Counting with Focus for Free

Zenglin Shi, Pascal Mettes, and Cees G. M. Snoek
University of Amsterdam

Abstract

This paper aims to count arbitrary objects in images. The leading counting approaches start from point annotations per object from which they construct density maps. Then, their training objective transforms input images to density maps through deep convolutional networks. We posit that the point annotations serve more supervision purposes than just constructing density maps. We introduce ways to repurpose the points for free. First, we propose supervised focus from segmentation, where points are converted into binary maps. The binary maps are combined with a network branch and accompanying loss function to focus on areas of interest. Second, we propose supervised focus from global density, where the ratio of point annotations to image pixels is used in another branch to regularize the overall density estimation. To assist both the density estimation and the focus from segmentation, we also introduce an improved kernel size estimator for the point annotations. Experiments on six datasets show that all our contributions reduce the counting error, regardless of the base network, resulting in state-of-the-art accuracy using only a single network. Finally, we are the first to count on WIDER FACE, allowing us to show the benefits of our approach in handling varying object scales and crowding levels. Code is available at https://github.com/shizenglin/Counting-with-Focus-for-Free

1. Introduction

This paper strives to count objects in images, whether they are people in crowds [10,35,40], cars in traffic jams [7] or cells in petri dishes [22]. The leading approaches for this challenging problem count by summing the pixels in a density map [13] as estimated with a convolutional neural network, *e.g.* [3, 11, 14, 22]. While this line of work has shown to be effective, the rich source of supervision from the point annotations is only used to construct the density maps for training. The premise of this work is that point annotations can be repurposed to further supervise counting optimization in deep networks, for free.

The main contribution of this paper is summarized in

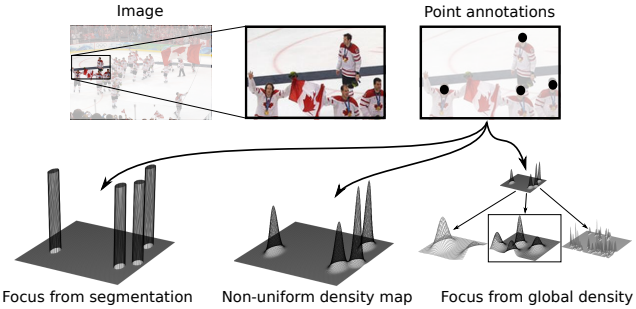

Figure 1: **Focus for free** in counting. From point supervision, we learn to obtain a focus from segmentation, a focus from global density, and an improved density maps. Combined, they result in better counting estimation irrespective of the base network.

Figure 1. Besides creating density maps, we show that points can be exploited as free supervision signal in two other ways. The first is focus from segmentation. From point annotations, we construct binary segmentation maps and use them in a separate network branch with an accompanying segmentation loss to focus on areas of interest only. The second is focus from global density. The relative amount of point annotations in images is used to focus on the global image density through another branch and loss function. Both forms of focus are integrated with the density estimation in a single network trained end-to-end with a multi-level loss. In standard attention mechanisms [9, 12, 19, 33], the weighing map is indirectly learned from a task-specific objective, *e.g.*image classification or object counting. We also rely on task-specific supervision, but we explicitly add novel supervised network branches for the segment and density weighting maps. We derive the necessary supervision from provided point annotations and name it focus for free.

Overall, we make three contributions in this paper: (*i*) We propose supervised focus from segmentation, a network branch which guides the counting network to focus on areas of interest. The supervision is obtained from the already provided point annotations. (*ii*) We propose supervised focus from global density, a branch which regularizes the counting network to learn a matching global den-

sity. Again the supervision is obtained for free from the point annotations. (*iii*) We introduce a new kernel density estimator for point annotations with non-uniform point distributions. For the deep network, we design an improved encoder-decoder network to deal with varying object scales in images. Experimental evaluation on six counting datasets shows the benefits of our focus for free, kernel estimation, and end-to-end network architecture, resulting in state-of-the-art counting accuracy. To further demonstrate the potential of our approach for counting under varying object scales and crowding levels, we provide the first counting results on WIDER FACE, normally used for large-scale face detection [35].

2. Related Work

Density-based counting. Deep convolutional networks are widely adopted for counting by estimating density maps from images. Early works, *e.g.* [24, 30, 37, 40], 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 [14] and conflicts resulting from optimization among different columns [1, 27].

Due to their architectural simplicity and training efficiency, single column deep networks have received increasing interest *e.g.* [3, 14, 20, 21, 28]. Cao *et al.* [3] , for example, propose an encoder-decoder network to predict high-resolution and high-quality density maps using a scale aggregation module. Li *et al.* [14] combine a VGG network with dilated convolution layers to aggregate multi-scale contextual information. Liu *et al.* [21] rely on a single network by leveraging abundantly available unlabeled crowd imagery in a learning-to-rank framework. Shi *et al.* [28] train a single VGG network with a deep negative correlation learning strategy to reduce the risk of over-fitting. We also employ single column networks, but rather than focusing solely on density map estimation, we repurpose the point annotations in multiple ways to improve counting.

