UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

Focus for Free in Density-Based Counting

Shi, Z.; Mettes, P. ; Snoek, C.G.M.

DOI
10.1007/s11263-024-01990-3
Publication date
2024
Document Version
Final published version
Published in
International Journal of Computer Vision

License
Article 25fa Dutch Copyright Act (https://www.openaccess.nl/en/policies/open-access-in-dutch-copyright-law-taverne-amendment)
Link to publication

Citation for published version (APA):
Shi, Z., Mettes, P., & Snoek, C. G. M. (2024). Focus for Free in Density-Based Counting. *International Journal of Computer Vision*, *132*(7), 2600-2617. https://doi.org/10.1007/s11263-024-01990-3

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:12 May 2025

International Journal of Computer Vision (2024) 132:2600–2617
https://doi.org/10.1007/s11263-024-01990-3

Focus for Free in Density-Based Counting

Zenglin Shi[1] · **Pascal Mettes**[2] · **Cees G. M. Snoek**[2]

Received: 8 June 2023 / Accepted: 1 January 2024 / Published online: 9 February 2024
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

Abstract
This work considers supervised learning to count from images and their corresponding point annotations. Where density-based counting methods typically use the point annotations only to create Gaussian-density maps, which act as the supervision signal, the starting point of this work is that point annotations have counting potential beyond density map generation. We introduce two methods that repurpose the available point annotations to enhance counting performance. The first is a counting-specific augmentation that leverages point annotations to simulate occluded objects in both input and density images to enhance the network's robustness to occlusions. The second method, foreground distillation, generates foreground masks from the point annotations, from which we train an auxiliary network on images with blacked-out backgrounds. By doing so, it learns to extract foreground counting knowledge without interference from the background. These methods can be seamlessly integrated with existing counting advances and are adaptable to different loss functions. We demonstrate complementary effects of the approaches, allowing us to achieve robust counting results even in challenging scenarios such as background clutter, occlusion, and varying crowd densities. Our proposed approach achieves strong counting results on multiple datasets, including ShanghaiTech Part_A and Part_B, UCF_QNRF, JHU-Crowd++, and NWPU-Crowd. Code is available at https://github.com/shizenglin/Counting-with-Focus-for-Free.

Keywords Object counting · Deep learning · Vision transformer

1 Introduction

The human ability to count is an intriguing cognitive skill that enables us to quantify and comprehend the world. From basic distinctions of "one" and "many", our counting abilities have evolved into intricate systems like the decimal system. This system equips us with the capacity to represent numbers of infinite magnitude, enabling us to count a plethora of entities, such as species in an ecosystem, stars in the celestial expanse, and buildings in bustling cityscapes. Counting also plays a significant role in computer vision.

In the realm of computer vision, counting objects or identifying the number of specific elements within images or videos is a fundamental task. Computer vision algorithms have employed a variety of approaches to accomplish this feat, such as detection, clustering, and regression. Counting by detection (Lin et al., 2001; Leibe et al., 2005; Wu and Nevatia, 2007; Li et al., 2008; Topkaya et al., 2014) counts objects by detecting them individually, which is effective when the number of objects is limited, but becomes slow and challenging in crowded scenes. Counting by clustering (Brostow & Cipolla, 2006; Rabaud & Belongie, 2006) counts objects by grouping them based on their motion patterns, requiring high frame rates and reliable motion information. Counting by regression (Chan et al., 2008; Chan & Vasconcelos, 2009; Chen et al., 2012; Idrees et al. , 2013; Chan and Vasconcelos, 2011) counts objects by learning a direct mapping between image features and a count value, which is suitable for crowded environments and is computationally efficient, but it does not provide the spatial distribution of the objects of interest.

Lempitsky & Zisserman (2010) propose density maps as a way to incorporate the spatial appearance and constellation

Communicated by Ming-Hsuan Yang.

✉ Zenglin Shi
iezlshi@126.com

Pascal Mettes
P.S.M.Mettes@uva.nl

Cees G. M. Snoek
cgmsnoek@uva.nl

[1] Hefei University of Technology, Hefei, China

[2] University of Amsterdam, Amsterdam, The Netherlands

of object crowds into counting. Their density-based counting requires just a point annotation per countable object in each training image. These points are smoothed with Gaussian kernels to generate density maps. Counting then becomes a pixel-wise regression problem. Early density-based counting methods (Lempitsky & Zisserman, 2010; Pham et al., 2015) compute low-level features (e.g., HOG, SIFT) and learn regressors to predict the density maps. Afterwards, deep learning has become dominant in density-based counting, as it predicts density maps in an end-to-end manner. Deep convolutional networks are widely adopted, e.g., (Zhang et al., 2016; Onoro-Rubio & López-Sastre , 2016; Li et al., 2018; Cao et al., 2018; Liu et al., 2018b, c, 2019b; Jiang et al., 2019; Sindagi and Patel, 2019; Xu et al., 2022; Xiong et al., 2023) and several recent studies have leveraged the power of vision transformers to further enhance the accuracy of crowd counting models, e.g., (Lin et al., 2022; Sun et al., 2021; Gao et al., 2022; Yang et al., 2022; Tian et al., 2021; Liang et al., 2022). Advanced loss functions have been proposed to address the limitations of ℓ_1 and ℓ_2-norm loss functions in density-based counting, e.g., (Shi et al., 2018a; Shen et al., 2018a; Chan & Vasconcelos, 2009; Wang et al., 2020a; Wan et al., 2021). These advanced functions consider outliers, prevent image blurring, and account for local coherence. Some approaches have also leveraged auxiliary information to enhance density regression, e.g., (Sam et al., 2017; Kang et al., 2020; Sindagi & Patel , 2017; Ranjan et al., 2018; Shi et al., 2018; Liu et al., 2018c; Babu Sam et al., 2018; Shi et al., 2019a; Yang et al., 2020; Liu et al., 2021; Zhang and Chan, 2022). While these approaches are effective, they only rely on the point annotations to generate the density maps for training. The premise of this paper is that point annotations serve a purpose for counting beyond creating density maps.

In the conference version of our work (Shi et al., 2019b), we propose two ways to obtain *focus for free*, *i.e.*, free additional supervision signals from the same point annotations. On a local level, we create binary segmentation maps from point annotations and train a segmentation branch to focus only on the regions of interest. We also leverage the relative number of point annotations per image to train a branch with a global density loss to focus on the overall image density. Since then, several works have used point annotations in different ways to enhance counting. Ma et al. (2020) propose a scale-aware probabilistic model that leverages the geometric distribution of point annotations to address the scale variation problem. Similarly, Liang et al. (2022b) suggest a focal inverse distance transform map instead of a Gaussian density map to locate objects more accurately in dense regions with overlapping points. Liu et al. (2020) propose learning patch-wise density maps that are created with local count information from point annotations to minimize the discrepancy between training targets and evaluation metrics. In addition, Jiang et al. (2020a); Rong & Li (2021) use local count information from point annotations to learn attention masks that indicate different density levels of various image regions, reducing the local count error. Modolo et al. (2021); Qian et al. (2022); Jiang et al. (2020) also improve counting by learning an auxiliary segmentation task or density-level classification task by their network architecture design. Instead of a binary segmentation map for attention, Cheng et al. (2021) propose a probability map that shows the likelihood of each pixel being an object, and then learn to predict the probability map as a probabilistic intermediate representation for counting.

This paper introduces two new ways to repurpose point annotations for free in counting as illustrated in Fig. 1. First, in Sect. 3.1 we introduce a counting-specific augmentation that utilizes point annotations to simulate occluded objects in both input and density images to enhance the network's robustness to occlusions. Second, in Sect. 3.2 we propose foreground distillation, which generates foreground masks from the point annotations. With these masks we train an auxiliary network on images with blacked-out backgrounds, which learns to distill foreground knowledge without the interference of the background.

In Sect. 3.3 we show that our new proposals are naturally embedded in our previous local and global approaches (Shi et al., 2019b). Where the counting network in the conference version was based on a convolutional neural network, we integrate all focus-for-free methods in this paper into vision transformers with advanced loss functions, which we detail in Sect. 3.4.

In Sect. 4, our approach is evaluated on five counting datasets. Our results showcase complementary effects of the focus-for-free approaches. When these approaches combined, our models can accurately count objects of interest even in challenging scenarios, such as background clutter, occlusion, and varying crowd densities. Moreover, we achieve strong performance on ShanghaiTech Part_A and Part_B, UCF_QNRF, JHU-Crowd++, and NWPU-Crowd. Before detailing our approach, we first provide a broader discussion on related work.

2 Related Work

2.1 Counting with Density Maps

The most prevalent technique for object counting in images is to generate density maps through regression. This method was first introduced by Lempitsky & Zisserman (2010) and has since been the basis of most works. Over time, density-based counting has progressed significantly, thanks to improved network architectures and advanced loss functions.

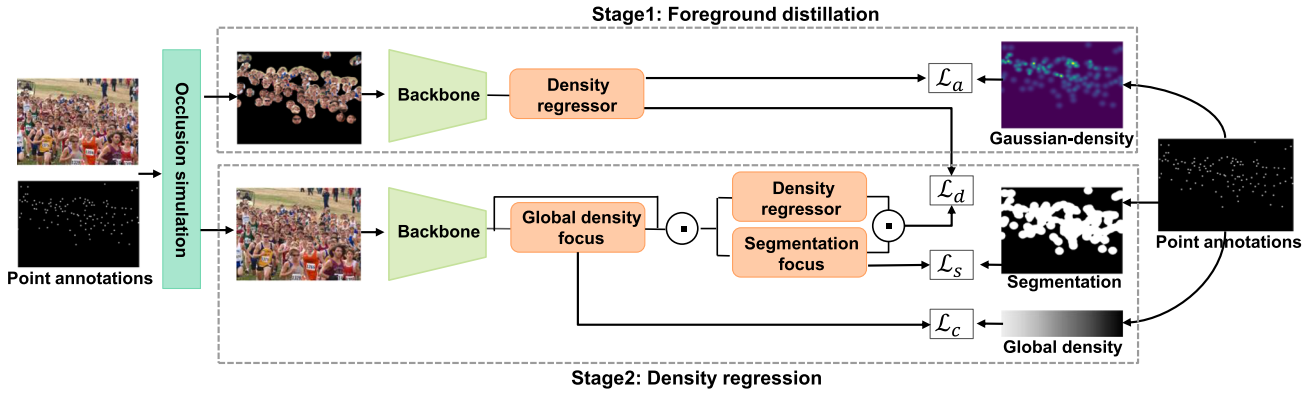


Fig. 1 Overview of our approach. We first repurpose the point annotations to simulate occlusion in images and density maps, improving robustness to crowded scenes (Sect. 3.1). Then we adopt a two-stage learning process, incorporating occlusion augmentation. In the first stage, indicated by the top dashed box, we generate foreground masks from the point annotations from which we train an auxiliary network on images with blacked-out backgrounds (Sect. 3.2). We employ a loss function $\mathcal{L}_a$ to supervise the training process, in which the Gaussian-density map is generated from the point annotations. In the second stage, depicted by the bottom dashed box, we transfer the knowledge acquired by the auxiliary network to the density estimation network via distillation. This transfer is guided by a loss function $\mathcal{L}_d$. To ensure that the density estimation network focuses on critical features, we utilize the repurposed point annotations to create a segmentation focus supervised with a segmentation loss $\mathcal{L}_s$ and a global density focus supervised with a global density classification loss (Sect. 3.3). Finally, we seamlessly integrate all these approaches into the vision transformer backbone, allowing for effective cooperation between the different components. Furthermore, our framework is flexible in accommodating different loss functions (Sect. 3.4)

