Shifted Autoencoders for Point Annotation Restoration in Object Counting

Yuda Zou*, Xin Xiao*, Peilin Zhou, Zhichao Sun, Bo Du, and Yongchao Xu ✉

National Engineering Research Center for Multimedia Software, Institute of Artificial Intelligence, School of Computer Science, Hubei Luojia Laboratory, Wuhan University, Wuhan, China
{zouyuda,xinxiao,peilinzhou,zhichaosun,dubo,yongchao.xu}@whu.edu.cn

Abstract. Object counting typically uses 2D point annotations. The complexity of object shapes and the subjectivity of annotators may lead to annotation inconsistency, potentially confusing counting model training. Some sophisticated noise-resistance counting methods have been proposed to alleviate this issue. Differently, we aim to directly refine the initial point annotations before training counting models. For that, we propose the Shifted Autoencoders (SAE), which enhances annotation consistency. Specifically, SAE applies random shifts to initial point annotations and employs a UNet to restore them to their original positions. Similar to MAE reconstruction, the trained SAE captures general position knowledge and ignores specific manual offset noise. This allows to restore the initial point annotations to more general and thus consistent positions. Extensive experiments show that using such refined consistent annotations to train some advanced (including noise-resistance) object counting models steadily/significantly boosts their performances. Remarkably, the proposed SAE helps to set new records on nine datasets. The code is available at `https://github.com/zouyuda220/SAE`.

Keywords: Object counting · Annotation refinement · Crowd counting

1 Introduction

Object counting [3,9,19,20,26,27,36,48], increasingly vital in domains like security surveillance [34], urban planning [18], and biological research [1], has benefited greatly from advancements in computer vision. Most object counting methods can be roughly classified into two categories: localization-based [25, 32, 36] and density-map-based approaches [4, 6, 8–10, 14, 24, 27, 38, 41, 42]. Localization-based methods focus on identifying individual objects with bounding box [32] or point [25, 36, 47] representation. In contrast, density-map-based methods apply regression techniques to estimate the density distribution of objects.

Object counting datasets [9, 15, 16, 28, 29, 35, 50], distinct from those for object detection, predominantly use 2D coordinate points for marking objects. This

* Equal contribution

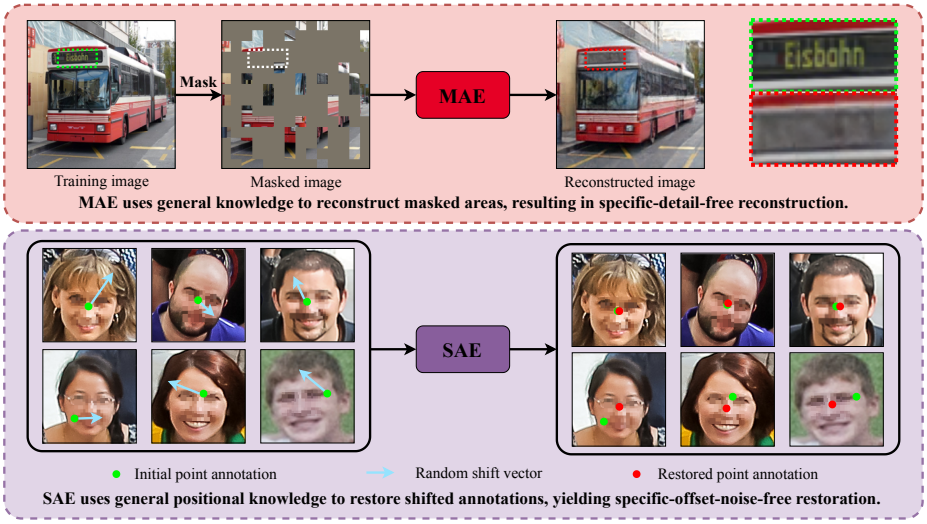

Fig. 1: Drawing inspiration from MAE, our SAE captures general positional knowledge by being trained to restore the shifted point annotations to their original positions. In the restoration phase, the trained SAE restores the initial point annotations to more common positions using the learned general positional knowledge.

annotation mode is particularly advantageous for densely packed or overlapping objects. However, this approach inevitably results in variations and inconsistencies in point annotations within the dataset, primarily due to the subjective decisions made by annotators in selecting the point annotation positions for different objects of the same type. Inconsistencies in point annotations may introduce ambiguity and confusion during the training phase of counting models, compromising their counting accuracy.

To mitigate the issue of annotation inconsistencies in object counting datasets, various strategies have been proposed. ADSCNet [2] employs prediction of the network to adjust the target density map using the Gaussian Mixture Model (GMM) [30]. Other methods like BL [27] and RSI [4] focus on enhancing noise resistance by altering the loss function and convolution filter, respectively. Building upon BL, NoiseCC [41] introduces a novel loss function that lowers the weights of uncertain regions on the density map, thereby reducing the influence of noisy annotations. These methods, though effective in enhancing the model's tolerance to annotation inconsistency, are primarily involved in the training phase based on the inconsistent annotations. Integration of these methods into other frameworks may complicate the training process. Additionally, their application to localization-based methods remains unverified, as they were originally designed for density-map-based approaches. Alternatively, directly improving initial annotations presents a more efficient and broadly applicable solution.

In this paper, we aim to directly refine the inconsistent point annotations to consistent ones. Drawing inspiration from Masked Autoencoders (MAE) [12], we introduce the Shifted Autoencoders (SAE). Specifically, as illustrated in Fig. 1, MAE is trained by masking a portion of an image and reconstructing the masked image to the original unmasked one. As a form of Denoised Autoencoders [39], the MAE captures the general knowledge and discards the specific knowledge of all the training data through the reconstruction training process on a vast number of images [39]. As a result, the trained MAE tends to reconstruct generic patterns and ignore specific patterns even for previously trained images (see the reasonable overall representation of the reconstructed bus and the missing text on the bus of Fig. 1). From the aspect of reconstruction quality, the image reconstructed by MAE is not perfectly the same as the original one. **Interestingly, such a similar imperfect restoration is what we need for point annotation restoration.**

Similar to MAE [12] that applies random masks to the original image and aims to reconstruct it, we apply random 2D shifts to the initial point annotations and predict a restoration vector field to restore these shifted point annotations to their original positions. While trained on numerous similar objects and their point annotations, the SAE, akin to MAE, is compelled to capture the general positional knowledge and discard the specific manual offset noise knowledge of all the manual annotations in the training data. In the restoration phase, the trained SAE regards the initial manual point annotations as shifted points and uses learned general positional knowledge to restore them. In this way, the restored point annotations mitigate their individual manual offset noise, presenting more consistency with each other (see the bottom of Fig. 1).