Recently, multi-task networks have shown to reduce the counting error [1, 10, 19, 25–27, 29]. Sam *et al.* [26], for example, train a classifier to select the optimal regressor from multiple independent regressors for particular input patches. Ranjan *et al.* [25] rely on one network to predict a high resolution density map and a helper-network to predict a density map at a low resolution. 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 propose network branches with supervised focus from segmentation and global density to repurpose the point annotations for free. Our focus for free benefits counting regardless of the base network, and is complementary to other state-of-the-art solutions.

Counting with attention. Attention mechanisms [34] have enabled progress in a wide variety of computer vision challenges [4, 6, 15, 39, 41]. Soft attention is the most widely used since it is differentiable and thus can be directly incorporated in an end-to-end trainable network. The common way to incorporate soft attention is to add a network branch with one or more hidden layers to learn an attention map which assigns different weights to different regions of an image. Spatial and channel attention are two well explored types of soft attention [4, 33]. Spatial attention learns a weighting map over the spatial coordinates of the feature map, while channel attention does so for the feature channels of the map.

A few works have investigated density-based counting with spatial attention [8, 12, 19]. Liu *et al.* [19], for example, estimate the density of a crowd by generating separate detection- and regression-based density maps. They fuse these two density maps guided by an attention map, which is implicitly learned together with the density map regression loss. While we share the notion of assisting the density-based counting with a focus, we show in this work that such an attention does not need to be learned from scratch and instead can be derived from the existing point annotations. More specifically, we construct a segmentation map and a global density derived from the ground-truth annotated points as two additional, yet free, supervision signals for better counting.

3. Focus for Free

We formulate the counting task as a density map estimation problem, see *e.g.* [13, 28, 40]. Given N training images $\{(X_i, \mathcal{P}_i)\}_{i=1}^{N}$, with $X_i \subset \mathcal{X}$ the input image and $\mathcal{P}_i$ a set of point annotations, one for each object, we use the point annotations to create a ground-truth density map by convolving the points with a Gaussian kernel,

$$D_i(p) = \sum_{P \in \mathcal{P}_i} \mathcal{N}(p|\mu = P, \sigma_P^2), \tag{1}$$

where p denotes a pixel location, P denotes a single point annotation and $\mathcal{N}(p|\mu = P, \sigma_P^2)$ is a normalized Gaussian kernel with mean P and an isotropic covariance σ_P^2. The global object count T_i of image X_i can be obtained by summing all pixel values within the density map D_i, *i.e.*, $T_i = \sum_{p \in X_i} D_i(p)$. Learning a transformation from input images to density maps is done through deep convolutional networks. Let $\Psi(X) : \mathbb{R}^{3 \times W \times H} \mapsto \mathbb{R}^{W \times H}$ denote such a mapping given an arbitrary deep network Ψ for image X, with W and H the width and height of the image. In this paper, we investigate two ways that repurpose the point annotations to help supervising the network Ψ from input images to density maps. An overview of our approach, in which multiple branches are combined on top of a base network, is shown in Figure 2.

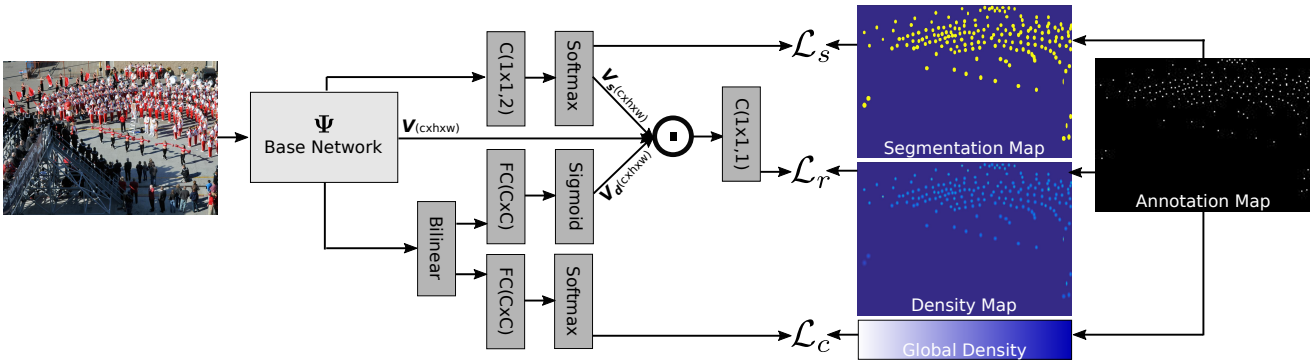

Figure 2: **Overview of our approach**. Top branch: focus from segmentation learns a focus map V_s with the aid of a segmentation map (Section 3.1). Bottom branch: focus from global density learns a focus map V_d with the aid of a global density (Section 3.2). Both supervision signals are obtained from the same point-annotations, for which we introduce an improved kernel estimator (Section 3.3). Both branches with focus for free are integrated with the output of a base network by element-wise multiplication and end-to-end optimized through a multi-level loss (Section 3.4).