Network Architectures Deep convolutional networks are widely adopted for counting by estimating density maps from images. Early works, e.g. (Zhang et al., 2016; Onoro-Rubio & López-Sastre , 2016; Sindagi & Patel , 2017), advocate a multi-column convolutional neural network to encourage different columns to respond to objects at different scales. Despite their success, these types of networks are hard to train due to structure redundancy (Li et al., 2018; Cheng et al., 2019b) and conflicts resulting from optimization among different columns (Shen et al., 2018b; Babu Sam et al., 2018). More recently, single-column deep networks have gained popularity due to their simpler architecture and improved training efficiency, *e.g.*, (Li et al., 2018; Cao et al., 2018; Liu et al., 2018c, b, 2019b; Jiang et al., 2019; Sindagi and Patel, 2019; Hu et al., 2020; Huang et al., 2020; Wang et al., 2022). For example, Li et al. (2018) combine a VGG network with dilated convolution layers to capture multi-scale contextual information. Sindagi and Patel (2019) propose a hierarchical attention-based network with attention mechanisms at various levels to enhance network features selectively. Hu et al. (2020) propose an encoder-decoder network with neural architecture search techniques.

Several recent studies have leveraged the power of vision transformers to enhance the accuracy of crowd counting models, e.g., (Lin et al., 2022; Sun et al., 2021; Gao et al., 2022; Yang et al., 2022; Tian et al., 2021; Liang et al., 2022; Shi et al., 2024). Lin et al. (2022) have incorporated global attention, learnable local attention, and instance attention into their counting model, by combining vanilla vision transformers and convolutional networks. Sun et al. (2021) introduces a global context learnable token to guide the counting. Gao et al. (2022) enhanced large-range contextual information by using a dilated Swin Transformer backbone and feature pyramid networks decoder. To model human top-down visual perception mechanisms, Yang et al. (2022) have proposed an overlap patching transformer block. Tian et al. (2021) adopt a pyramid transformer and a multi-scale regression head to achieve improved counting performance.

Loss Functions ℓ_1- and ℓ_2-norm loss functions are widely used to measure the per-pixel differences between estimated and ground-truth density maps for network training, but they suffer from problems such as sensitivity to outliers and image blur, pixel independent assumption neglecting the local coherence, and spatial correlation in density maps. Therefore, Shi et al. (2018a) propose a negative correlation loss to increase the robustness against outliers. Cao et al. (2018) add the local pattern consistency loss to reduce the sensitivity of ℓ_1- and ℓ_2-norm loss. Shen et al. (2018a) introduce an adversarial loss to make the blurring density maps sharp. Cheng et al. (2019a) propose a maximum excess over pixels loss to learn spatial awareness and be robust to noises. Ma et al. (2019) propose a Bayesian loss by calculating class conditional distributions for each annotated points rather than generating discrete density map as supervision. However, Bayesian loss cannot well handle false positives in the background, and requires a special design for the background region. Wang et al. (2020a) propose an advanced loss function by computing the optimal transport distance between predicted density maps and the ground-truth point maps. Wan et al. (2021) further propose an unbalanced optimal transport

loss function to preserve the count of the predicted densities and annotated points. Different from pixel-wise ℓ_1- and ℓ_2-norm loss, optimal transport loss produces penalties by considering all nearby pixels according to the distances rather than the pixel itself. Therefore, such losses better exploit the position information of the point annotations to provide high-quality supervision. However, optimal transport loss and its variants utilize the Sinkhorn algorithm (Peyré et al., 2019) to obtain the optimal transport matrix, which requires a number of iterations and are carried out in each training step, leading to inefficient training.

This work strives to improve counting by repurposing point annotations, which is compatible with any network architecture and loss function. We show that our approach can work well with both convolutional networks and vision transformers, supervised by an ℓ_1-norm loss or a distribution matching loss.

2.2 Learning Auxiliary Tasks for Counting

Several approaches have been proposed to enhance density regression by leveraging auxiliary information, e.g. (Sam et al., 2017; Sindagi & Patel , 2017; Ranjan et al., 2018; Shi et al., 2018; Liu et al., 2018c; Shen et al., 2018b; Babu Sam et al., 2018; Liu et al., 2018a; Yan et al., 2019; Zhao et al. , 2019; Shi et al., 2019a; Yang et al., 2020; Liu et al., 2021; Zhang et al., 2022), taking inspirations from other vision works such as (Liu et al., 2022; Shi et al., 2021b; Shi et al., 2021; Chen et al., 2021). Sam et al. (2017), for example, train a classifier to select the optimal regressor from multiple independent regressors for particular input patches. Ranjan et al. (2018) utilize one network to predict a high-resolution density map and another network to predict a low-resolution density map to help improve the accuracy of crowd counting. Zhao et al. (2019) propose a method involving three heterogeneous attributes, including geometric, semantic, and numeric attributes, as auxiliary tasks to assist in the crowd counting task. Zhao et al. (2019) learn an auxiliary perspective map prediction task to address the issue of scale variation in counting. Liu et al. (2021) developed a novel multi-experts training framework for crowd counting that exploits relations within samples.

In this paper, we also investigate counting from a multi-task perspective, but from a different point of view. We posit that the point annotations serve more purposes than just constructing density maps, and we introduce occlusion simulation, foreground distillation, and focus-for-free (Shi et al., 2019b). Occlusion augmentation leverages point annotations to simulate occluded objects in both the input and density images, enabling the network to become more robust to occlusions. Foreground distillation generates foreground masks from point annotations, and uses the masks to distill foreground counting knowledge, which effectively reduces the impact of background pixels on counting accuracy. Focus-for-free outlines supervised focus from segmentation and global density classification to repurpose the point annotations for free.

3 Focus for Free: Repurpose Points to Count

We formulate the counting task as a density map regression problem. Density-based counting relies on point annotations indicating the locations of objects in training images. These annotations are a set of coordinates $\mathcal{P}$, where each point $P=(x, y) \in \mathcal{P}$ corresponds to an object's location. From the point annotations a Gaussian-density map is created through convolutions with a normalized Gaussian kernel:

$$D(p) = \sum_{i=1}^{N} \mathcal{N}\left(p|\mu = P, \sigma_P^2\right), \tag{1}$$

where p denotes a pixel location, and $\mathcal{N}(\cdot)$ is a normalized Gaussian kernel with mean P and an isotropic variance σ_P. Determining σ_P for each point P is difficult because of object-size variations caused by perspective distortions. A solution is to estimate the radius r of an object as a function of the K nearest neighbor annotations, as done in the Geometry-Adaptive estimation of Zhang et al. (2016).

Given a density map, we optimize a neural network f to learn how to map input images I_i to their output density maps D_i, forming a dataset $(I_i, D_i)_{i=1}^N$. During inference, we obtain the global object count T_i of an input image I_i by summing all the pixel values within the predicted density map $D_i'=f(I_i)$, as follows: $T_i=\sum p \in I_i D_i'(p)$. This operation integrates the predicted density values across the image domain and provides an estimate of the number of objects present in the image.

We will now present three methods that enhance counting by repurposing existing point annotations beyond creating Gaussian-density maps. We begin by introducing a counting-specific augmentation that leverages point annotations to simulate occluded objects in both input and density images. This augmentation improves the network's ability to handle occlusions, as discussed in Sect. 3.1. Then, we propose foreground distillation in Sect. 3.2. This method involves generating foreground masks based on point annotations and utilizing these masks to distill foreground knowledge. By incorporating this approach, we reduce the impact of background pixels on counting accuracy. To ensure that the network focuses on critical features during density regression, we introduce both local and global focus with explicit supervision derived from point annotations in Sect. 3.3. Finally, we integrate all the aforementioned methods into vision transformers with advanced loss functions, which

are detailed in Sect. 3.4. A comprehensive overview of our method is provided in Fig. 1.

3.1 Counting-Specific Occlusion Simulation

In counting, we often deal with congested scenes and many objects, which leads to occlusions. To address this challenge, we reuse annotation points to augment overlapping objects. Occlusions often occur when objects overlap, making it difficult to count them accurately. One approach to handle different levels of occlusion is to train a network with a sufficient number of samples for each level. However, the distribution of occlusion levels in existing benchmarks is long-tailed, meaning there are not enough images to learn from natural occlusions (Zhang et al., 2016; Idrees et al., 2018; Sindagi et al., 2020; Guerrero-Gómez-Olmedo et al., 2015). To address this issue, we suggest simulating various occlusion levels by using annotation points. This will enable us to train the network on a more diverse set of occlusion levels, improving its ability to identify and count overlapping objects, even in congested scenes.

Occlusion Simulation Given a training sample consisting of an input image and its point annotations, $(I, \{P_i\}_{i=1}^{N})$, we represent any object $\mathcal{O}_i$ in the image I as $(x_i, y_i, 2\sigma_i)$. Here, (x_i, y_i) represents the coordinate of the corresponding annotated point P_i, and σ_i is the variance of the corresponding Gaussian kernel. This representation assumes that the object $\mathcal{O}_i$ is circular and its radius is approximately equal to $2\sigma_i$, as proposed in Lempitsky & Zisserman (2010). To generate occlusion scenarios, we randomly select an object, $\mathcal{O}_{occ}$, to be occluded. We then select one of its neighboring objects, denoted as $\mathcal{O}_{copy}$, which will be copied and pasted to occlude $\mathcal{O}_{occ}$. The pasted position of $\mathcal{O}_{copy}$ determines how $\mathcal{O}_{occ}$ will be occluded, defined as:

$$\begin{aligned} x_{paste} &= \lfloor r \cdot cos(\theta) \rfloor + x_{occ}, \\ y_{paste} &= \lfloor r \cdot sin(\theta) \rfloor + y_{occ}, \end{aligned} \tag{2}$$

where $r{=}r_{copy} + r_{occ}\epsilon_r$ and $\theta{=}2\pi\epsilon_\theta$. ϵ_r and ϵ_θ are randomly sampled from $\mathcal{U}(0, 1)$. $\lfloor \cdot \rfloor$ is the floor operator. r and θ decide how much and where $\mathcal{O}_{occ}$ will be occluded.

