanned Aerial Vehicle Benchmark: Object Detection and Tracking. In *ECCV*. 375–391.
[7] Kaiwen Duan, Dawei Du, Honggang Qi, and Qingming Huang. 2019. Detecting Small Objects Using a Channel-Aware Deconvolutional Network. *TCSVT* (2019).
[8] Mark Everingham, S. M. Ali Eslami, Luc J. Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. 2015. The Pascal Visual Object Classes Challenge: A Retrospective. *IJCV* 111, 1 (2015), 98–136.
[9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *CVPR*. 3354–3361.
[10] Ross B. Girshick. 2015. Fast R-CNN. In *ICCV*. 1440–1448.
[11] Eran Goldman, Roei Herzig, Aviv Eisenschtat, Oria Ratzon, Itsik Levi, Jacob Goldberger, and Tal Hassner. 2019. Precise Detection in Densely Packed Scenes. *CoRR* abs/1904.00853 (2019).
[12] Ricardo Guerrero-Gómez-Olmedo, Beatriz Torre-Jiménez, Roberto Javier López-Sastre, Saturnino Maldonado-Bascón, and Daniel Oñoro-Rubio. 2015. Extremely Overlapping Vehicle Counting. In *IbPRIA*. 423–431.
[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR*. 770–778.
[14] Meng-Ru Hsieh, Yen-Liang Lin, and Winston H. Hsu. 2017. Drone-Based Object Counting by Spatially Regularized Regional Proposal Network. In *ICCV*. 4165–4173.
[15] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-Excitation Networks. In *CVPR*. 7132–7141.
[16] Tao Kong, Fuchun Sun, Anbang Yao, Huaping Liu, Ming Lu, and Yurong Chen. 2017. RON: Reverse Connection with Objectness Prior Networks for Object Detection. In *CVPR*. 5244–5252.
[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *NeurIPS*. 1106–1114.
[18] Issam H. Laradji, Negar Rostamzadeh, Pedro O. Pinheiro, David Vázquez, and Mark W. Schmidt. 2018. Where Are the Blobs: Counting by Localization with Point Supervision. In *ECCV*. 560–576.
[19] Hei Law and Jia Deng. 2018. CornerNet: Detecting Objects as Paired Keypoints. In *ECCV*. 765–781.
[20] Victor S. Lempitsky and Andrew Zisserman. 2010. Learning To Count Objects in Images. In *NeurIPS*. 1324–1332.
[21] Congcong Li, Dawei Du, Libo Zhang, Tiejian Luo, Yanjun Wu, Qi Tian, Longyin Wen, and Siwei Lyu. 2019. Data Priming Network for Automatic Check-Out. In *ACM MM*.
[22] Wei Li, Hongliang Li, Qingbo Wu, Xiaoyu Chen, and King Ngi Ngan. 2019. Simultaneously Detecting and Counting Dense Vehicles From Drone Images. *TIE* 66, 12 (2019), 9651–9662.
[23] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. 2017. Feature Pyramid Networks for Object Detection. In *CVPR*. 936–944.
[24] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. 2017. Focal Loss for Dense Object Detection. In *ICCV*. 2999–3007.
[25] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. 2016. SSD: Single Shot MultiBox Detector. In *ECCV*.
[26] Xialei Liu, Joost van de Weijer, and Andrew D. Bagdanov. 2018. Leveraging Unlabeled Data for Crowd Counting by Learning to Rank. In *CVPR*. 7661–7669.
[27] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*. 3431–3440.
[28] T. Nathan Mundhenk, Goran Konjevod, Wesam A. Sakla, and Kofi Boakye. 2016. A Large Contextual Dataset for Classification, Detection and Counting of Cars with Deep Learning. In *ECCV*. 785–800.
[29] Ken Perlin. 1985. An image synthesizer. In *SIGGRAPH*. 287–296.
[30] Ken Perlin. 2002. Improving noise. *TOG* 21, 3 (2002), 681–682.
[31] Joseph Redmon and Ali Farhadi. 2017. YOLO9000: Better, Faster, Stronger. In *CVPR*. 6517–6525.
[32] Joseph Redmon and Ali Farhadi. 2018. YOLOv3: An Incremental Improvement. *CoRR* abs/1804.02767 (2018). arXiv:1804.02767 http://arxiv.org/abs/1804.02767
[33] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NeurIPS*. 91–99.
[34] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *TPAMI* 39, 6 (2017), 1137–1149.
[35] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *MICCAI*. 234–241.
[36] Abhinav Shrivastava, Rahul Sukthankar, Jitendra Malik, and Abhinav Gupta. 2016. Beyond Skip Connections: Top-Down Modulation for Object Detection. *CoRR* abs/1612.06851 (2016). arXiv:1612.06851 http://arxiv.org/abs/1612.06851
[37] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*.
[38] Tobias Stahl, Silvia L. Pintea, and Jan C. van Gemert. 2019. Divide and Count: Generic Object Counting by Image Divisions. *TIP* 28, 2 (2019), 1035–1044.
[39] Richard Szeliski. 2010. *Computer vision: algorithms and applications*. Springer Science & Business Media.
[40] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. 2019. FCOS: Fully Convolutional One-Stage Object Detection. *CoRR* abs/1904.01355 (2019). http://arxiv.org/abs/1904.01355
[41] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. 2019. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. *CoRR* abs/1910.03151 (2019).
[42] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. 2018. Non-Local Neural Networks. In *CVPR*. 7794–7803.
[43] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. 2018. CBAM: Convolutional Block Attention Module. In *ECCV*. 3–19.
[44] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. 2019. RepPoints: Point Set Representation for Object Detection. *CoRR* abs/1904.11490 (2019). arXiv:1904.11490 http://arxiv.org/abs/1904.11490
[45] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas S. Huang. 2016. UnitBox: An Advanced Object Detection Network. In *ACM MM*.
[46] Cong Zhang, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. 2015. Cross-scene crowd counting via deep convolutional neural networks. In *CVPR*. 833–841.
[47] Jiachi Zhang, Xiaolei Shen, Tianqi Zhuo, and Hong Zhou. 2017. Brain Tumor Segmentation Based on Refined Fully Convolutional Neural Networks with A Hierarchical Dice Loss. *CoRR* abs/1712.09093 (2017). http://arxiv.org/abs/1712.09093
[48] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z. Li. 2018. Single-Shot Refinement Neural Network for Object Detection. In *CVPR*. 4203–4212.
[49] Shanghang Zhang, Guanhang Wu, João P. Costeira, and José M. F. Moura. 2017. FCN-rLSTM: Deep Spatio-Temporal Neural Networks for Vehicle Counting in City Cameras. In *ICCV*. 3687–3696.
[50] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. 2016. Single-Image Crowd Counting via Multi-Column Convolutional Neural Network. In *CVPR*. 589–597.
[51] Tinghui Zhou, Philipp Krähenbühl, Mathieu Aubry, Qi-Xing Huang, and Alexei A. Efros. 2016. Learning Dense Correspondence via 3D-Guided Cycle Consistency. In *CVPR*. 117–126.
[52] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. 2019. Objects as Points. *CoRR* abs/1904.07850 (2019). http://arxiv.org/abs/1904.07850
[53] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. 2017. EAST: An Efficient and Accurate Scene Text Detector. In *CVPR*. 2642–2651.