3.1. Focus from segmentation

The first way to repurpose the point annotations is to provide a spatial focus. Intuitively, pixels that are within a specific range of any point annotation should be of high focus, while pixels in undesired regions should be mostly disregarded. In the standard setup where the optimization is solely dependent on the density map, each pixel counts equally to the network loss. Given that only a fraction of the pixels are near point annotations, the loss will be dominated by the majority of irrelevant pixels. To overcome this limitation, we reuse the point annotations to create a binary segmentation map and exploit this map to provide the focused supervision through a stand-alone loss function.

Segmentation map. The binary segmentation map is obtained as a function of the point annotations and their estimated variance. The binary value for each pixel p in training image i is determined as:

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

Equation 2 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.

Segmentation focus. Let $V \in \mathbb{R}^{C \times W \times H}$ denote the output of the base network. We add a new branch on top of the network denoted as $\mathcal{F}_s$ with network parameters θ_s. Furthermore, let θ_n denote the parameters of the base network. We propose a per-pixel weighted focal loss [17] to obtain a supervised focus from segmentation for input image X:

$$\mathcal{L}_s(X; \theta_n, \theta_s) = \sum_{l \in \{0,1\}} -\alpha^l S^l \\ (1 - \mathcal{F}_s(X; \theta_n, \theta_s))^{\gamma_s} log(\mathcal{F}_s(X; \theta_n, \theta_s)), \tag{3}$$

where $\alpha^l = 1 - \frac{|S^l|}{|S|}$. The focal parameter γ_s is set to 2 throughout this network, as recommended by [17]. The segmentation branch is visualized at the top of Figure 2.

Network details. After the output of the base network, we perform a 1×1 convolution layer with parameters $\theta_s \in \mathbb{R}^{C \times 2 \times 1 \times 1}$, followed by a softmax function δ to generate a per-pixel probability map $\mathbb{P}_i = \delta(\theta_s V) \in \mathbb{R}^{2 \times W \times H}$. From this probability map, the second value along the first dimension represents the probability of each pixel being part of the segmentation foreground. We furthermore tile this slice C times to construct a separate output tensor $V_s \in \mathbb{R}^{C \times W \times H}$, which will be used in the density estimation branch itself.

3.2. Focus from global density

Next to a spatial focus, point annotations can also be repurposed by examining their context. It is well known that low density crowds exhibit coarse texture patterns while high density crowds exhibit very fine texture patterns. Here, we exploit this knowledge for the task of counting. Given a network output $V \in \mathbb{R}^{W \times H \times C}$, we employ a bilinear pooling layer [5, 18] to capture the feature statistics in a global context, which is known to be particularly suitable for texture and fine-grained recognition [5, 18]. In this work, we match global contextual patterns to the distribution of points in training images to obtain a supervised focus from global density.

Global density. For patch j in training image i, its global density is given as:

$$G_{j,i} = \frac{|\mathcal{P}_{j,i}|}{L}, \tag{4}$$

where $|\mathcal{P}_{j,i}|$ denotes the number of point annotations in patch j and L denotes the global density step size, which

is computed for a dataset as:

$$L = \left\lfloor \max_{i=1,..,N} \left(\frac{|\mathcal{P}_i|}{Z_i} \cdot Z_{j,i} \right) / M \right\rfloor + 1, \tag{5}$$

with Z_i and $Z_{j,i}$ the number of pixels in image i and patch j respectively. Intuitively, the step size computes the maximum global density over image patches and M states how many global density levels are used overall.

Global density focus. With $V \in \mathbb{R}^{C \times W \times H}$ again the output of the base network, we add a second new branch $\mathcal{F}_c$ with network parameters θ_c. We propose the following global density loss function:

$$\mathcal{L}_c(X;\theta_n,\theta_c) = \sum_{l \in \{0,1,..,M\}} -G^l \\ (1 - \mathcal{F}_c(X;\theta_n,\theta_c))^{\gamma_c} log(\mathcal{F}_c(X;\theta_n,\theta_c)), \tag{6}$$

where γ_c is set to 2 as well. The above loss function aims to match the global density of the estimated density map with the global density of the ground truth density map. The corresponding global density branch is visualized at the bottom of Figure 2.