Blending The essence of training the density-based counting network lies in establishing a mapping between objects and their corresponding densities, regardless of the objects' spatial positions and orientations. Consequently, when simulating occlusions, our approach involves augmenting both the objects and the density map to maintain this crucial alignment. Notably, we have observed that the artifacts at the boundaries, which arise when pasting objects onto an image, can adversely affect the network's training. To address this issue, we employ a blending technique to smooth out these boundary artifacts. Specifically, we first copy and paste the object $\mathcal{O}_{copy}$ to the position (x_{paste}, y_{paste}) of the image I, generating a new image I'. Then, we compute the binary mask α of the pasted object $\mathcal{O}_{paste}$, which we represent by $(x_{paste}, y_{paste}, r_{paste})$ where $r_{paste}{=}r_{copy}$. To smooth out the edges of the pasted object, we apply a Gaussian filter to the binary mask α and obtain a smoothed mask $\tilde{\alpha}$. From $\tilde{\alpha}$ we construct the occluded image I_{occ} and its corresponding ground-truth density map:

$$\begin{aligned} I_{occ} &= (1 - \tilde{\alpha}) \odot I + \tilde{\alpha} \odot I', \\ D_{occ} &= D + G(\mathcal{O}_{paste}), \end{aligned} \tag{3}$$

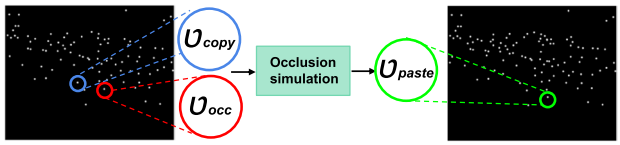
where D denotes the ground-truth density map of the image I, $G(\mathcal{O}_{paste})$ denotes the density map of the pasted object $\mathcal{O}_{paste}$, and G denotes the Gaussian kernel. Equation 3 states that the pixels to be occluded in the original image are replaced by the pixels of the pasted object to obtain the occluded image. The corresponding occluded density map is obtained by simply adding the density map of the pasted object to the original density map.

Occluding Adaptively For a training image, we need to decide how many of its objects will be occluded with our approach. A simple way is to set a fixed ratio. However, the training image may already have naturally occluded objects. Thus, we should create new occlusion for the training image according to its current occlusion level. Intuitively, the lower the occlusion level, the higher the amount of occlusions that should be simulated. To compute the occlusion level, we create an occlusion map M for the training image by,

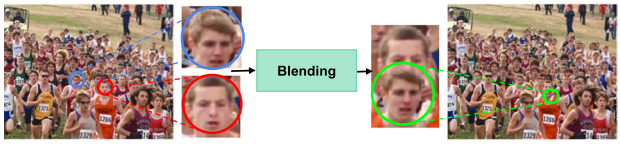
$$\begin{aligned} M(x, y) &= \sum_{i=1}^{N} S(x, y; \mathcal{O}_i), \text{ where} \\ S(x, y; \mathcal{O}_i) &= \mathbb{1}(||x - x_i||^2 + ||y - y_i||^2 \leq (r_i)^2). \end{aligned} \tag{4}$$

Here (x, y) denotes a pixel location, $\mathcal{O}_i{=}(x_i, y_i, r_i)$ denotes an object, and $\mathbb{1}(\cdot)$ is a binary indicator function, which states that a pixel obtains a value of one if it is within an object region. Hence, each pixel value in the occlusion map M indicates the amount of object regions it belongs to. The occlusion level $\bar{M}$ is computed as the average value of non-zero pixels in the occlusion map M. Then the percentage of the objects to be occluded can be adaptively obtained by $\beta/\bar{M}$ where β is the upper boundary, and $\beta{=}0.3$ empirically performs well.

Our approach explicitly simulates occlusions tailored for counting tasks on both the input images and density maps. It does this by modeling how and where one object is occluded by another one, and adaptively deciding how many of the objects will be occluded. In contrast, standard copy-paste data augmentations, e.g., Yun et al. (2019); Ghiasi et al. (2021), are focused on generating new training samples through random object copying and pasting but do not explicitly produce samples with occlusions. As a result, our



(a) Occlusion simulation for point annotations



(b) Occlusion simulation for image

Fig. 2 Occlusion simulation. **a** We randomly choose two object points, one for the object which will be occluded, denoted by $\mathcal{O}_{occ}$ and another for the object which will be copied and pasted, denoted by $\mathcal{O}_{copy}$. The pasted position of $\mathcal{O}_{copy}$ is computed based on Eq. (2), as indicated by $\mathcal{O}_{paste}$. **b** We generate an occluded image with $\mathcal{O}_{occ}$, $\mathcal{O}_{copy}$, and $\mathcal{O}_{paste}$. To avoid boundary artifacts, we perform blending to smooth out the boundary artifacts

approach is notably more effective in accurately counting occluded objects, as demonstrated in Table 1

3.2 Foreground Distillation

For the second way to repurpose point annotations, we introduce foreground distillation to obtain better density maps without the impact of background. Typically, a network for density map prediction is learned by optimizing a loss function of the form $\mathcal{L}_d{=}\ell_p(f_d(I) - D)$ where f_d the network, I the input image, D the Gaussian-density map, and ℓ_p the Lp norm loss function. We reformulate the loss function by dividing the density map into two components, one representing the background of the input image and the other corresponding to the foreground containing the objects of interest: $\mathcal{L}_d{=}\ell_p(f_d(I)_{bg}){+}\ell_p(f_d(I)_{fg} - D_{fg})$, where D_{bg} is naturally omitted as every single element in it is zero. Based on this reformulation, we identify two issues with current approaches. First, when $\ell_p(f_d(I)_{bg})$ is not optimized perfectly, they are prone to misidentifying the background as objects (densities). Second, while trying to transform objects into densities, they cannot fully utilize their learning capacity as they predict densities by $f_d(I)_{fg}$ not only on objects but also on the background.

Auxiliary Network. A common approach to mitigate the impact of background pixels is to train a separate segmentation network that differentiate between foreground and background regions. However, achieving a perfect foreground segmentation can be difficult, which may compromise the effectiveness of this approach. To remedy this issue, we propose to further distill foreground counting knowledge. Specifically, we introduce an auxiliary network that is trained on images where the background is blacked-out. This enables the network to learn the intrinsic features of the foreground and reduces the influence of background pixels. The training loss is given as:

$$\mathcal{L}_a{=}\ell_p\big(f_a(I \odot S) - D\big), \tag{5}$$

where f_a is the auxiliary network that has the same architecture as the density prediction network f_d. D is the Gaussian-density map, $I \odot S$ is the image with background areas blacked-out, and S is the foreground mask, which is derived as a function of the point annotations and their estimated variance. The binary value for each pixel location p is determined as:

$$S(p) = \begin{cases} 1 & \text{if } \exists_{P\in\mathcal{P}}\big(||p - P||^2 \leq \sigma_P^2\big), \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Eq. 6 states that a pixel p obtains a value of one if at least one point P is within its variance range σ_P as specified by a kernel estimator.

Distillation Loss The optimization of the auxiliary network will be stopped when it achieves maximum accuracy on the validation set. Then we distill the knowledge of the trained auxiliary network to the density prediction network with the loss:

$$\mathcal{L}_d{=}\ell_p\big(D' - f_a(I \odot S)\big), \tag{7}$$

where D' represents the predicted density map. Note that we normalize the output density from the auxiliary network to align its total count with that of the Gaussian density map D. Unlike traditional distillation methods that compress information (Hinton et al., 2015), our distillation technique aims to extract foreground counting knowledge. We employ an identical architecture for both the auxiliary network and the primary density prediction network, enabling us to utilize the former as a pre-trained network for the latter. This reduces the training time of the density prediction network significantly. In the density prediction loss function $\mathcal{L}_d$, we replace the Gaussian density map with our distilled density map for the final training, rather than do both. This flexible approach can be employed either independently or in conjunction with traditional segmentation-based methods to mitigate the impact of the background.

3.3 Local and Global Focus

Here, we outline the original global and local focus for free, which are complementary to the occlusion and distillation approaches.

Local Segmentation Focus Intuitively, pixels within a specific range of any point annotation should be of high focus, while pixels in undesired regions should be mostly disregarded. However, in the standard optimization setup where

Table 1 Effect of occlusion simulation

	T2T-Vit						CSRNet+					
	Part_A			UCF_QNRF			Part_A			UCF_QNRF		
	Low	High	Overall	Low	High	Overall	Low	High	Overall	Low	High	Overall
Base network	36.6	94.7	60.1	17.8	103.5	90.8	39.2	95.9	61.9	18.7	106.5	93.6
w/ Cutout (DeVries & Taylor, 2017)	35.9	93.6	58.9	**15.7**	101.9	88.6	38.6	95.0	61.2	**16.5**	104.7	91.8
w/ CutMix (Yun et al., 2019)	35.7	93.2	58.6	16.4	99.6	87.9	38.0	94.3	60.6	17.8	103.1	90.6
w/ ours	**34.9**	**91.7**	**57.2**	16.8	**98.3**	**87.1**	**37.8**	**92.2**	**59.1**	18.3	**101.3**	**88.9**

Compared to a base network and our reimplementation of two related works, our occlusion augmentation reduces the count error (MAE) for the entire test set, the *low* occlusion set, and especially for the *high* occlusion set
Bold numbers highlight the best counting results