We conduct extensive experiments on eleven datasets of three applications (crowd counting, remote sensing object counting, and cell counting). Compared to training with the initial point annotations, training with the ones revised by SAE steadily boosts the performance of some state-of-the-art counting methods.

The main contributions of this work are threefold: 1) We present the idea of directly refining the point annotations to be more consistent, which is beneficial for both density-map-based and localization-based object counting methods. 2) We novelly introduce the Shifted AutoEncoders (SAE), which effectively captures the general positional knowledge and ignores specific manual offset noise within the training data, yielding consistent point annotations. 3) The proposed SAE helps to set new state-of-the-art results on nine datasets.

2 Related Work

There are roughly two types of object counting methods: density-map-based methods [4,6,8–10,14,41] and localization-based methods [25,32,36]. Both types of methods heavily rely on the point annotation quality. Existing methods [2, 4, 27, 41, 45] coping with the inconsistent point annotations mainly focus on enhancing the model's tolerance to annotation noise. We will shortly review these methods in the following.

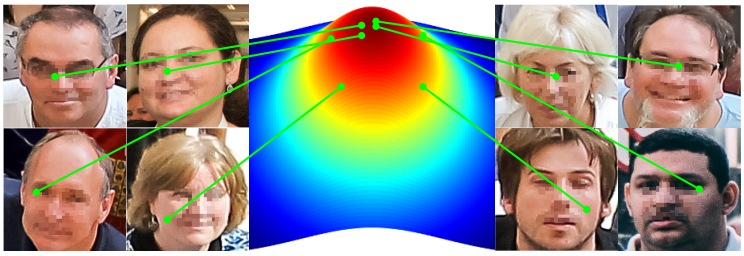

Fig. 2: Illustrative example of relative spatial distribution (approximating a Gaussian distribution [45]) of point annotations *w.r.t.* corresponding heads.

2.1 Density-Map-Based Object Counting

Density-map-based methods have emerged as the predominant approach in object counting. These methods involve creating a learning target in the form of a density map, which is constructed by applying Gaussian kernels to smooth point annotations. Models are then trained to emulate this map, with counting results derived through spatial integration over the density map. Recent improvements in this field include the creation of more complex network architectures [8, 19, 24, 48], refinement of loss functions [15, 27, 41, 43], introduction of new density map formats [6, 22, 40, 44], and the integration of scale variability factors [9, 37, 50]. While density-map-based methods have achieved significant success, they are limited in providing individual object location information.

2.2 Localization-Based Object Counting

Localization-based object counting methods [25, 32, 36] focus on directly pinpointing each target object, offering broader applicability. Early methods regard counting as an object detection problem using pseudo bounding boxes [32], and struggle in congested areas. Recent advancements like P2PNet [36] and PET [25] go beyond bounding boxes. P2PNet employs point localization through Hungarian matching [17] with fixed anchor points. In contrast, PET strategically places anchor points, further improving the counting accuracy. While localization-based methods offer detailed object localization information, they may not perform well in extremely dense areas.

2.3 Methods Focusing on Annotation Inconsistencies

ADSCNet [2] introduces a novel framework that leverages network prediction to refine target density map using Gaussian Mixture Models (GMM). Both BL [27] and RSI [4] emphasize enhancing noise resistance in their respective frameworks. In particular, BL achieves this through the introduction of an innovative loss function. RSI redesigns the convolutional filter. Building upon BL, NoiseCC [41] incorporates a loss function that reduces the loss weights of uncertain regions on the density map, mitigating the impact of noisy annotations.

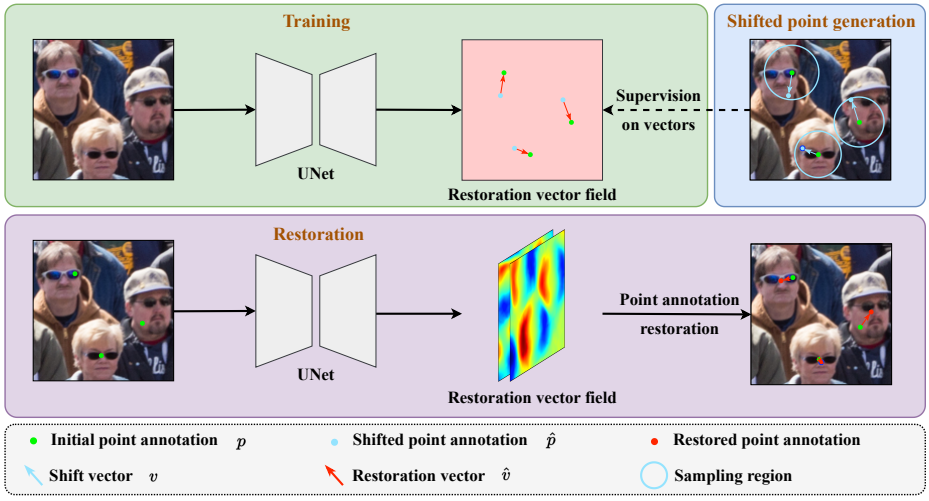

Fig. 3: The pipeline of the proposed Shifted Autoencoders (SAE), consisting of three steps: 1) Shifted point generation by adding random shift vectors to the initial annotated points; 2) Training the SAE with generated shift vectors by restoring shifted points to their original positions based on predicted restoration vector field; 3) Self-restoration that shifts the originally annotated points with corresponding restoration vectors in the predicted vector field.

These methods primarily focus on enhancing the model's tolerance to annotation noise rather than directly improving the quality of the initial point annotations. Additionally, applying these methods each time incurs additional training costs. Furthermore, these methods are primarily tailored for density-based methods, which may impose limitations on their applicability to localization-based methods. In contrast, our proposed approach focuses on directly refining initial point annotations, presenting a more efficient and broadly applicable solution.