Network details. For network output V, we first perform an outer product $B = VV^T \in \mathbb{R}^{C \times C}$, followed by a mean pooling along the second dimension to aggregate the bilinear features over the image, *i.e.* $\hat{B} = \frac{1}{C} \sum_{i=1}^{C} B[:,i] \in \mathbb{R}^{C \times 1}$. The bilinear vector $\hat{B}$ is ℓ_2-normalized, followed by signed square root normalization, which has shown to be effective in bilinear pooling [18]. Then we use a fully connected layer with parameters $\theta_c \in \mathbb{R}^{C \times M}$ followed by a softmax function δ_c to make individual prediction $\mathbb{C} = \delta_c(\theta_c \hat{B}) \in \mathbb{R}^{M \times 1}$ for the global density. Furthermore, another fully-connected layer with parameters $\theta_d \in \mathbb{R}^{C \times C}$ followed by sigmoid function δ_d also on top of the bilinear pooling layer is added to generate global density focus output $\mathbb{D} = \delta_d(\theta_d \hat{B}) \in \mathbb{R}^{C \times 1}$. We note that this results in a focus over the channel dimensions, complementary to the focus over the spatial dimensions from segmentation. Akin to the focus from segmentation, we tile the output vector into $V_d \in \mathbb{R}^{C \times W \times H}$, also to be used in the density estimation branch.

3.3. Non-uniform kernel estimation

Both the density estimation itself and the focus from segmentation require a variance estimation for each point annotation, where the variance corresponds to the size of the object. Determining the variance σ_P for each point P is difficult because of object-size variations caused by perspective distortions. A common solution is to estimate the size (*i.e.* the variance) of an object as a function of the K nearest neighbour annotations, *e.g.* the Geometry-Adaptive Kernel of Zhang *et al.* [40]. However, this kernel is effective only under the assumption that objects in images are uniformly distributed, which typically does not happen in counting practice. As such, we introduce a simple kernel that estimates the variance of a point annotation P by splitting an image into local regions:

$$\sigma_P = \frac{1}{|R_{(w,h)}|} \sum_{a \in R_{(w,h)}} \beta \bar{d}_a, \quad \bar{d}_a = \frac{1}{K} \sum_{k=1}^{K} d_{k,a} \tag{7}$$

where w and h are the hyper-parameters which determine the range of point annotation P-centered local region R, and we set their value to one-eighth of image size in our experiments. a denotes an arbitrary point annotation located in R. $|R_{(w,h)}|$ means the number of p. $\bar{d}_p$ indicates the average distance between annotated point p and its k nearest neighbors, and β is a user-defined hyper-parameter. By estimating the variance of point annotations locally, we no longer have to assume that points are uniformly distributed over the whole image.

3.4. Architecture and optimization

Network. To maximize the ability to focus and use the most accurate kernel estimation, we want the network output to be of the same width and height as the input image. Recently, encoder-decoder networks have been transferred from other visual recognition tasks [16, 36] to counting [3, 25, 27, 38]. We found that to make the encoder-decoder architectures better suited for counting, the wide variation in object-scale under perspective distortions needs to be addressed. As such, in our encoder-decoder architecture a distiller module is added between the step from encoder to decoder. The purpose of this module is to aggregate multi-level information from the encoder by distilling the most vital information for counting.

For the encoder, we make the original dilated residual network [36] suitable for our task by changing the channel of the feature maps after level 4 from 256/512 to 96 to reduce the model's parameters for the sake of avoiding over-fitting, given the low amount of training examples in counting. After the encoder, the distiller module fuses the features from level 4, 5, 7 and 8 in the encoder module by using skip connections and a concatenation operation. Then four convolution layers are used to further process the fused features to obtain a more compact representation. The reason why we do not fuse the features from level 6 is that level 6 comprises convolution layers with large dilation rates, which is prone to cause gridding artifacts [31, 36]. Compared to other works which fuse multiple networks with different kernels to deal with object-scale variations [24, 30, 40], the proposed network aggregates the features from different layers which have different receptive fields, and is much more efficient and easy to train. The decoder module uses 3 deconvolution layers with a kernel size of 4×4 and a stride size of 2×2 to progressively

recover the spatial resolution. To avoid the checkerboard artifact problem caused by regular deconvolutional operation [23, 31], we add two convolution layers after each deconvolution layer. We provide a detailed ablation on the encoder-distiller-decoder network in the supplementary material.

Multi-level loss. The final counting network with a focus for free contains three branches, $\mathcal{F}_r$ for the pixel-wise density estimation, $\mathcal{F}_s$ for the binary segmentation, and $\mathcal{F}_c$ for the global density prediction. Let $(\theta_n, \theta_r, \theta_s, \theta_c, \theta_d)$ denote the network parameters for the base network and the branches. For the density estimation, we first combine the outputs of the base network V with the tiled outputs V_s and V_d from the focus for free. We fuse the three sources of information by element-wise multiplication and feed the fusion to a 1×1 convolution layer with parameters $\theta_r \in \mathbb{R}^{C \times 1 \times 1 \times 1}$, resulting in an output density map.