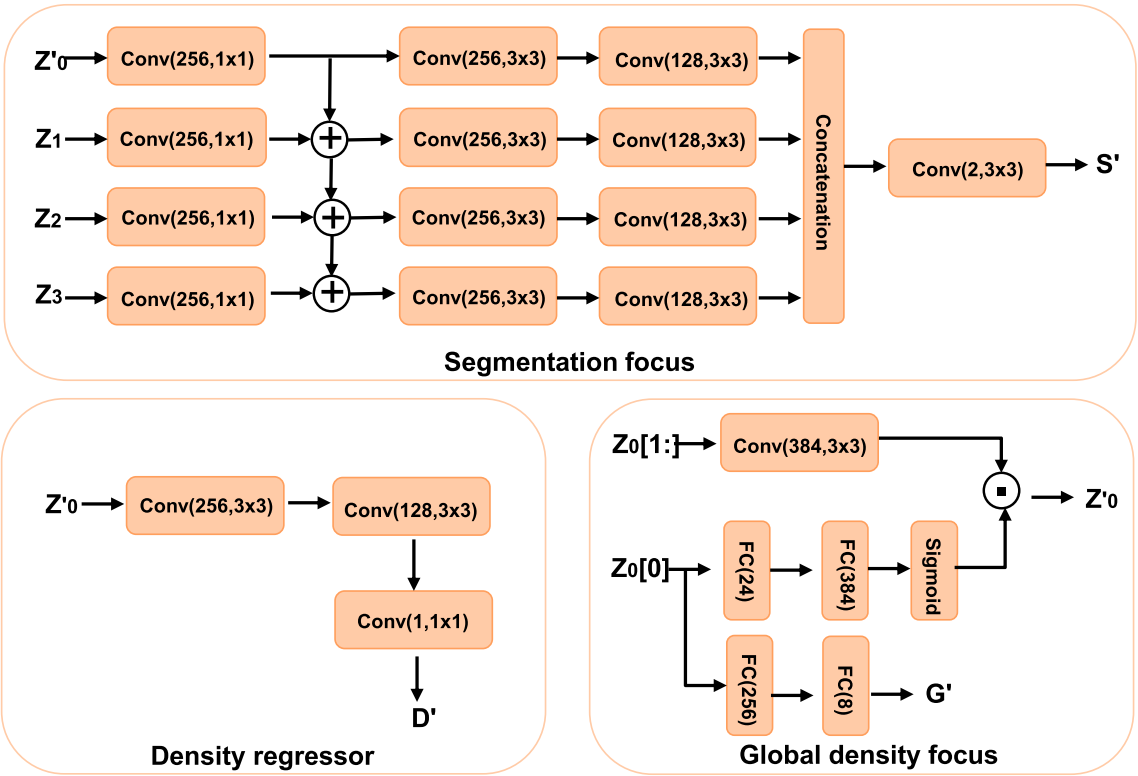


Fig. 3 The network architectures of density regressor, global density focus, and segmentation focus. The convolution layer is denoted as Conv(N, K×K), where N represents the number of channels and K×K represents the kernel size. The fully connected layer is represented as FC(M), where M indicates the number of nodes. Following each convolution layer and fully connected layer, there is a ReLU activation layer applied. Element-wise addition and multiplication are denoted by the symbols $\oplus$ and $\odot$, respectively. Z_0, Z_1, Z_2, Z_3 are the outputs from different layers in the transformer encoder. D', S', and G' denotes the predicted density map, segmentation map, and global density, respectively

the density map is the sole factor determining the loss, each pixel is treated equally. As a result, irrelevant pixels dominate the loss since only a fraction are near-point annotations. To overcome this limitation, we reuse the point annotations to create a binary segmentation map and exploit this map to provide focused supervision through a standalone loss per-pixel weighted focal loss:

$$\mathcal{L}_s = \sum_{l\in\{0,1\}} -\alpha^l S^l (1-S')^{\gamma_s} log(S'), \tag{8}$$

where S is the segmentation ground-truth created according to Eq. (6). $\alpha^l = 1 - \frac{|S^l|}{|S|}$ and $S'=f_s(I)$ represents the segmentation map predicted by a branch network f_s. The focal parameter γ_s is set to 2 throughout this network, as recommended by Lin et al. (2017).

We utilize the predicted segmentation map $f_s(I)$ to highlight the desired foreground areas while suppressing false background areas in the output density map. This is done by incorporating the mask into the density prediction loss func-

tion. Specifically, through the following masked loss function

$$\mathcal{L}_d = \ell_p\big(f_d(I) \odot f_s(I) - D\big), \tag{9}$$

where $\odot$ denotes element-wise multiplication. This approach reduces background errors and improves density prediction.

Global Density Focus Aside from providing a local focus through foreground segmentation, point annotations can also be repurposed to achieve a global focus by examining contextual information through global density classification. In dense prediction tasks, it is crucial to comprehend and leverage contextual information. In the context of per-pixel density prediction for counting, global context is associated with the overall crowd density in the image. To capture such global context, we learn a global density classification task. We then utilize this captured global context to develop a global focus that highlights feature maps that align with the global density reflected by the annotation points, leading to improved counting performance.

To perform global density classification, we first obtain a global density ground-truth from the point annotation. Specifically, the global density for a patch j in the training image i is given as:

$$G_{j,i} = \frac{|\mathcal{P}_{j,i}|}{L}, L = \left\lfloor \max_{i=1,..,N} \left(\frac{|\mathcal{P}_i|}{Z_i} \cdot Z_{j,i} \right) / M \right\rfloor + 1. \tag{10}$$

Here $|\mathcal{P}_{j,i}|$ denotes the number of object points in patch j, L denotes the global density step size, Z_i and $Z_{j,i}$ denote the number of pixels in image i and patch j respectively. The step size calculation involves determining the highest global density across image patches, while the parameter M denotes the number of global density levels used in the process. By leveraging the global density ground-truth, we train a network f_c for classifying global density with the following loss:

$$\mathcal{L}_c = \sum_{l \in \{0,1,..,M\}} -G^l(1-G')^{\gamma_c} log(G'), \tag{11}$$

where $G' = f_c(I)$ and γ_c is set to 2. The f_c model comprises a backbone network and a classifier. The backbone network is shared with the per-pixel density prediction network f_d. After capturing the global context, we develop a global focus that emphasizes feature maps aligned with the global density. To achieve this, we employ a new branch network f_g to generate a global density focus output. The resulting focus output is then utilized by the density prediction network's backbone to highlight feature maps that exhibit similar global contextual patterns to the ground-truth density maps.

3.4 Network and Optimization

Counting Transformer Vision transformer architectures have shown significant promise in counting and are quickly becoming the preferred backbone. This has motivated us to integrate our methods into transformers, capitalizing on their capabilities to advance counting models.

For the transformer encoder, we adopt the progressive tokenization module introduced by Yuan et al. (2021) instead of relying on the basic tokenization approach employed in Vit (Dosovitskiy et al., 2020). This tokenization module enables us to aggregate adjacent tokens into a single token, thereby incorporating local structural information from surrounding tokens and reducing the token length in a step-by-step manner. To be more specific, we utilize a Token-to-Token (T2T) mechanism, where tokens produced by a transformer layer are first reconstructed as an image, then split into overlapping patches, and finally aggregated by flattening the patches. This process allows the local structure of surrounding patches to be embedded into the tokens, which are subsequently fed into the next transformer layer. By performing T2T iteratively, the local structure is integrated into the tokens, and the token length is reduced via the aggregation process. We use an efficient deep-narrow backbone structure to learn the representations from these tokens. A decoder, consisting of three convolution layers, is employed as a density regressor to estimate the density map.

The decoder for segmentation focus is designed by adopting a multi-level feature aggregation approach. We begin by selecting four layers from the encoder, denoted by $\{z_0, z_1, z_2, z_3\}$ where z_0 represents the last layer of the encoder. To emphasize specific selected layers, we deploy four independent streams, each dedicated to processing one of the chosen layers. Within each stream, we utilize a 3-layer convolutional network to process the corresponding features. To encourage interactions between different streams, we introduce a top-down aggregation design through element-wise addition after the first layer. Subsequently, after the third layer, we obtain the fused feature by concatenating the outputs of all streams along the channel dimension. This fused feature is then fed into a convolution layer to produce the segmentation map.

In the case of the global density focus, our decoder comprises three branches. We divide the patch tokens obtained from z_0 into a global density class token $z_0[0]$ and the remaining patch tokens $z_0[1:]$. The first branch processes the patch tokens $z_0[1:]$ using a convolution layer for re-weighting. The second branch generates a global density focus map using two fully-connected layers followed by a Sigmoid activation layer, with the global density class token $z_0[0]$ serving as input. Meanwhile, the third branch leverages the global density class token $z_0[0]$ to predict the global density by employing two fully-connected layers, allowing $z_0[0]$ to cap-

ture global context information. The generated focus map by the second branch is then utilized to refine the local feature map from the first branch. This refinement process enables the network to emphasize critical feature channels, leading to enhanced features. Figure 3 provides the details of the three decoders.

Composite Loss The training of the final counting network is carried out in two distinct stages. Initially, an auxiliary network is trained to perform foreground distillation by minimizing the loss function $\mathcal{L}_a$ (as defined in Eq. 5). In the second stage, we incorporate focus for free into the counting network by training it using a composite loss function. This loss function comprises three distinct components: the distillation loss $\mathcal{L}_d$ (defined in Eq. 7), the segmentation loss $\mathcal{L}_s$ (defined in Eq. 8), and the global density classification loss $\mathcal{L}_c$ (defined in Eq. 11), *i.e.*,

$$\mathcal{L} = \mathcal{L}_d + \lambda_s \mathcal{L}_s + \lambda_c \mathcal{L}_c, \tag{12}$$

where (λ_s, λ_c) denotes the weighting parameters of the different loss functions. Occlusion augmentation is used in both stages. We can also use a distribution matching loss (Wang et al., 2020a) for $\mathcal{L}_d$ to obtain an improved result, where $\mathcal{L}_d$ is defined as:

$$\mathcal{L}_d = \ell_C(D', D) + \lambda_{ot}\ell_{OT}(D', D) + \lambda_{tv}\ell_{TV}(D', \hat{D}). \tag{13}$$

Here, ℓ_C is used to minimize the difference of the total count between the predicted density map D' and the annotated point density map D. ℓ_{OT} is used to minimize the divergence between D' and D by regarding the density map as a probability map. ℓ_{TV} is used to help ℓ_{OT} to handle the low-density areas of the crowd. Since the low-density areas are often impacted by the background, we use our distilled density map $\hat{D}{=}f_a(I \odot S)$ instead of D to distill foreground counting knowledge. $(\lambda_{ot}, \lambda_{tv})$ are the weights for the corresponding loss term. Throughout this work, these parameters are set to (0.1, 0.01), following Wang et al. (2020a).

4 Experiments and Results

We first describe the experimental setup. We then analyze the effect of repurposing point annotations proposed in this work, comparing each to their respective baseline. Finally, we report the results of our final approach and compare them to the state-of-the-art.