3 Methodology

3.1 Overview

The quality of point annotation affects the training of counting models, which is vital for the final counting accuracy. Yet, the subjective decisions made by annotators in selecting the point annotation positions often lead to inconsistent annotation. Taking crowd counting as an example, the relative spatial distribution of manual point annotations *w.r.t.* corresponding objects (heads) roughly obeys a Gaussian function [45]. As shown in Fig. 2, the manual point annotations share a common distribution center (representing general position), and have specific and different offsets from this center (indicating specific manual offset noise). Inspired by Masked Autoencoders (MAE) [12], we propose the Shifted Autoen-

coders (SAE), a novel approach designed to improve the consistency of point annotations within object counting datasets before training counting models.

The workflow of SAE comprises three steps (see Fig. 3). Firstly, we generate shifted point annotations by applying random shifts to the initial point annotations. Secondly, the SAE network is trained to restore these shifted point annotations to their original positions by predicting a restoration vector field. While trained on numerous similar objects and their point annotations, the SAE, similar to MAE [12], is compelled to capture the general positional knowledge and ignores the specific manual offset noise knowledge of all the manual annotations in training data. Finally, we regard initial manual point annotations as shifted point annotations, and adopt the trained SAE to restore them toward the common distribution center, yielding more consistent annotations.

3.2 Shifted Point Annotation Generation

Given an image X in the training set, it is associated with a set of point annotations $P = \{p_i = (x_i, y_i)\}$, where x_i (*resp.* y_i) represents the x (*resp.* y) coordinate of the i-th point annotation p_i. The i ranges from 1 to N, where N denotes the total number of point annotations within this image. For each annotation point $p_i = (x_i, y_i)$ in P, we apply a random 2D shift vector v_i, which can be decomposed into two separate components: angle a_i and magnitude m_i. Each component is considered independently. Specifically, we uniformly sample from $(0, 2\pi)$ for the angle a_i. To ensure that shift vectors remain within a reasonable range like the limited masking ratio of MAE [12], we set an upper bound r_i for the magnitude m_i. For simplicity, we uniformly sample from $(0, r_i)$ for m_i. In the following, we detail how to set r_i and generate shifted point annotation.

Radius of Sampling Region for Shift Vector. As depicted in Fig. 3, the shift vector v_i is the vector pointing from the initial point annotation p_i towards the shifted point annotation $\hat{p_i}$. This means the upper bound, r_i, for the magnitude of v_i is actually the radius of the circular region within which the $\hat{p_i}$ is randomly sampled. Naturally, the radius r_i is expected to equal the scope radius of the corresponding object in the ideal case.

Object counting annotations are typically in the form of two-dimensional points, which provide no direct information about the sizes or shapes of the corresponding objects. These 2D point annotations can only provide information about the spatial distribution of these objects and the distances between them on the image plane. For object counting, some density-map-based methods [19, 50] approximate the size of objects of the same type based on the distances between them. Following them, a straightforward solution to roughly determine the size of objects of the same type is to leverage the distance information between them. Specifically, for each point annotation p_i in P, we denote the Euclidean distance to its nearest neighbor as d_i. As stated in [50], the pixel associated with the i-th object corresponds to an area roughly of a radius proportional to d_i Therefore, a simple choice to set the radius r_i^s is given by:

$$r_i^s = \alpha \times d_i, \tag{1}$$

(a) Sampling regions with $r = \alpha \times d$ **(b)** Sampling regions with $r = \alpha \times min(d, \overline{d}^{\mathcal{N}_3})$

Fig. 4: Illustration of different strategies to define the radius of sampling region for each annotated point in crowd images. See Eq. (1) and Eq. (2) and corresponding text for more details. The hyper-parameter α is set to 0.4. Best viewed by zooming in on the electronic version.

where α is a hyper-parameter which is constrained not to exceed 0.5. As illustrated in Fig. 4a, this 0.5 constraint prevents the overlapping of sampling regions with neighboring point annotations, which may bring confusion on the target point p_i, towards which the shifted point $\hat{p}_i$ should be restored.

This simple setting for the radius r_i is acceptable to approximate the size for most objects (see Fig. 4a). However, for some objects that are distributed sparsely, the radius obtained based on Eq. (1) would be significantly oversized. This is due to their distances to the corresponding nearest neighbors being much greater than their spatial size. As illustrated in the white rectangle of Fig. 4a, an oversized sampling radius will introduce an excessive amount of background to the sampling region. For these objects, the proposed SAE will be forced to consider the background areas far from the objects as a part of them, which compromises the training of SAE. To mitigate this, we comprehensively take into account the size information of neighboring objects of p_i. Specifically, the final radius r_i for the annotation point p_i is defined by:

$$r_i = \alpha \times min(d_i, \overline{d}_i^{\mathcal{N}_3}), \tag{2}$$

where $\overline{d}_i^{\mathcal{N}_3}$ denotes the mean value of d for the three nearest neighboring annotation points of p_i. By combining d of neighboring point annotations, the sampling regions of the sparsely distributed point annotations are better aligned with the spatial scales of their objects. An illustration example is shown in the white rectangle of Fig. 4b.

Shift Vector and Shifted Point Annotation. With the angle a_i and the magnitude m_i, the 2D shift vector v_i for the annotation point p_i can be generated as below:

$$v_i = \left(v_i^x, v_i^y\right) = \left(m_i \times \cos a_i, m_i \times \sin a_i\right) \tag{3}$$

By imposing v_i to the corresponding initial point annotation p_i, the shifted point annotation $\hat{p_i}$ is obtained as follows:

$$\hat{p_i} = (\hat{x_i}, \hat{y_i}) = (x_i + v_i^x, y_i + v_i^y) \tag{4}$$

3.3 SAE Network Training

SAE Network Architecture. In our SAE network, we employ a lightweight UNet architecture [31], incorporating a VGG16 [33] backbone, to predict the restoration vector field $F \in \mathbb{R}^{2\times H\times W}$. This field consists of two channels, $F_x \in \mathbb{R}^{H\times W}$ and $F_y \in \mathbb{R}^{H\times W}$, corresponding to the x and y axes, respectively. Here, H and W denote the height and the width of the input image, respectively.

Training Objective. In our SAE, we aim to restore the shifted point annotations $\{\hat{p_i}\}$ generated in Sec. 3.2 to their corresponding initial positions $\{p_i\}$. In alignment with the approach adopted in MAE [12], we adopt the Mean Squared Error (MSE) loss to train our SAE network. Note that we only compute the MSE loss with F on the coordinates of shifted point annotations $\{\hat{p_i}\}$. Specifically, the restoration shift vectors $\{\hat{v_i} = \big(F_x(\hat{p_i}), F_y(\hat{p_i})\big)\}$ are supervised to align with the inverse of the corresponding shift vectors $\{v_i\}$. Formally, the training objective $\mathcal{L}$ for our SAE is given by:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \left\| \hat{v_i} - (-v_i) \right\|^2, \tag{5}$$

where N refers to the total number of annotated points in a given image within the training set.