For the density estimation, the $L2$ loss is a common choice, but it is also known to be sensitive to outliers, which hampers generalization [2]. We prefer to learn the density estimation branch by jointly optimizing the $L2$ and $L1$ loss, which adds robustness to outliers:

$$\begin{aligned}\mathcal{L}_r(X;\theta_n,\theta_r,\theta_d) = \frac{1}{2} \parallel \mathcal{F}_r(X;\theta_n,\theta_r,\theta_d) - Y \parallel_2^2 + \\ \parallel \mathcal{F}_r(X;\theta_n,\theta_r,\theta_d) - Y \parallel_1,\end{aligned} \tag{8}$$

where Y denotes the ground truth density map. Empirically, we also find that this combined loss is preferred over only using the $L1$ or $L2$ loss. The loss functions of the three branches are summed to obtain the final objective function:

$$\begin{aligned}\mathcal{L}(X;\theta_n,\theta_r,\theta_s,\theta_c,\theta_d) = \lambda_r \mathcal{L}_r(X;\theta_n,\theta_r,\theta_d) + \\ \lambda_s \mathcal{L}_s(X;\theta_n,\theta_s) + \lambda_c \mathcal{L}_c(X;\theta_n,\theta_c),\end{aligned} \tag{9}$$

where $(\lambda_r, \lambda_s, \lambda_c)$ denote the weighting parameters of the different loss functions. Throughout this work these parameters are set to $(1, 10, 1)$, since the loss values of the segmentation branch are typically an order of magnitude lower than the others.

4. Experimental Setup

4.1. Datasets

ShanghaiTech [40] consists of 1198 images with 330,165 people. This dataset is divided into two parts: **Part_A** with 482 images in which crowds are mostly dense (33 to 3139 people), and **Part_B** with 716 images, where crowds are sparser (9 to 578 people). Each part is divided into a training and testing subset as specified in [40]. **TRANCOS** [7] contains 1,244 images from different roads to count vehicles, varying from 9 to 105. We train on the given training data (403 images) and validation data (420 images) without any other datasets, and we evaluate on the test data (421 images). **Dublin Cell Counting (DCC)** [22] is a cell microscopy dataset, consisting of 177 images, with a cell count from 0 to 100. For training 100 images are used, the remaining 77 form the test set. **UCF-QNRF** [10] is a recent large-scale crowd dataset, consisting of 1,535 images, with the count ranging from 49 to 12,865. For training 1201 images are used, the remaining 334 form the test set. **WIDER FACE** [35] is a face detection benchmark. In this paper, we repurpose it for counting as a complementary crowd dataset. Compared to ShanghaiTech [40] and UCF-QNRF [10], WIDER FACE is more challenging due to large variations in scale, occlusion, pose, and background clutter. Moreover, it contains more images, in total $32,203$, divided in 40% training, 10% validation and 50% testing. The ground truth of the test set is unavailable, so we report on the validation set. Each face is annotated by a bounding box, instead of a point, which enables us to evaluate our kernel estimator and allows for ablation under varying object scales and crowding levels.

4.2. Implementation details

Pre-processing. For all datasets, we normalize the input RGB images by dividing all values by 255. During training, we augment the images by randomly cropping 128×128 patches. No cropping is performed during testing.

Network. We implement our method with TensorFlow on a machine with a single GTX 1080 Ti GPU. The network is trained using Adam with a mini-batch of 16. We set the β_1 to 0.9, β_2 to 0.999 and the initial learning rate to 0.0001. Training is terminated after a maximum of 1000 epochs.

Kernel computation. For datasets with dense objects, *i.e.* ShanghaiTech Part_A, TRANCOS and UCF-QNRF, we use our proposed kernel with $\beta = 0.3$ and $k = 5$. For ShanghaiTech Part_B and DCC, we set the Gaussian kernel variance to $\sigma = 5$ and $\sigma = 10$ respectively, following [14, 28]. For WIDER FACE, we obtain the Gaussian kernel variance by leveraging the box annotations. For the focus from global density, we use $M = 8$ density levels for ShanghaiTech Part_A and UCF-QNRF, and 4 for the other datasets.

4.3. Evaluation metrics

Count error. We report the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) metrics given count estimates and ground truth counts [28, 37, 40]. Since these global metrics ignore where objects have been counted, we also report results using the Grid Average Mean absolute Error (GAME) metric. [7]. GAME aggregates count estimates over local regions as: $GAME(L) = \frac{1}{N} \cdot \sum_{n=1}^{N} (\sum_{l=1}^{4^L} |(y_n^l - \tilde{y}_n^l)|)$, with N the number of images and y_n^l and $\tilde{y}_n^l$ the ground truth and the estimated counts in a region l of the n^{th} image. 4^L denotes the number of grids, non-overlapping regions which cover the full image. When

Table 1: **Effect of focus from segmentation** in terms of MAE on ShanghaiTech Part_A and WIDER FACE. Across both datasets and across multiple object scales (small, medium, large), our approach outperforms the base network, even when adding spatial attention.