4.1 Experimental setup

Datsets We consider five counting datasets in this paper, as commonly used in the recent literature, *e.g.*, (Bai et al., 2020; Wan et al., 2021; Liu et al., 2021; Wang et al., 2021; Song

Table 2 Effect of foreground distillation

	T2T-Vit						CSRNet+					
	Part_A			UCF_QNRF			Part_A			UCF_QNRF		
	BG	FG	Overall	BG	FG	Overall	BG	FG	Overall	BG	FG	Overall
Base network	3.2	59.1	60.1	8.2	89.7	90.8	3.4	61.5	61.9	8.7	91.3	93.6
w/ standard distillation (Hinton et al., 2015)	3.1	58.8	59.8	8.1	89.1	90.2	3.3	61.1	61.4	8.6	91.0	93.1
w/ attention-injective (Liu et al., 2019a)	3.0	58.2	59.1	5.8	88.5	89.4	3.3	59.7	60.5	6.1	89.9	91.2
w/ segmentation attention (Modolo et al., 2021)	**2.8**	57.7	58.2	4.4	87.6	88.8	**2.9**	58.8	59.3	4.5	88.3	89.4
w/ background feature filtering (Mo et al., 2020)	2.9	57.1	57.7	4.8	86.7	87.9	3.1	58.2	58.8	4.3	87.1	88.3
w/ decoupled learning (Cheng et al., 2021)	3.0	56.3	57.1	**4.2**	**85.8**	**86.4**	3.1	57.8	58.2	3.5	**86.2**	**87.3**
w/ our foreground distillation	2.9	**55.7**	**56.5**	4.8	86.3	86.7	3.0	**57.1**	**57.6**	**3.4**	86.8	87.8

Compared to base networks and our reimplementation of five related works, our simple foreground distillation better reduces the overall count error (MAE), for both background (BG) and foreground (FG)
Bold numbers highlight the best counting results

Fig. 4 Occlusion Creation. Illustrative examples for creating new occlusions on images and corresponding density maps. Our occlusion-augmented image and density look quite natural, as masked in the green regions (Color figure online)

et al., 2021; Ma et al., 2021): ***ShanghaiTech*** (Zhang et al., 2016) consists of 1198 images with 330,165 pedestrians. This dataset is divided into two parts: ***Part_A*** with 482 images in which crowds are mostly dense, and ***Part_B*** with 716 images, where crowds are sparser. Each part is divided into a training and testing subset as specified in (Zhang et al., 2016). ***UCF-QNRF*** (Idrees et al., 2018) consists of 1535 images, with the count ranging from 49 to 12,865. For training 1201 images are available, and the remaining 334 form the test set. ***JHU-CROWD++*** (Sindagi et al., 2020) consists of 4372 images with a total of 1.51 million point annotations. The dataset is split into a training set of 2272 images, a validation set of 500 images and a testing set of 1600 images. ***NWPU-Crowd*** (Wang et al., 2020) consists of 5109 images with over 2 million point annotations. The dataset is split into a training set of 3109 images, a validation set of 500 images and a testing set of 1500 images.

For all datasets, we augment the images by randomly performing horizontal flipping, and randomly cropping 256×256 patches for ShanghaiTech Part_A, 384×384 patches for JHU-CROWD++ and NWPU-Crowd, and 512×512 patches for ShanghaiTech Part_B and UCF-QNRF. For Gaussian density map generation, we follow the previously suggested dataset settings from (Zhang et al., 2016; Idrees et al., 2018; Sindagi et al., 2020).

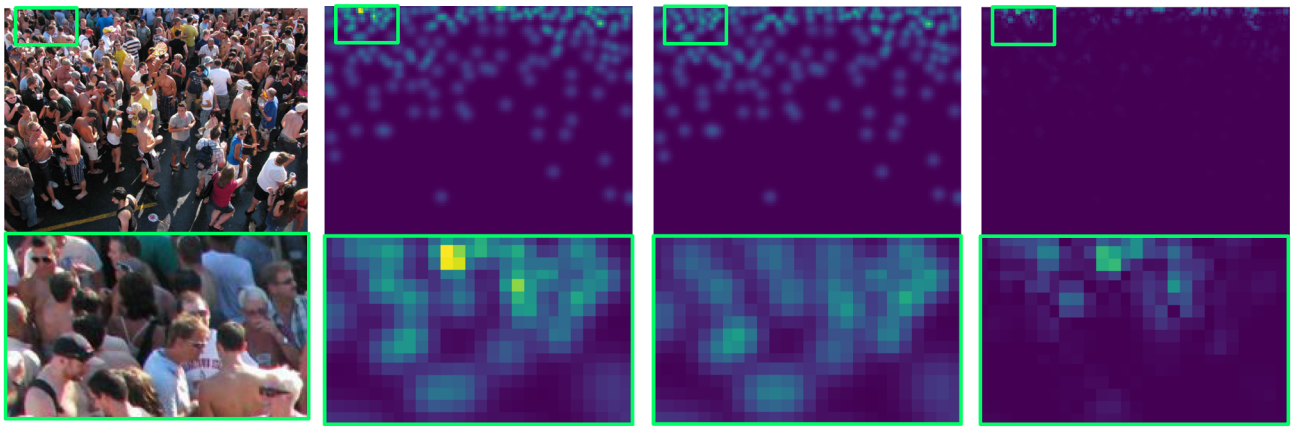
Implementation Details We adopted T2T-Vit-14 (Yuan et al., 2021) as our transformer backbone network and integrated our methods using the guidelines outlined in Sect. 3.4. Additionally, we incorporated CSRNet (Li et al., 2018) as our convolutional backbone network. CSRNet has been widely used in the literature, as demonstrated by its inclusion in multiple studies (Zhao et al. , 2019; Ma et al., 2019; Wan & Chan, 2020; Wang et al., 2020a; Wan et al., 2021; Ma et al., 2021). To enhance its performance, we introduced batch normalization layers into our implementation of CSRNet, similar to the approach taken by Bai et al. (2020). This improved version is called CSRNet+. For our integration with CSRNet+, we used the same backbone network as the frontend network and the same segmentation focus and density regressor as the back-end network. The global density focus is the same as that used with T2T-Vit-14, but we employed a bilinear pooling layer instead of using the global density class token. Both networks have been initialized with their ImageNet pre-trained models for faster training. Adam with batches of 10 is used for optimization. By default, we have used a ℓ_1 loss function. The learning rate has been fixed at 1e-4 for CSRNet+ and le-5 for T2T-Vit-14.

Metrics We report the standardized Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) metrics given count estimates and their ground-truth.

4.2 Counting-Specific Occlusion Simulation

We first demonstrate the potential of our occlusion handling augmentation. To highlight its effectiveness in highly occluded images, we divide the test images into two sets according to their occlusion level, as computed by Eq. (4). The test images are grouped into the *low* occlusion set if their occlusion level is lower than 1.5, the remaining images are grouped into the *high* occlusion set. For ShanghaiTech Part_A, we obtain 109 low and 73 high occlusion images and for UCF_QNRF, we obtain 49 low and 285 high occlusion images.

To highlight that the key here is occlusions, not just the augmentation, we have additionally investigated the effect of adding well-known occlusion augmentation approaches such as Cutout (DeVries & Taylor, 2017) and CutMix (Yun et al., 2019). Cutout randomly zero-outs a region in the training



(a) Training image (b) Gaussian density (c) Distilled density (d) Difference

Fig. 5 Density maps from distillation. Illustrative examples of Gaussian and distilled density maps. Compared to Gaussian densities, distilled densities better reflect the true foreground knowledge, reducing local noisy maxima (yellow region) and matching the objects in the image, especially occluded objects (Color figure online)

Table 3 Effect of local and global focus for free

	T2T-Vit		CSRNet+	
	Part_A	QNRF	Part_A	QNRF
Base network	60.1	90.8	61.9	93.6
w/ Chen et al. (2017)	59.2	89.7	61.2	91.8
w/ local segmentation focus	58.4	88.3	59.4	90.6
w/ SE Woo et al. (2018)	59.7	89.3	61.2	91.8
w/CBAM Hu et al. (2018)	59.4	88.9	60.6	90.5
w/ global density focus	58.7	87.6	59.1	88.6
w/ combined focus for free	56.9	85.8	57.6	86.4

Our focus for free consisits of focus from segmentation and focus from global density, which incorporates explicit supervisions for spatial attention and channel attention, outperforms both the base networks and three related works that implement spatial attention and channel attention implicitly. When combined, our focus for free further reduces MAE significantly across datasets and architectures

images. Instead of simply removing pixels, CutMix replaces the removed regions with a patch from another image. Cutout and CutMix create new training samples, but they do not create new occluded samples. Differently, our approach creates a variety of occluded samples by explicitly simulating the occlusion in the real world.

The results are shown in Table 1. Compared to the T2T-Vit base counting network, both Cutout and CutMix reduce the counting error, but cannot compete with our occlusion simulation (Part_A: 58.9 for Cutout, 58.6 for CutMix, and 57.2 for ours - UCF_QNRF: 88.6 for Cutout, 87.9 for CutMix, and 87.1 for ours). Especially, our approach works much better when occlusion levels are high (Part_A: 93.6 for Cutout, 93.2 for CutMix, and 91.7 for ours - UCF_QNRF: 101.9 for Cutout, 99.6 for CutMix, and 98.3 for ours). This experiment solidifies our point: occlusions weight heavily on the counting error, and there are not enough natural occlusions to learn from. Instead, by simulating occlusions, we can easily reduce the MAE effectively.