After training on a variety of similar objects and their corresponding point annotations, the SAE, similar to MAE [12], captures the general positional knowledge while discarding the specific manual offset noise of all the manual point annotations in the training data. Indeed, imaging a synthetic case where the same head repeats multiple times with different point annotations spreading around a fixed position (*e.g.*, head center), the optimal choice to simultaneously cope with the varied annotations is to restore them to the same fixed position. In practice, optimizing the SAE network on a large number of objects also tends to restore the manual annotations towards the distribution center, yielding more consistent annotation.

3.4 Point Annotation Restoration

The trained SAE network is then utilized to revise the initial point annotations. More precisely, each image from the training set is processed through the trained SAE network to generate the corresponding 2D restoration vector field map F. The initial point annotations are treated as shifted point annotations and are restored by the predicted restoration vector field map in a manner analogous to the training phase. Specifically, the coordinate of the initial point annotation, p_i, is used to sample the corresponding restoration vector $\hat{v_i}$ within F. The final

restored point annotations are obtained by adding the restoration vectors to the corresponding initial point annotations. Formally, the set of restored point annotations P_r is given by:

$$P_r = \{(x_i + \hat{v}_i^x, y_i + \hat{v}_i^y)\} = \{\big(x_i + F_x(p_i), y_i + F_y(p_i)\big)\} \tag{6}$$

Thanks to the captured general positional knowledge and avoidance of specific manual offset noise of our SAE, the restored point annotations have better consistency among themselves. Compared with training with the initial point annotations, using more consistent point annotations introduces less confusion during the training of counting models, resulting in improved counting accuracy.

4 Experiments

4.1 Experimental Setting

Datasets. To evaluate the effectiveness of the proposed approach, we perform extensive experiments on eleven datasets spanning three domains: crowd counting, remote sensing object counting, and cell counting. These datasets include: SH PartA [50], SH PartB [50], UCF-QNRF [15], JHU++ [35], RSOC_building [9], RSOC_small-vehicle [9], RSOC_large-vehicle [9], RSOC_ship [9], MBM [16], ADI [29], and DCC [28].

Evaluation Metrics. In object counting tasks, the Mean Absolute Error (MAE) and Mean Square Error (MSE) serve as the principal evaluation metrics. Lower values for these metrics indicate better counting performance.

Implementation Details. During the training phase for SAE, we augment the images with random scaling, horizontal flipping, and randomly cropping to 512×512 pixels (except for the ADI dataset [29] which uses 128×128 pixels due to limited resolution). Shift vectors for point annotations are randomly regenerated at each iteration. We employ the Adam optimizer with a weight decay of 5×10^{-4} and a fixed learning rate of 1×10^{-4}. SAE training proceeds for 100 epochs with batches of 8 for each dataset on an RTX 3090 GPU using the PyTorch framework. To ensure a fair comparison, we train the SAE on the training set of each dataset individually, thereby avoiding the introduction of extra data. It is noteworthy that the proposed **SAE plays the role only in the point annotation restoration phase, rather than the object counting phase**. During the training phase for object counting, we follow the same implementation details as the baseline counting methods. The notation & SAE below indicates that the counting methods are trained using point annotations revised by SAE. Otherwise, they are trained with the initial manual point annotations.

4.2 Experimental Results

We conduct extensive object counting experiments on eleven publicly available datasets across three distinct domains: crowd counting, remote sensing object counting, and cell counting. Firstly, Fig. 5 presents some qualitative results of

Fig. 5: Visualization of restored point annotation given by the proposed SAE. The bottom row provides a zoomed-in view within the white box of the top row. Green points: initial point annotations; Red points: revised point annotations; Yellow points: initial point annotation coincides with the corresponding revised point annotation.

restored point annotation given by our SAE, illustrating the SAE's effectiveness in enhancing the consistency of point annotations for various object types. To further quantify SAE's effectiveness, we conduct quantitative comparisons in terms of object counting accuracy by training some existing counting methods (including both density-map-based and localization-based approaches) with initial annotations and restored annotations by SAE, respectively. The detailed quantitative comparison is given in the following.

Crowd Counting. We mainly evaluate SAE on two types of counting methods.

Comparison with State-of-the-Art Methods. We conduct a comprehensive evaluation of the SAE-refined point annotations compared with the initial ones, employing a variety of SOTA density-map-based and localization-based crowd counting models. The quantitative results, depicted in Tab. 1, demonstrate the effectiveness of SAE on various baseline methods. A notable improvement is observed when applied to P2PNet [36], where SAE achieves a reduction of 5.8 in MAE and 8.4 in MSE on UCF-QNRF dataset. Remarkably, the integration of SAE leads to the establishment of new state-of-the-art results in three of the four assessed datasets: SH PartA [50], SH PartB [50], and JHU++ [35]. On average, SAE contributes to a reduction of 3.0 in MAE and 6.4 in MSE compared to the results of three baseline methods (P2PNet [36], MAN [24], and STEERER [11]) on the four extensively used datasets.

Effectiveness on Anti-noise Methods. Further investigations are conducted to assess SAE's compatibility with those anti-noise approaches: BL [27], NoiseCC [41], and RSI [4]. While these methods primarily focus on improving noise resistance, they do not inherently improve the quality of the initial point annotations. As depicted in Tab. 1, SAE further steadily improves these anti-noise frameworks,

Table 1: Quantitative comparison of crowd counting results on the ShanghaiTech [50], UCF-QNRF [15], and JHU-Crowd++ [35] datasets. & SAE indicates training with point annotations refined by SAE. The best performance is in **boldface**. † represents reproduced results with official codes.