	Part_A	WIDER FACE			
	overall	small	medium	large	overall
Base network	74.8	9.2	2.7	2.2	4.7
w/ Spatial attention [4]	84.5	8.7	2.6	3.1	4.8
w/ Segmentation focus	**72.3**	**8.6**	**2.3**	**2.0**	**4.3**

L is set to 0 the GAME is equivalent to the MAE.

Density map quality. Finally, we report PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity in Image [32]), to evaluate the quality of the predicted density maps. We only report these on ShanghaiTech Part_A because they are not commonly reported on the other datasets.

5. Results

5.1. Focus from segmentation

We first analyze the effect of focus from segmentation on both ShanghaiTech Part_A and WIDER FACE. 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. Unless stated otherwise, the encoder-distiller-decoder network is used as base network in all experiments. The second baseline adds a spatial attention on top of this base network, as proposed in [4]. The results are shown in Table 1.

For ShanghaiTech Part_A, the base network obtains an MAE of 74.8. The addition of spatial-attention increases the count error to 84.5 MAE, as it fails to emphasize relevant features. In contrast, focus from segmentation can explicitly guide the network to focus on task-relevant regions and it reduces the count error from 74.8 to 72.3 MAE.

For WIDER FACE, the box annotations allow us to perform an ablation on the accuracy as a function of the object scale. We define the scale levels of each image as $I_{scale} = \frac{F_s}{F_n}$, where F_s and F_n denote face size and face number. We sort the test images in ascending order according to their scale level. Finally, the test images are divided uniformly into three sets: small, medium and large. In Table 1, we provide the results across multiple object scales. We observe that across all object scales, our approach is preferred, reducing the MAE from 4.7 (base network) and 4.8 (with spatial attention) to 4.3. The ablation also reveals why spatial attention is not very effective overall; while improvements are obtained when objects are small, spatial attention performs worse when objects are large. Segmentation focus from reused point annotations avoids such issues.

Table 2: **Effect of focus from global density** in terms of MAE on ShanghaiTech Part_A and WIDER FACE. Our approach is preferred for both datasets. The ablation study on WIDER FACE shows our focus from global density is most effective when scenes are sparse in number of objects.

	Part_A	WIDER FACE			
	overall	sparse	medium	dense	overall
Base network	74.8	2.1	2.5	9.5	4.7
w/ Channel attention [4]	73.4	1.6	2.3	**7.8**	3.9
w/ Squeeze-and-excitation [9]	72.6	1.7	**1.6**	**7.8**	3.7
w/ Global-density focus	**71.7**	**0.9**	**1.6**	8.0	**3.5**

5.2. Focus from global density

Next, we demonstrate the effect of our proposed focus from global density. For this experiment, we again compare to two baselines. Apart from the base network, we compare to the channel attention of [4] and the squeeze-and-excitation block of [9]. For fair comparison, we replace the mean pooling used in the channel attention of [4] with bilinear pooling as used in our method for the sake of better encoding global context cues. The counting results are shown in Table 2. Channel-attentions can reduce the error (from 74.8 to 73.4 and 72.6 MAE) in ShanghaiTech Part_A compared to using the base network only, since the attention maps are learned on top of a pooling layer which encodes global context cues. Our focus from global density reduces the count error further to 71.7 MAE due to more specific focus from free supervision.

To demonstrate that our focus has a lower error on different crowding levels, we perform a further ablation on WIDER FACE. We define the crowding levels of each image as $I_{crowding} = \frac{F_s}{I_s} * \frac{F_n}{I_s}$, where F_s, I_s, and F_n denote face size, image size, and face number respectively. Then we sort the test images in ascending order according to their global density level. Finally, the test images are divided uniformly into three sets, sparse, medium and dense. As shown in Table 2, our method achieves the lowest error especially when scenes are sparse.

5.3. Combined focus for free

In the aforementioned experiments, we have shown that each focus matters for counting. In this experiment, we combine them. The results are shown in Table 3. The combination achieves a reduced MAE of 67.9 on ShanghaiTech Part_A, and obtains a reduced MAE of 3.2 on WIDER FACE. We compare to alternative combined attention baselines, *i.e.*, spatial-channel attention [4] and the convolutional block attention module [33]. While the combinations of attentions achieves better results than using the base network alone, our approach is preferred across datasets, object scales, and crowding levels.

The focus for free is agnostic to the base network. To demonstrate this capability, we have applied it to four dif-

Table 3: **Effect of combined focus** in terms of MAE on ShanghaiTech Part_A and WIDER FACE. Across dataset, object scale, and crowding level our approach outperforms the base network and a combined spatial and channel attention variant.