4.3 Foreground Distillation

In this experiment, we evaluate the effectiveness of foreground distillation. We compare it to the standard distillation (Hinton et al., 2015) on top of the same counting network, where we use the same network architecture for both student and teacher for a fair comparison. We also compare it to the other four baselines that were proposed for reducing background errors as well. The first baseline (Liu et al., 2019a) is a cascade counting approach, which cascades a crowd-level attention prediction network and a density estimation network. The second baseline (Modolo et al., 2021) reduces the background error by learning a segmentation attention map for the output density map. The third one (Mo et al., 2020) filters the background noise on the intermediate features by learning a head segmentation mask. The fourth baseline (Cheng et al., 2021) learns to count with decoupled learning, which first regresses a probability map to identify possible object regions, and then predicts a density map on

Table 4 Effect of combining our three methods in terms of MAE on ShanghaiTech Part_A and UCF_QNRF

Three methods of repurposing point annotations			T2T-Vit		CSRNet+	
Occlusion simulation	Foreground distillation	Focus for free	Part_A	UCF_QNRF	Part_A	UCF_QNRF
			60.1	90.8	61.9	93.6
✓			57.2	87.1	59.1	88.9
✓	✓		55.8	85.5	56.2	86.2
✓		✓	56.3	84.9	56.8	85.6
	✓	✓	52.2	82.6	53.5	83.7
✓	✓	✓	**51.6**	**80.9**	**52.7**	**81.8**

With each new addition, the counting error decreases, showing their complementary nature
Bold numbers highlight the best counting results

Table 5 Comparison to the state-of-the-art on ShanghaiTech Part_A, Part_B, UCF_QNRF, JHU-CROWD++ and NWPU-Crowd

	Backbone	Part_A		Part_B		UCF_QNRF		JHU-CROWD++		NWPU-Crowd	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
ℓ_1/ℓ_2-norm loss											
Xu et al. (2022)	VGG-16	65.8	112.1	8.6	13.9	104.4	174.2	76.4	292.7	94.1	388.2
Wan et al. (2020)	VGG-16	63.8	99.2	7.8	12.7	99.5	173.0	69.7	268.3	100.5	415.5
Ma et al. (2021)	VGG-16	58.4	97.9	–	–	96.3	155.7	65.1	269.3	–	–
Shu et al. (2022)	VGG19	57.5	94.3	6.9	11.0	80.3	137.6	57.0	235.7	**76.8**	**343.0**
Cheng et al. (2021)	VGG-16	57.2	93.0	6.3	10.7	81.7	137.9	73.7	292.5	85.5	361.5
Bai et al. (2020)	VGG-16	55.4	97.7	6.4	11.3	**71.3**	132.5	–	–	–	–
Tran et al. (2022)	ViT-B/32	54.8	**80.9**	8.6	13.8	87.0	141.9	–	–	–	–
Cheng et al. (2022)	ResNet-50	54.8	89.1	**6.2**	**9.9**	81.6	153.7	58.2	**245.1**	–	–
Wang et al. (2021)	VGG-16	54.6	91.2	6.4	10.9	81.1	131.7	–	–	–	–
Ours (CSRNet+)	VGG-16	52.7	88.6	6.9	11.2	81.8	133.5	57.9	258.4	78.5	347.8
Ours (T2T-Vit)	T2T-Vit-14	**51.6**	84.2	6.5	10.3	80.9	**129.5**	**56.8**	254.6	77.4	345.9
DM loss											
Wan et al. (2021)	VGG19	61.3	95.4	7.3	11.7	84.3	147.5	59.9	259.5	79.3	346.1
Wang et al. (2020a)	VGG-19	59.7	95.7	7.4	11.8	85.6	148.3	–	–	88.4	388.6
Liu et al. (2021)	ResNet-18	55.4	91.3	6.9	10.3	**76.2**	**121.5**	59.9	259.5	**74.7**	**267.9**
Ma et al. (2021)	VGG-19	55.0	92.7	–	–	80.7	146.3	59.3	248.9	–	–
Sun et al. (2021)	T2T-Vit-14	53.1	**82.2**	7.3	11.5	83.8	143.4	54.8	**208.5**	82.0	366.9
Song et al. (2021)	VGG-16	52.7	85.1	6.3	**9.9**	85.3	154.5	–	–	77.4	362.0
Ours (CSRNet+)	VGG16	51.2	85.7	6.7	10.9	80.2	131.7	56.4	252.4	77.9	349.8
Ours (T2T-Vit)	T2T-Vit-14	**50.4**	84.2	**6.2**	10.1	78.4	130.9	**54.1**	248.3	76.1	330.5

We combine our three methods with CSRNet+ and T2T-Vit to obtain counting results that are either competitive or better than the current state-of-the-art, highlighting their potential for counting. **Bold** indicates the best counting results, and underline indicates the second-best counting results

the probability map to count the objects. All baselines use the same backbone and training optimizations as our method for a fair comparison.

As shown in Table 2, the standard distillation just marginally improves the counting performance. For instance, on ShanghaiTech Part_A, the MAE reduces from 60.1 to 59.8, and on UCF_QNRF, it decreases from 90.8 to 90.1, when using the T2T-Vit base network. However, all four other baselines outperform standard distillation. Notably, our proposed method surpasses the performance of these four baselines [59.1 (Liu et al., 2019a), 58.2 (Modolo et al., 2021), 57.7 (Mo et al., 2020), 57.1 (Cheng et al., 2021)] on ShanghaiTech Part_A, achieving an impressive MAE of 56.5. On UCF_QNRF, our method outperforms all baselines except the fourth one (Cheng et al., 2021). It is worth mentioning that Cheng et al. (2021) utilizes a two-network cascade for improved results, albeit at the cost of inefficient inference.

Furthermore, we also observe a substantial reduction in MAE when applying our proposed method with the CSRNet+ base network. In summary, foreground distillation emerges as a straightforward yet effective approach, adaptable to any network, and capable of substantially enhancing counting accuracy.

In Fig. 5, we provide some examples of Gaussian and distilled density maps. As the auxiliary network is trained on images with background removal, we observe that the distilled density map more accurately captures foreground counting information, effectively reducing local noise and exhibiting improved alignment with the spatial distribution of objects when compared to the Gaussian density map. Thus, distillation provides a more natural relation between the foreground of the images and density maps.

In the density prediction loss function, as described in Eq. (7), we opt to replace the Gaussian density map with our distilled density map for the final training phase, rather than employing both simultaneously. To assess whether using both density maps could enhance performance, we conducted an experiment involving a combined loss with a dynamic weight. This can be expressed as: $\mathcal{L}{=}\lambda * \mathcal{L}_d + (1-\lambda) * \mathcal{L}_g$. Here, $\mathcal{L}_d$ represents our distillation loss, and $\mathcal{L}_g$ denotes the Gaussian density loss. We set $\lambda = 1/(1+\eta \cdot n)$, where n denotes the epoch number and η is the decay rate set to 0.01. When utilizing the T2T-Vit base network, we observe that the MAE is 56.9 on the ShanghaiTech Part_A dataset and 87.4 on UCF_QNRF when employing the combined loss, whereas it is 56.5 and 86.7 when using the foreground distillation loss alone. Consequently, we have chosen not to use the Gaussian density loss in conjunction with foreground distillation.

4.4 Local and Global Focus

Next, we demonstrate the effect of our proposed focus for free that includes focus from segmentation and focus from global density. We first compare each to their respective baseline approaches. Then, we report the results of the combination of the two methods.

Local Segmentation Focus We first analyze the effect of the proposed segmentation focus. We compare to two baselines. The first performs counting using the base network, where the loss is only optimized with respect to the density map estimation. The second baseline adds spatial attention on top of this base network, as proposed in Chen et al. (2017). The results are shown in Table 3. For ShanghaiTech Part_A, the T2T-Vit base network obtains an MAE of 60.1 and the addition of spatial-attention actually increases the count error to 59.2 MAE, as it fails to emphasize relevant features. In contrast, our proposed focus from segmentation can explicitly guide the network to focus on task-relevant regions, and it reduces the count error from 60.1 to 58.4 MAE.

Global Density Focus Next, we demonstrate the effect of our proposed global density focus. For this experiment, we again compare it with the SE module (Hu et al., 2018) and the channel attention module of CBAM (Woo et al., 2018), while keeping the base network unchanged. To ensure a fair comparison, we use the same MLP network for all the approaches. The results of our experiment are presented in Table 3. Both SE and CBAM can already improve the base network's performance based on both CSRNet+ and T2T-Vit. Our proposed method outperforms SE and CBAM, reducing the count error even further, due to its explicit supervision.

Combined Focus for Free In the aforementioned experiments, we have shown that each focus matters for counting. In this experiment, we combine these two focuses for more accurate counting, in view that these two focuses aid density map estimation respectively from a local and global perspective, complementing each other. The results are shown in Table 3. The combination achieves a reduced count error of 56.9 MAE on ShanghaiTech Part_A, and obtains a reduced count error of 85.8 MAE on UCF_QNRF.

4.5 Comparative Evaluation

For the final experiments, we first show the effect of combining the three methods we propose for improving counting by repurposing point annotations. Then we compare to the state-of-the-art in counting.

Combining the Three Approaches Until now, we have established that every method plays a role in counting on its own. In this study, we aim to assess whether these methods are also complementary to each other. The results of this experiment are shown in Table 4, utilizing both T2T-Vit and CSRNet+ as backbones. Compared to standard counting with just the base networks, we reduce the MAE considerably by handling occluded objects with occlusion simulation (Sect. 3.1). By combining occlusion simulation with foreground distillation (Sect. 3.2) or focus for free (Sect. 3.3), or combining foreground distillation with focus for free, the MAE is further reduced. Combining all three solutions achieves the lowest count error across datasets and base networks.

Comparison to the State-of-the-Art As the ultimate test, we draw a comparison to the state-of-the-art in counting on five datasets. We report our results using both CSRNet+ and T2T-Vit as base networks, and both ℓ_1 and distribution matching (DM) (Wang et al., 2020a) as loss functions. Table 5 shows the comparative evaluation over all datasets and metrics. The results show that despite using canonical counting backbones, which on their own can not compete with the state-of-the-art, we obtain the best or second-best counting results on 8 of the 10 metrics when using ℓ_1 loss. This highlights the potential of our simple solutions. We also compare to recent counting alternatives that utilize the advanced DM