Method	SH PartA		SH PartB		UCF-QNRF		JHU-Crowd++	
	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓
CSRNet [19] (CVPR'18)	68.2	115.0	10.6	16.0	-	-	85.9	309.2
CAN [26] (CVPR'19)	62.3	100.0	7.8	12.2	107.0	183.0	100.1	314.0
ADSCNet [2] (CVPR'20)	55.4	97.7	6.4	11.3	**71.3**	132.5	-	-
GL [43] (CVPR'21)	61.3	95.4	7.3	11.7	84.3	147.5	59.9	259.5
CLTR [21] (ECCV'22)	56.9	95.2	6.5	10.6	85.8	141.3	59.5	240.6
NoiseCC [45] (TPAMI'23)	61.8	104.3	7.1	12.4	83.8	147.8	59.1	259.6
NoiseCC [45] & MAN (TPAMI'23)	56.4	89.4	6.5	10.3	75.3	128.3	53.0	208.6
CrowdHat [46] (CVPR'23)	51.2	81.9	5.7	9.4	75.1	126.7	52.3	211.8
AWCCNet [14] (ICCV'23)	56.2	91.3	-	-	76.4	130.5	52.3	207.2
PET [25] (ICCV'23)	49.3	78.8	6.2	9.7	79.5	144.3	58.5	238.0
Gramformer [23] (AAAI'24)	54.7	87.1	-	-	76.7	129.5	53.1	228.1
BL† [27] (ICCV'19)	62.7	99.5	7.6	13.00	87.4	149.6	74.4	290.0
BL & SAE	59.5 (↓3.2)	89.4 (↓10.1)	6.9 (↓0.7)	11.9 (↓1.1)	81.6 (↓5.8)	146.5 (↓3.1)	60.8 (↓13.6)	230.7 (↓59.3)
NoiseCC† [41] (NeurIPS'20)	62.1	100.0	7.5	11.4	86.1	149.7	67.5	255.4
NoiseCC & SAE	59.4 (↓2.7)	91.4 (↓8.6)	6.8 (↓0.7)	9.8 (↓1.6)	81.9 (↓4.2)	134.4 (↓15.3)	58.9 (↓8.6)	231.8 (↓23.6)
RSI-ResNet50† [4] (CVPR'22)	54.4	89.0	6.6	9.8	81.2	152.0	58.8	245.1
RSI-ResNet50 & SAE	52.4 (↓2.0)	85.4 (↓3.6)	6.0 (↓0.6)	9.2 (↓0.6)	77.5 (↓3.7)	146.2 (↓5.8)	54.9 (↓3.9)	240.2 (↓4.9)
P2PNet† [36] (ICCV'21)	52.8	85.8	6.5	10.9	91.7	157.0	66.8	259.5
P2PNet & SAE	**48.2** (↓4.6)	**76.1** (↓9.7)	6.2 (↓0.3)	10.0 (↓0.9)	85.9 (↓5.8)	148.6 (↓8.4)	62.4 (↓4.4)	253.7 (↓5.8)
MAN† [24] (CVPR'22)	55.6	93.2	7.1	10.5	77.5	132.7	53.2	219.9
MAN & SAE	52.2 (↓3.4)	81.9 (↓11.3)	**5.4** (↓1.7)	**7.0** (↓3.5)	74.2 (↓3.3)	128.2 (↓4.5)	**49.7** (↓3.5)	**204.4** (↓15.5)
STEERER† [11] (ICCV'23)	56.5	89.8	7.1	10.7	74.1	129.5	55.8	223.2
STEERER & SAE	54.3 (↓2.2)	84.5 (↓5.3)	6.4 (↓0.7)	10.1 (↓0.6)	71.4 (↓2.7)	**125.1** (↓4.4)	52.9 (↓2.9)	216.4 (↓6.8)

yielding an average enhancement of 4.1 in MAE and 11.5 in MSE across the four diverse crowd counting datasets.

Remote sensing object counting. We mainly evaluate SAE on two types of counting methods.

Comparison with State-of-the-Art Methods. In contrast to crowd counting datasets, remote sensing object counting task presents a different challenge with their diversity, encompassing four distinct types of objects. We apply SAE to several state-of-the-art remote sensing object counting methods. The quantitative results, detailed in Tab. 2, clearly demonstrate the effective performance of SAE. It not only consistently outperforms baseline methods but also sets new state-of-the-art results in three of the four datasets: RSOC_Small-vehicle, RSOC_Large-vehicle, and RSOC_Ship [9]. Notably, SAE achieves, on average, an improvement of 9.2 in MAE and 40.6 in MSE over the three baseline models (P2PNet [36], ASPDNet [9], and PSGCNet [8]) across these four datasets.

Effectiveness on Anti-noise Methods. In addition to state-of-the-art approaches, we also examine the effectiveness of our SAE on two anti-noise methods, BL [27] and NoiseCC [41]. These anti-noise methods aim to enhance the model's noise resilience, primarily through adjustments in the loss functions. The comparative results shown in Tab. 2 reveal that SAE further augments the performance for these anti-noise methods, resulting in an average improvement of 10.2 in MAE and 51.6 in MSE across the four datasets.

Cell counting. We mainly evaluate SAE on two types of counting methods.

Comparison with State-of-the-Art Methods. The proposed SAE method extends its utility beyond natural image datasets. We also demonstrate its ability to

Table 2: Quantitative comparison of remote sensing object counting results on RSOC datasets [9]. & SAE indicates training with point annotations refined by SAE. The best performance is in **boldface**. † represents reproduced results with official codes.

Method	RSOC_Building		RSOC_Small-vehicle		RSOC_Large-vehicle		RSOC_Ship	
	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓
MCFA [7] (TGRS'21)	7.9	11.8	238.5	625.9	12.9	20.3	50.5	65.2
ADMAL [6] (TGRS'22)	**5.6**	**7.7**	115.6	210.8	11.7	17.3	45.1	64.8
eFreeNet [13] (TGRS'23)	**5.6**	**7.7**	195.9	463.6	14.6	19.8	65.3	85.5
LMSFFNet [49] (TGRS'23)	6.3	9.4	141.7	273.0	12.7	27.1	49.5	70.0
DOPNet [5] (TGRS'24)	-	-	62.4	**167.8**	12.5	20.1	-	-
BL† [27] (ICCV'19)	11.5	16.3	173.0	477.8	12.3	24.2	58.8	195.5
BL & SAE	11.1 (↓0.4)	15.9 (↓0.4)	130.6 (↓42.4)	341.4 (↓136.4)	8.5 (↓3.8)	14.6 (↓9.6)	51.0 (↓7.8)	73.6 (↓121.9)
NoiseCC† [41] (NeurIPS'20)	7.8	11.3	168.8	529.1	15.5	31.4	53.0	72.1
NoiseCC & SAE	7.3 (↓0.5)	10.2 (↓1.1)	149.7 (↓19.1)	389.4 (↓139.7)	14.6 (↓0.9)	30.8 (↓0.6)	46.3 (↓6.7)	69.0 (↓3.1)
P2PNet† [36] (ICCV'21)	6.3	9.1	63.1	198.3	8.1	13.2	28.2	42.6
P2PNet & SAE	5.7 (↓0.6)	8.1 (↓1.0)	**53.3** (↓9.8)	170.4 (↓27.9)	**7.0** (↓1.1)	**11.6** (↓1.6)	**25.0** (↓3.2)	**39.5** (↓3.1)
ASPDNet† [9] (TGRS'20)	7.6	11.5	252.6	718.5	19.7	27.8	81.8	110.3
ASPDNet & SAE	6.7 (↓0.9)	10.6 (↓0.9)	182.0 (↓70.6)	353.4 (↓365.1)	16.9 (↓2.8)	24.4 (↓3.4)	75.1 (↓6.7)	97.0 (↓13.3)
PSGCNet† [8] (TGRS'22)	7.4	11.1	112.1	289.6	11.8	16.4	39.5	68.5
PSGCNet & SAE	6.6 (↓0.8)	10.2 (↓0.9)	104.9 (↓7.2)	227.2 (↓62.4)	9.7 (↓2.1)	14.0 (↓2.4)	34.6 (↓4.9)	62.8 (↓5.7)