	Part_A	**WIDER FACE**						
	overall	small	medium	large	sparse	medium	dense	overall
Base network	74.8	9.2	2.7	2.2	2.1	2.5	9.5	4.7
w/ Spatial- & channel-attention [4]	71.6	8.3	2.0	2.3	1.8	2.6	8.2	4.2
w/ Convolutional block attention module [33]	73.5	8.4	2.0	1.1	1.2	1.8	8.5	3.8
w/ Our combined focus	**67.9**	**7.7**	**1.3**	**0.6**	**0.9**	**1.4**	**7.3**	**3.2**

Table 4: **Focus for free across base networks** on ShanghaiTech Part_A and WIDER FACE. Base network results based on our reimplementations. Regardless of the base network, our combined focus from segmentation and global density lowers the count error.

	Part_A		**WIDER FACE**	
Network from	base	**w/** our focus	base	**w/** our focus
Zhang *et al.* [40]	114.5	**110.1**	7.1	**6.1**
Cao *et al.* [3]	75.2	**72.7**	8.5	**8.2**
Li *et al.* [14]	74.0	**72.4**	4.3	**3.9**
This paper	74.8	**67.9**	4.7	**3.2**

ferent base networks. Apart from our base network, we consider the multi-column network of Zhang *et al.* [40], the deep single column network of Li *et al.* [14] and the encoder-decoder network of Cao *et al.* [3]. We have re-implemented these networks and use the same experimental settings as in our base network. The results in Table 4 show that our focus for free lowers the count error for all these networks on ShanghaiTech Part_A and WIDER FACE.

5.4. Non-uniform kernel estimation

Next, we study the benefit of our proposed kernel for generating more reliable ground-truth density maps. For this experiment, we compare to the Geometry-Adaptive Kernel (GAK) of Zhang *et al.* [40]. For WIDER FACE, the spatial extent of objects is provided by the box annotations and we use this additional information to measure the variance quality of our kernel compared to the baseline. The counting and variance results are shown in Table 5. The proposed kernel has a lower count error than the commonly used GAK on both ShanghaiTech Part_A and WIDER FACE. To show that this improvement is due to the better estimation of the object size of interest, we compare the estimated variances σ obtained by different methods with the ground truth variance obtained by leveraging the box annotations of WIDER FACE. Our kernel reduces the MAE of σ from 2.6 to 2.2 compared to GAK.

5.5. Comparison to the state-of-the-art

Global count comparison. Table 6 shows the proposed approach outperforms all other models in terms of MAE

Table 5: **Benefit of our kernel** on ShanghaiTech Part_A and WIDER FACE. A network with our kernel obtains lower count error than with GAK [40] (see MAE (n) columns). To show this improvement is due to better object size estimation, we compare our kernel to the ground-truth on WIDER FACE, see MAE (σ) column, which indicates a lower size error than with GAK.

	Part_A	**WIDER FACE**	
Kernel from	MAE (n)	MAE (n)	MAE (σ)
GAK [40]	67.9	4.2	2.6
This paper	**65.2**	3.6	**2.2**
Ground-truth	n.a.	**3.2**	n.a.

on all six datasets. The proposed method achieves a new state of the art on ShanghaiTech Part_B, and a competitive result on ShanghaiTech Part_A in terms of RMSE. Shen *et al.* [27] achieve the lowest RMSE on ShanghaiTech Part_A, but their approach is not competitive on Part_B. Moreover, they rely on four networks with a total of 4.8 million parameters, while our proposal just needs a single network with 2.6 million parameters. On TRANCOS our method reduces the count error from 3.6 (by Issam *et al.* [11] and Li *et al.* [14]) to 2.0. A considerable reduction. For the DCC dataset proposed by Marsden *et al.* [22], we predict a more accurate global count without any post-processing, reducing the error rate from 8.4 to 3.2. On UCF-QNRF we achieve a much better MAE and RMSE than Idrees *et al.* [10]. For WIDER FACE, we evaluate using MAE and a normalized variant (NMAE). For NMAE, we normalize the MAE of each test image by the ground-truth face count. Again, our method achieves best results on both MAE and NMAE compared to the existing methods.

Local count comparison. Figure 3 shows the results obtained by various methods in terms of the commonly used GAME metric on TRANCOS. The higher the GAME value, the more counting methods are penalized for local count errors. For all GAME settings, our method sets a new state-of-the-art. Furthermore, the difference to other methods increases as the GAME value increases, indicating our method localizes and counts extremely overlapping vehicles more accurately compared to alternatives.

Density map quality. To demonstrate that our method

Table 6: **Comparison to the state-of-the-art for global count error** on ShanghaiTech Part_A, Part_B, TRANCOS, DCC, UCF-QNRF and WIDER FACE. Results on WIDER FACE based on our reimplementations. Results by Zhang *et al.* on UCF-QNRF taken from Idrees *et al.* Our results set a new state-of-the-art on all six datasets for almost all metrics.