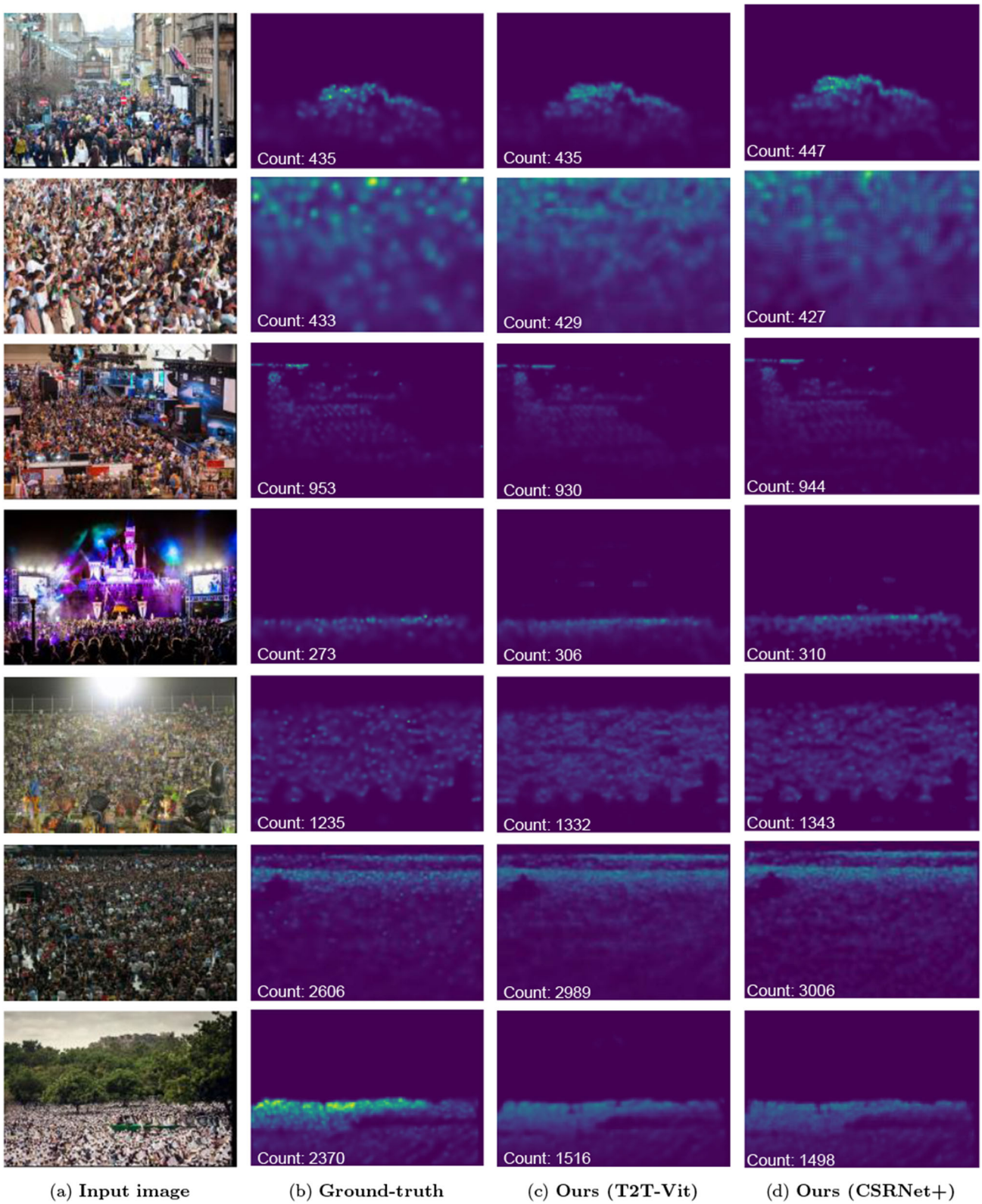


Fig. 6 Success and failure cases obtained with ℓ_1-norm loss. When objects are individually visible, we can count them accurately (first four rows). Further improvements are required for extremely dense scenes where individual objects are hard to distinguish, or where objects blend with the context (last three rows)

Fig. 7 Success and failure cases obtained with DM loss. When objects are individually visible, we can count them accurately (first four rows). Further improvements are required for extremely dense scenes where individual objects are hard to distinguish, or where objects blend with the context (last three rows)

loss and find that we maintain the best or second-best results for all metrics. We conclude that with our proposed methods, we can make further reduce the counting errors. We show some success and failure results obtained by our methods in Figs. 6 and 7. Even in challenging scenes with relatively sparse small objects or relatively dense large objects, our method is able to achieve an accurate count (first four rows). Our approach fails when dealing with extremely dense scenes where individual objects are hard to distinguish, or where objects blend with the context (last three rows). Such scenarios remain open counting challenges.

Complexity Analysis The occlusion simulation and foreground distillation have no impact on inference time because they are used only in the training phase. It's only the local segmentation focus and global density focus that introduce some complexity and increase inference time. However, it's worth noting that we've designed these focus components to be lightweight. Specifically, when using the CSRNet+ base network, we introduce just one convolution layer with parameters of $C \times 2 \times 1 \times 1$ for the local segmentation focus and two dense-connected layers with parameters of $C \times C$ for the global density focus (where $C = 64$ in our study). As a result, our methods cause only a slight increase in inference time, for example, from 0.27 s (CSRNet+ alone) to 0.29 s (CSRNet+ with our focus) per image when working with the UC_QNRF dataset.

5 Conclusion

This paper aims to enhance density-based counting by leveraging point annotations beyond their original purpose of creating a density map. We propose occlusion augmentation, foreground distillation, and focus for free. Foreground distillation involves generating foreground masks from point annotations and utilizing these masks to distill foreground knowledge. By doing so, this approach mitigates the influence of background pixels on counting accuracy. Counting-specific occlusion augmentation leverages point annotations to simulate occluded objects in both the input and density images, enabling the network to become more robust to occlusions. This approach improves the network's ability to handle challenging and crowded scenarios where objects may be partially or fully overlapping. Additionally, we utilize explicit supervision derived from point annotations to offer both local and global focus. This approach ensures that the network pays attention to crucial features in the counting process. These approaches work complementary to improve counting accuracy and can be seamlessly integrated into various network architectures while accommodating different loss functions. The effectiveness of the methods is evaluated through experiments on five benchmark datasets. The results demonstrate that the proposed approaches consistently achieve remarkable counting results across various scenes. However, the work also highlights that counting in extremely dense scenes still remains an open problem. This suggests that further advancements are necessary to attain perfect counting in such challenging scenarios.

References

Babu Sam, D., Sajjan, N.N., Venkatesh Babu, R., & Srinivasan, M. (2018). Divide and grow: Capturing huge diversity in crowd images with incrementally growing CNN. In *CVPR*.

Bai, S., He, Z., Qiao, Y., Hu, H., Wu, W., & Yan, J. (2020). Adaptive dilated network with self-correction supervision for counting. In *CVPR*.

Brostow, G.J., & Cipolla, R. (2006). Unsupervised bayesian detection of independent motion in crowds. In *CVPR*.

Cao X, Wang, Z., Zhao, Y., & Su, F. (2018). Scale aggregation network for accurate and efficient crowd counting. In *ECCV*.

Chan, A.B., & Vasconcelos, N. (2009). Bayesian poisson regression for crowd counting. In *ICCV*, pp 545–551.

Chan, A. B., & Vasconcelos, N. (2011). Counting people with low-level features and Bayesian regression. *IEEE Transactions on Image Processing, 21*(4), 2160–2177.

Chan, A.B., Liang, Z.S.J., &Vasconcelos, N. (2008). Privacy preserving crowd monitoring: Counting people without people models or tracking. In *CVPR*.

Chen, K., Loy, C.C., Gong, S., & Xiang, T. (2012) Feature mining for localised crowd counting. In *BMVC*.

Chen, L., Zhang, H., Xiao, J., Nie, L., Shao, J., Liu, W., Chua, T.S. (2017). Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *CVPR*.

Chen, S., Shi, Z., Mettes, P., & Snoek, C.G. (2021). Social fabric: Tubelet compositions for video relation detection. In *ICCV*.

Cheng, J., Xiong, H., Cao, Z., & Lu, H. (2021). Decoupled two-stage crowd counting and beyond. *IEEE Transactions on Image Processing, 30*, 2862–2875.

Cheng, Z.Q., Li, J.X., Dai, Q., Wu, X., & Hauptmann, A.G. (2019a). Learning spatial awareness to improve crowd counting. In *ICCV*.

Cheng, Z.Q., Li, J.X., Dai, Q., Wu, X., He, J.Y., & Hauptmann, A, (2019b), Improving the learning of multi-column convolutional neural network for crowd counting. In *ACM MM*.

Cheng, Z.Q., Dai, Q., Li, H., Song, J., Wu, X., & Hauptmann, A.G. (2022). Rethinking spatial invariance of convolutional networks for object counting. In *CVPR*.

DeVries, T., & Taylor, G.W. (2017). Improved regularization of convolutional neural networks with cutout. arXiv:1708.04552

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., & Uszkoreit, J., (2020) An image is worth 16x16 words: Transformers for image recognition at scale. arXiv:2010.11929

Gao, J., Gong, M., & Li, X. (2022). Congested crowd instance localization with dilated convolutional swin transformer. *Neurocomputing, 513*, 94–103.

Ghiasi, G., Cui, Y., Srinivas, A., Qian, R., Lin, T.Y., Cubuk, E.D., Le, Q.V., & Zoph, B. (2021). Simple copy-paste is a strong data augmentation method for instance segmentation. In *CVPR*.

Guerrero-Gómez-Olmedo, R., Torre-Jiménez, B., López-Sastre, R., Maldonado-Bascón, S., Onoro-Rubio, D. (2015). Extremely overlapping vehicle counting. In *IbPRIA*.

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. arXiv:1503.02531

Hu. J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In *CVPR*.

Hu, Y., Jiang, X., Liu, X., Zhang, B., Han, J., Cao, X., & Doermann, D. (2020). Nas-count: Counting-by-density with neural architecture search. In *ECCV*.

Huang, S., Li, X., Cheng, Z.Q., Zhang, Z., & Hauptmann, A. (2020). Stacked pooling for boosting scale invariance of crowd counting. In *ICASSP*.

Idrees H, Saleemi I, Seibert C, & Shah M (2013) Multi-source multi-scale counting in extremely dense crowd images. In *CVPR*

Idrees, H., Tayyab, M., Athrey, K., Zhang, D., Al-Maadeed, S., Rajpoot, N., & Shah, M. (2018). Composition loss for counting, density map estimation and localization in dense crowds. In *ECCV*.

Jiang, X., Xiao, Z., Zhang, B., Zhen, X., Cao, X., Doermann, D., Shao, L. (2019). Crowd counting and density estimation by trellis encoder-decoder networks. In *CVPR*.

Jiang, X., Zhang, L., Xu, M., Zhang, T., Lv, P., Zhou, B., Yang, X., & Pang, Y. (2020a). Attention scaling for crowd counting. In *CVPR*.

Jiang, X., Zhang, L., Zhang, T., Lv, P., Zhou, B., Pang, Y., Xu, M., & Xu, C. (2020). Density-aware multi-task learning for crowd counting. *IEEE Transactions on Multimedia, 23*, 443–453.

Kang, D., Dhar, D., & Chan, A. (2020). Incorporating side information by adaptive convolution. *International Journal of Computer Vision, 128*, 2897–2918.

Leibe, B., Seemann, E., & Schiele, B. (2005). Pedestrian detection in crowded scenes. In *CVPR*.

Lempitsky, V., & Zisserman, A. (2010). Learning to count objects in images. In *NeurIPS*.

Li M, Zhang Z, Huang K, Tan T (2008) Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In: ICPR, pp 1–4

Li, Y., Zhang, X., & Chen, D. (2018). Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *CVPR*.