Table 3: Quantitative comparison of cell counting results on MBM [16], ADI [29], and DCC [28]. & SAE indicates training with point annotations refined by SAE. The best performance is in **boldface**. † represents reproduced results with official codes.

Method	MBM		ADI		DCC	
	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓
BL† [27] (ICCV'19)	8.4	10.3	13.7	18.8	3.3	5.3
BL & SAE	6.3 (↓ 2.1)	8.8 (↓ 1.5)	13.2 (↓ 0.5)	18.2 (↓ 0.6)	2.9 (↓ 0.4)	4.9 (↓ 0.4)
NoiseCC† [41] (NeurIPS'20)	7.2	9.3	14.0	18.8	3.8	5.6
NoiseCC & SAE	6.0 (↓ 1.2)	8.6 (↓ 0.7)	13.1 (↓ 0.9)	17.9 (↓ 0.9)	3.4 (↓ 0.4)	5.1 (↓ 0.5)
P2PNet† [36] (ICCV'21)	5.7	8.0	12.4	17.1	3.7	6.0
P2PNet & SAE	4.4 (↓ 1.3)	6.5 (↓ 1.5)	11.9 (↓ 0.5)	16.2 (↓ 0.9)	3.2 (↓ 0.5)	5.6 (↓ 0.4)
Chen *et al.* † [3] (JBHI'21)	5.6	8.3	12.7	19.6	5.7	7.6
Chen *et al.* & SAE	4.5 (↓ 1.1)	7.5 (↓ 0.8)	11.4 (↓ 1.3)	15.8 (↓ 3.8)	4.7 (↓ 1.0)	7.1 (↓ 0.5)
SAUNet† [10] (IEEE ACM T COMPUT BI'21)	5.7	7.7	14.3	18.5	3.0	4.8
SAUNet & SAE	**4.2** (↓ 1.5)	**5.8** (↓ 1.9)	**11.2** (↓ 3.1)	**15.1** (↓ 3.4)	**2.6** (↓ 0.4)	**3.4** (↓ 1.4)

resolve annotation inconsistency challenges encountered in cell counting. As detailed in Tab. 3, the proposed SAE consistently improves cell counting accuracy. In particular, we establish new state-of-the-art records across all the three cell counting datasets. Notably, with SAU [10] already achieving excellent results on DCC [28] with an MAE of 3.0 and an MSE of 4.8, using the restored point annotation by SAE further enhances the performance, reducing the MAE to 2.6 and MSE to 3.4, respectively.

Effectiveness on Anti-noise Methods. Additionally, we also explore SAE's universal applicability in conjunction with two established anti-noise methods: BL [27] and NoiseCC [41]. These two methods focus on improving noise robustness by introducing novel loss functions. As shown in Tab. 3, the proposed SAE consistently improves their performance, which further validates the effectiveness and universality of the proposed SAE.

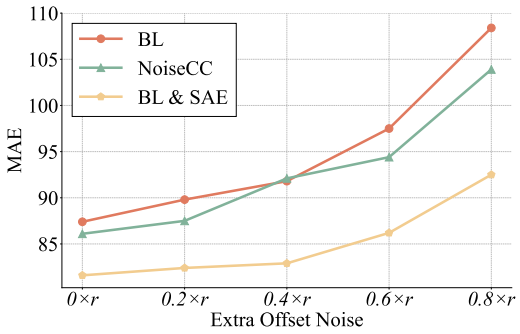

Fig. 6: Evaluation of counting robustness to different levels of synthetic offset noise on the UCF-QNRF [15] dataset. We shift the original point annotation by different proportions of radius r (defined in Eq. (2)) in random directions to synthesize annotation noise of different levels.

4.3 Robustness to Extra Point Annotation Noise

We also evaluate the robustness of our SAE to point annotation noise of different degrees. For that, Firstly, we emulate real-world human annotation noise errors by introducing varying offsets to each annotation point p. The magnitude of offsets are set to $0.2 \times r$, $0.4 \times r$, $0.6 \times r$, and $0.8 \times r$ pixels, where r is defined in Eq. (2). Secondly, we train two anti-noise crowd counting methods: BL [27] and NoiseCC [41] on these noisy point annotations. We then apply our SAE to restore the noisy point annotations, and train a new BL on our restored point annotations. These experiments are conducted on the UCF-QNRF [15] dataset.

As shown in Fig. 6, increasing noise levels of annotation poses challenges to all methods in maintaining counting accuracy. Remarkably, SAE consistently outperforms the other two techniques and is less affected by increasing noise levels (in particular for extra noise $\leq 0.4 \times r$), achieving lower MAE. These findings confirm SAE's effectiveness and robustness in handling noisy and imperfect point annotation, a common occurrence in real-world counting tasks.

4.4 Ablation Study on Hyper-parameter α in Eq. (2)

The proposed SAE mainly involves one hyper-parameter α, which is involved in Eq. (2) and plays a crucial role in defining the sampling area for shifted point generation. To evaluate the impact of the hyper-parameter α on our model's performance, we conduct an ablation study with α values set to 0.3, 0.4, 0.5, and 0.6. This study utilizes BL [27] as the baseline counting method, and covers three diverse datasets: UCF-QNRF [15], Remote_small-vehicle [9], and MBM [16].