Liang, D., Chen, X., Xu, W., Zhou, Y., & Bai, X. (2022). Transcrowd: Weakly-supervised crowd counting with transformers. *SCIENCE CHINA Information Sciences, 65*(6), 160104.

Liang, D., Xu, W., Zhu, Y., & Zhou, Y. (2022). Focal inverse distance transform maps for crowd localization. *IEEE Transactions on Multimedia, 25*, 6040–6052.

Lin, H., Ma, Z., Ji, R., Wang, Y., & Hong, X. (2022). Boosting crowd counting via multifaceted attention. In *CVPR*.

Lin, S. F., Chen, J. Y., & Chao, H. X. (2001). Estimation of number of people in crowded scenes using perspective transformation. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 31*(6), 645–654.

Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P. (2017). Focal loss for dense object detection. In *ICCV*.

Liu, J., Gao, C., Meng, D., Hauptmann, A.G. (2018a). Decidenet: Counting varying density crowds through attention guided detection and density estimation. In *CVPR*.

Liu, L., Wang, H., Li, G., Ouyang, W., & Lin, L. (2018b). Crowd counting using deep recurrent spatial-aware network. In *IJCAI*.

Liu, N., Long, Y., Zou, C., Niu, Q., Pan, L., Wu, H. (2019a). Adcrowdnet: An attention-injective deformable convolutional network for crowd understanding. In *CVPR*.

Liu, W., Salzmann, M., Fua, P. (2019b). Context-aware crowd counting. In *CVPR*.

Liu, X., van de Weijer, J., Bagdanov, A.D. (2018c). Leveraging unlabeled data for crowd counting by learning to rank. In *CVPR*.

Liu, X., Yang, J., Ding, W., Wang, T., Wang, Z., & Xiong, J. (2020). Adaptive mixture regression network with local counting map for crowd counting. In *ECCV*.

Liu, X., Li, G., Han, Z., Zhang, W., Yang, Y., Huang, Q., Sebe, N. (2021). Exploiting sample correlation for crowd counting with multi-expert network. In *ICCV*.

Liu, Y., Cheng, M. M., Fan, D. P., Zhang, L., Bian, J. W., & Tao, D. (2022). Semantic edge detection with diverse deep supervision. *International Journal of Computer Vision, 130*(1), 179–198.

Ma, Z., Wei, X., Hong, X., & Gong, Y. (2019). Bayesian loss for crowd count estimation with point supervision. In: *ICCV*.

Ma, Z., Wei, X., Hong, X., & Gong, Y. (2020) Learning scales from points: A scale-aware probabilistic model for crowd counting. In *ACM MM*.

Ma, Z., Hong, X., Wei, X., Qiu, Y., Gong, Y. (2021). Towards a universal model for cross-dataset crowd counting. In *ICCV*.

Mo, H., Ren, W., Xiong, Y., Pan, X., Zhou, Z., Cao, X., & Wu, W. (2020). Background noise filtering and distribution dividing for crowd counting. *IEEE Transactions on Image Processing, 29*, 8199–8212.

Modolo, D., Shuai, B., Varior, R.R., & Tighe, J. (2021). Understanding the impact of mistakes on background regions in crowd counting. In *WACV*.

Onoro-Rubio, D., & López-Sastre, R.J. (2016) Towards perspective-free object counting with deep learning. In *ECCV*.

Peyré, G., Cuturi, M., et al. (2019). Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning, 11*(5–6), 355–607.

Pham, V.Q., Kozakaya, T., Yamaguchi, O., & Okada, R. (2015). Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation. In *ICCV*.

Qian, Y., Zhang, L., Hong, X., Donovan, C.R., & Arandjelovic, O. (2022). Segmentation assisted u-shaped multi-scale transformer for crowd counting. In *BMVC*.

Rabaud, V., Belongie, S. (2006). Counting crowded moving objects. In *CVPR*.

Ranjan, V., Le, H., & Hoai, M. (2018). Iterative crowd counting. In *ECCV*

Rong, L., & Li, C. (2021). Coarse-and fine-grained attention network with background-aware loss for crowd density map estimation. In *WACV*.

Sam, D.B., Surya, S., & Babu, R.V. (2017). Switching convolutional neural network for crowd counting. In *CVPR*.

Shen, Z., Xu, Y., Ni, B., Wang, M., Hu, J., Yang, X. (2018a). Crowd counting via adversarial cross-scale consistency pursuit. In *CVPR*.

Shen, Z., Xu, Y., Ni, B., Wang, M., Hu, J., & Yang, X. (2018b). Crowd counting via adversarial cross-scale consistency pursuit. In *CVPR*.

Shi, M., Yang, Z., Xu, C., & Chen, Q. (2019a). Revisiting perspective information for efficient crowd counting. In *CVPR*.

Shi, Z., Zhang, L., Liu, Y., Cao, X., & Ye, Y., Cheng, M.M., Zheng, G. (2018a). Crowd counting with deep negative correlation learning. In *CVPR*.

Shi, Z., Zhang, L., Sun, Y., & Ye, Y. (2018). Multiscale multitask deep netvlad for crowd counting. *IEEE TII, 14*(11), 4953–4962.

Shi, Z., Mettes, P., & Snoek, C.G.M. (2019b). Counting with focus for free. In *ICCV*,

Shi, Z., Chen, Y., Gavves, E., Mettes, P., & Snoek, C. G. (2021). Unsharp mask guided filtering. *IEEE Transactions on Image Processing, 30*, 7472–7485.

Shi, Z., Mettes, P., Zheng, G., Snoek, C. (2021b). Frequency-supervised mr-to-ct image synthesis. In *MICCAI workshop on deep generative models and data augmentation*.

Shi, Z., Sun, Y., Zhang, M. (2024). Training-free object counting with prompts. In *WACV*.

Shu, W., Wan, J., Tan, K.C., Kwong, S., Chan, A.B. (2022). Crowd counting in the frequency domain. In *CVPR*.

Sindagi, V.A., Patel, V.M. (2017). Generating high-quality crowd density maps using contextual pyramid cnns. In *ICCV* pp. 1861–1870.

Sindagi, V. A., & Patel, V. M. (2019). Ha-CCN: Hierarchical attention-based crowd counting network. *IEEE Transactions on Image Processing, 29*, 323–335.

Sindagi, V. A., Yasarla, R., & Patel, V. M. (2020). Jhu-crowd++: Large-scale crowd counting dataset and a benchmark method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. https://doi.org/10.1109/TPAMI.2020.3035969

Song, Q., Wang, C., Jiang, Z., Wang, Y., Tai, Y., Wang, C., Li, J., Huang, F., Wu, Y. (2021). Rethinking counting and localization in crowds: A purely point-based framework. In *ICCV*.

Sun, G., Liu, Y., Probst, T., Paudel, D.P., Popovic, N., & Van Gool, L. (2021). Boosting crowd counting with transformers. arXiv:2105.10926

Tian, Y., Chu, X., Wang, H. (2021). Cctrans: Simplifying and improving crowd counting with transformer. arXiv:2109.14483

Topkaya, I.S., Erdogan, H., Porikli, F. (2014). Counting people by clustering person detector outputs. In *AVSS*

Tran, N.H., Huy, T.D., Duong, S.T., Nguyen, P., Hung, D.H., Nguyen, C.D.T., Bui, T., Truong, S.Q., & VinBrain, J. (2022). Improving local features with relevant spatial information by vision transformer for crowd counting. In *BMVC*.

Wan, J., Chan, A. (2020). Modeling noisy annotations for crowd counting. In *NeurIPS*

Wan, J., Wang, Q., & Chan, A. B. (2020). Kernel-based density map generation for dense object counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 44*(3), 1357–1370.

Wan, J., Liu, Z., Chan, A.B. (2021). A generalized loss function for crowd counting and localization. In *CVPR*.

Wang, B., Liu, H., Samaras, D., Hoai, M. (2020a). Distribution matching for crowd counting. In *NeurIPS*.

Wang, C., Song, Q., Zhang, B., Wang, Y., Tai, Y., Hu, X., Wang, C., Li, J., Ma, J., & Wu, Y. (2021). Uniformity in heterogeneity: Diving deep into count interval partition for crowd counting. In *ICCV*.

Wang, Q., Gao, J., Lin, W., & Li, X. (2020). Nwpu-crowd: A large-scale benchmark for crowd counting and localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 43*(6), 2141–2149.

Wang, Y., Ma, Z., Wei, X., Zheng, S., Wang, Y., & Hong, X. (2022). Eccnas: Efficient crowd counting neural architecture search. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 18*(1s), 1–19.

Woo, S., Park, J., Lee, J.Y., & Kweon, I.S. (2018). Cbam: Convolutional block attention module. In *ECCV*.

Wu, B., & Nevatia, R. (2007). Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors. *International Journal of Computer Vision, 75*(2), 247.

Xiong, H., Lu, H., Liu, C., Liu, L., Shen, C., & Cao, Z. (2023). From open set to closed set: Supervised spatial divide-and-conquer for object counting. *International Journal of Computer Vision., 131*(7), 1722–1740.

Xu, C., Liang, D., Xu, Y., Bai, S., Zhan, W., Bai, X., & Tomizuka, M. (2022). Autoscale: Learning to scale for crowd counting. *International Journal of Computer Vision, 130*(2), 405–434.

Yan, Z., Yuan, Y., Zuo, W., Tan, X., Wang, Y., Wen, S., & Ding, E. (2019). Perspective-guided convolution networks for crowd counting. In *ICCV*.

Yang, S., Guo, W., Ren, Y. (2022). Crowdformer: An overlap patching vision transformer for top-down crowd counting. In *IJCAI*.

Yang, Y., Li, G., Wu, Z., Su, L., Huang, Q., Sebe, N. (2020). Reverse perspective network for perspective-aware object counting. In *CVPR*.

Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z.H., Tay, F.E., Feng, J., & Yan, S. (2021). Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *ICCV*.

Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., & Yoo, Y. (2019). Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*.

Zhang, J., Cheng, Z.Q., Wu, X., Li, W., Qiao, J.J. (2022). Crossnet: Boosting crowd counting with localization. In *ACM MM*.

Zhang, Q., & Chan, A. B. (2022). Wide-area crowd counting: Multi-view fusion networks for counting in large scenes. *International Journal of Computer Vision, 130*(8), 1938–1960.

Zhang, Y., Zhou, D., Chen, S., Gao, S., Ma, Y. (2016). Single-image crowd counting via multi-column convolutional neural network. In *CVPR*.

Zhao, M., Zhang, J., Zhang, C., & Zhang, W. (2019). Leveraging heterogeneous auxiliary tasks to assist crowd counting. In *CVPR*.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.