The results, presented in Tab. 4, show how different α values influence counting accuracy. Generally, in configurations that avoid overlap, we observe performance improvements compared to the baseline method. For α set to 0.6,

Table 4: Quantitative results of BL [27] (on UCF-QNRF [15], RSOC_Small-vehicle (RSV) [9], and MBM [16] dataset) trained with point annotations refined by SAE with different values of α involved in Eq. (2). The best performance is in boldface.

α	UCF-QNRF		RSV		MBM	
	MAE↓	MSE↓	MAE↓	MSE↓	MAE↓	MSE↓
Baseline	87.4	149.6	173.0	477.8	8.4	10.3
0.3	86.5	148.3	149.7	389.4	7.3	9.2
0.4	**81.6**	146.5	**130.6**	**341.4**	**6.3**	8.8
0.5	82.4	**146.3**	138.2	364.8	6.9	**8.6**
0.6	91.0	158.9	190.6	516.5	8.3	10.8

the overlapping sampling ranges for different objects introduce confusion in the training process, leading to performance that falls short of the baseline. Indeed, a smaller value of α leads to a more constrained sampling range, which may reduce the SAE model's effectiveness in refining annotations with large deviations from their general positions. In contrast, a larger α tends to include more background regions within the sampling range. Notably, when the sampling areas for different annotation points overlap, it leads to ambiguity. This overlap challenges SAE's ability to distinguish between regions associated with distinct objects. Except for this ablation study, we set α to 0.4 for our SAE in all experiments.

4.5 Limitation

As described in Sec. 3.2, the proposed SAE approximates the radius of the sampling region based on the spatial distribution of objects. While this strategy is functional and simple, there could be more effective strategies for determining the optimal radius, such as depth estimation. Nevertheless, these alternatives might complicate the proposed method and compromise its universality.

5 Conclusion

In this paper, we focus on the inconsistency problem of point annotations in object counting tasks. This is often caused by the inevitable subjective nature of annotators. To mitigate this issue, we propose the novel Shifted Autoencoders (SAE) to revise the initial point annotations. Similar to MAE which uses general knowledge to reconstruct masked areas, resulting in specific-detail-free image reconstruction, our SAE leverages general positional knowledge to restore shifted annotations, yielding specific-offset-noise-free point restoration. Using such restored consistent point annotation to train the counting model improves the counting accuracy. Extensive experiments on eleven datasets from three different object counting tasks verify the effectiveness of the proposed SAE. Besides, based on the proposed SAE, we set new state-of-the-art results on nine of the eleven datasets. We hope that this work could shed light on the research direction of directly refining the annotation in point-based vision tasks.

Acknowledgements

This work was supported in part by the NSFC 62222112, 62225113, and 62176186, and the Innovative Research Group Project of Hubei Province (2024AFA017).

References

1. Arteta, C., Lempitsky, V., Noble, J.A., Zisserman, A.: Interactive object counting. In: Eur. Conf. Comput. Vis. pp. 504–518 (2014)
2. Bai, S., He, Z., Qiao, Y., Hu, H., Wu, W., Yan, J.: Adaptive dilated network with self-correction supervision for counting. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 4594–4603 (2020)
3. Chen, Y., Liang, D., Bai, X., Xu, Y., Yang, X.: Cell localization and counting using direction field map. IEEE Journal of Biomedical and Health Informatics **26**(1), 359–368 (2021)
4. Cheng, Z.Q., Dai, Q., Li, H., Song, J., Wu, X., Hauptmann, A.G.: Rethinking spatial invariance of convolutional networks for object counting. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 19638–19648 (2022)
5. Cui, M., Ding, G., Yang, D., Chen, Z.: Dopnet: Dense object prediction network for multiclass object counting and localization in remote sensing images. IEEE Transactions on Geoscience and Remote Sensing **62**, 1–15 (2024)
6. Ding, G., Cui, M., Yang, D., Wang, T., Wang, S., Zhang, Y.: Object counting for remote-sensing images via adaptive density map-assisted learning. IEEE Transactions on Geoscience and Remote Sensing **60**, 1–11 (2022)
7. Duan, Z., Wang, S., Di, H., Deng, J.: Distillation remote sensing object counting via multi-scale context feature aggregation. IEEE Transactions on Geoscience and Remote Sensing **60**, 1–12 (2021)
8. Gao, G., Liu, Q., Hu, Z., Li, L., Wen, Q., Wang, Y.: Psgcnet: A pyramidal scale and global context guided network for dense object counting in remote-sensing images. IEEE Transactions on Geoscience and Remote Sensing **60**, 1–12 (2022)
9. Gao, G., Liu, Q., Wang, Y.: Counting from sky: A large-scale data set for remote sensing object counting and a benchmark method. IEEE Transactions on Geoscience and Remote Sensing **59**(5), 3642–3655 (2020)
10. Guo, Y., Krupa, O., Stein, J., Wu, G., Krishnamurthy, A.: Sau-net: A unified network for cell counting in 2d and 3d microscopy images. IEEE/ACM Transactions on Computational Biology and Bioinformatics **19**(4), 1920–1932 (2022)
11. Han, T., Bai, L., Liu, L., Ouyang, W.: Steerer: Resolving scale variations for counting and localization via selective inheritance learning. In: Int. Conf. Comput. Vis. pp. 21848–21859 (2023)
12. He, K., et al.: Masked autoencoders are scalable vision learners. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 16000–16009 (2022)
13. Huang, Y., Jin, Y., Zhang, L., Liu, Y.: Remote sensing object counting through regression ensembles and learning to rank. IEEE Transactions on Geoscience and Remote Sensing **61**, 1–17 (2023)
14. Huang, Z.K., Chen, W.T., Chiang, Y.C., Kuo, S.Y., Yang, M.H.: Counting crowds in bad weather. In: Int. Conf. Comput. Vis. pp. 23308–23319 (2023)
15. Idrees, H., Tayyab, M., Athrey, K., Zhang, D., Al-Maadeed, S., Rajpoot, N., Shah, M.: Composition loss for counting, density map estimation and localization in dense crowds. In: Eur. Conf. Comput. Vis. pp. 532–546 (2018)

16. Kainz, P., Urschler, M., Schulter, S., Wohlhart, P., Lepetit, V.: You should use regression to detect cells. In: Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervention. pp. 276–283 (2015)
17. Kuhn, H.W.: The hungarian method for the assignment problem. Naval Research Logistics Quarterly **2**(1-2), 83–97 (1955)
18. Li, T., Chang, H., Wang, M., Ni, B., Hong, R., Yan, S.: Crowded scene analysis: A survey. IEEE Trans. Circuit Syst. Video Technol. **25**(3), 367–386 (2014)
19. Li, Y., Zhang, X., Chen, D.: Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 1091–1100 (2018)
20. Liang, D., Xie, J., Zou, Z., Ye, X., Xu, W., Bai, X.: Crowdclip: Unsupervised crowd counting via vision-language model. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 2893–2903 (2023)
21. Liang, D., Xu, W., Bai, X.: An end-to-end transformer model for crowd localization. In: Eur. Conf. Comput. Vis. pp. 38–54 (2022)
22. Liang, D., Xu, W., Zhu, Y., Zhou, Y.: Focal inverse distance transform maps for crowd localization. IEEE Trans. Multimedia **25**, 6040–6052 (2023)
23. Lin, H., Ma, Z., Hong, X., Shangguan, Q., Meng, D.: Gramformer: Learning crowd counting via graph-modulated transformer. In: AAAI. pp. 3395–3403 (2024)
24. Lin, H., Ma, Z., Ji, R., Wang, Y., Hong, X.: Boosting crowd counting via multifaceted attention. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 19628–19637 (2022)
25. Liu, C., Lu, H., Cao, Z., Liu, T.: Point-query quadtree for crowd counting, localization, and more. In: Int. Conf. Comput. Vis. pp. 1676–1685 (2023)
26. Liu, W., Salzmann, M., Fua, P.: Context-aware crowd counting. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 5099–5108 (2019)
27. Ma, Z., Wei, X., Hong, X., Gong, Y.: Bayesian loss for crowd count estimation with point supervision. In: Int. Conf. Comput. Vis. pp. 6142–6151 (2019)
28. Marsden, M., McGuinness, K., Little, S., Keogh, C.E., O'Connor, N.E.: People, penguins and petri dishes: Adapting object counting models to new visual domains and object types without forgetting. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 8070–8079 (2018)
29. Paul Cohen, J., Boucher, G., Glastonbury, C.A., Lo, H.Z., Bengio, Y.: Count-ception: Counting by fully convolutional redundant counting. In: Int. Conf. Comput. Vis. Worksh. pp. 18–26 (2017)
30. Reynolds, D.A., et al.: Gaussian mixture models. Encyclopedia of biometrics **741**(659-663) (2009)
31. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Proc. Int. Conf. Med. Image Comput. Comput.-Assisted Intervention. pp. 234–241 (2015)
32. Sam, D.B., Peri, S.V., Sundararaman, M.N., Kamath, A., Babu, R.V.: Locate, size, and count: Accurately resolving people in dense crowds via detection. IEEE Trans. Pattern Anal. Mach. Intell. **43**(8), 2739–2751 (2020)
33. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Int. Conf. Learn. Represent. pp. 1–14 (2015)
34. Sindagi, V.A., Patel, V.M.: Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting. In: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). pp. 1–6 (2017)

35. Sindagi, V.A., Yasarla, R., Patel, V.M.: Jhu-crowd++: Large-scale crowd counting dataset and a benchmark method. IEEE Trans. Pattern Anal. Mach. Intell. **44**(5), 2594–2609 (2022)
36. Song, Q., Wang, C., Jiang, Z., Wang, Y., Tai, Y., Wang, C., Li, J., Huang, F., Wu, Y.: Rethinking counting and localization in crowds: A purely point-based framework. In: Int. Conf. Comput. Vis. pp. 3365–3374 (2021)
37. Song, Q., Wang, C., Wang, Y., Tai, Y., Wang, C., Li, J., Wu, J., Ma, J.: To choose or to fuse? scale selection for crowd counting. In: AAAI. pp. 2576–2583 (2021)
38. Sun, G., An, Z., Liu, Y., Liu, C., Sakaridis, C., Fan, D.P., Van Gool, L.: Indiscernible object counting in underwater scenes. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 13791–13801 (2023)
39. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: Int. Conf. Mach. Learn. pp. 1096–1103 (2008)
40. Wan, J., Chan, A.: Adaptive density map generation for crowd counting. In: Int. Conf. Comput. Vis. pp. 1130–1139 (2019)
41. Wan, J., Chan, A.: Modeling noisy annotations for crowd counting. In: Adv. Neural Inform. Process. Syst. vol. 33, pp. 3386–3396 (2020)
42. Wan, J., Kumar, N.S., Chan, A.B.: Fine-grained crowd counting. IEEE Trans. Image Process. **30**, 2114–2126 (2021)
43. Wan, J., Liu, Z., Chan, A.B.: A generalized loss function for crowd counting and localization. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 1974–1983 (2021)
44. Wan, J., Wang, Q., Chan, A.B.: Kernel-based density map generation for dense object counting. IEEE Trans. Pattern Anal. Mach. Intell. **44**(3), 1357–1370 (2020)
45. Wan, J., Wu, Q., Chan, A.B.: Modeling noisy annotations for point-wise supervision. IEEE Trans. Pattern Anal. Mach. Intell. **45**(12), 15065–15080 (2023)
46. Wu, S., Yang, F.: Boosting detection in crowd analysis via underutilized output features. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 15609–15618 (2023)
47. Xu, C., Liang, D., Xu, Y., Bai, S., Zhan, W., Bai, X., Tomizuka, M.: Autoscale: Learning to scale for crowd counting. Int. J. Comput. Vis. **130**(2), 405–434 (2022)
48. Xu, C., Qiu, K., Fu, J., Bai, S., Xu, Y., Bai, X.: Learn to scale: Generating multipolar normalized density maps for crowd counting. In: Int. Conf. Comput. Vis. pp. 8382–8390 (2019)
49. Yi, J., Shen, Z., Chen, F., Zhao, Y., Xiao, S., Zhou, W.: A lightweight multiscale feature fusion network for remote sensing object counting. IEEE Transactions on Geoscience and Remote Sensing **61**, 1–13 (2023)
50. Zhang, Y., Zhou, D., Chen, S., Gao, S., Ma, Y.: Single-image crowd counting via multi-column convolutional neural network. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 589–597 (2016)