	Part_A				**Part_B**		**TRANCOS**	**DCC**	**UCF-QNRF**		**WIDER FACE**	
	MAE	RMSE	PSNR	SSIM	MAE	RMSE	MAE	MAE	MAE	RMSE	MAE	NMAE
Zhang *et al.* [40]	110.2	173.2	21.4	0.52	26.4	41.3	-	-	277.0	426.0	7.1	1.10
Marsden *et al.* [22]	85.7	131.1	-	-	17.7	28.6	9.7	8.4	-	-	-	-
Shen *et al.* [27]	75.7	**102.7**	-	-	17.2	27.4	-	-	-	-	-	-
Li *et al.* [14]	68.2	115.0	23.8	0.76	10.6	16.0	3.6	-	-	-	4.3	0.53
Cao *et al.* [3]	67.0	104.5	-	-	8.4	13.6	-	-	-	-	8.5	1.10
Idrees *et al.* [10]	-	-	-	-	-	-	-	-	132.0	191.0	-	-
This paper	**65.2**	109.4	**25.4**	**0.78**	**7.2**	**12.2**	**2.0**	**3.2**	**93.8**	**146.5**	**3.2**	**0.40**

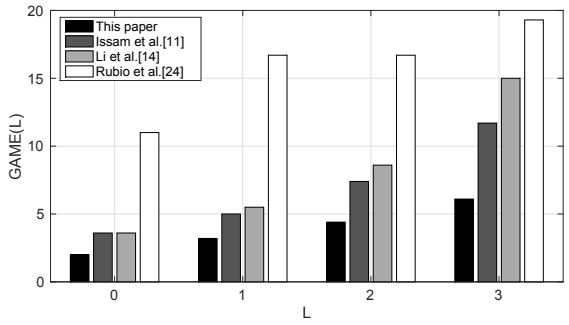

Figure 3: **Comparison to the state-of-the-art for local count error** on vehicles from TRANCOS. Note the difference to other methods increases as the GAME value grows, indicating our method localizes and counts extremely overlapping vehicles more accurately.

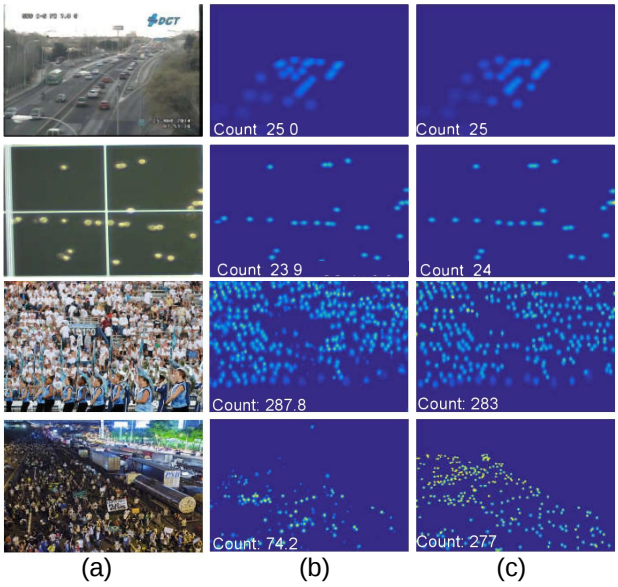

Figure 4: **Density map quality.** (a) Sample images, (b) predicted density map, and (c) the ground truth. When objects are individually visible, we can count them accurately. Further improvements are required for dense settings where objects are hard to distinguish.

also generates better quality density maps, we provide results on ShanghaiTech Part_A for the PSNR and SSIM metrics. In agreement with the results in MAE and RMSE, our method also achieves a better performance along this dimension. Compared to methods such as [14], which produces a density map with a reduced resolution and recovers the resolution by bilinear interpolation, our method directly learns the full resolution density maps with higher quality.

Success and failure cases. Finally, we show some success and failure results in Figure 4. Even in challenging scenes with relatively sparse small objects or relatively dense large objects, our method is able to achieve an accurate count (first three 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 row). Such scenarios remain open challenges.

6. Conclusion

This paper introduces two ways to repurpose the point annotations used as supervision for density-based counting. Focus from segmentation guides the counting network to focus on areas of interest, and focus from global density regularizes the counting network to learn a matching global density. Our focus for free aids density estimation from a local and global perspective, complementing each other. This paper also introduces a non-uniform kernel estimator. Experiments show the benefits of our proposal across object scales, crowding levels and base networks, resulting in state-of-the-art counting results on five benchmark datasets. The gap towards perfect counting and our qualitative analysis shows that counting in extremely dense scenes remains an open problem. Further boosts are possible when counting is able to deal with this extreme dense scenario.

References

[1] Deepak Babu Sam, Neeraj N. Sajjan, R. Venkatesh Babu, and Mukundhan Srinivasan. Divide and grow: Capturing huge diversity in crowd images with incrementally growing cnn. In *CVPR*, 2018. 2

[2] Vasileios Belagiannis, Christian Rupprecht, Gustavo Carneiro, and Nassir Navab. Robust optimization for deep regression. In *ICCV*, 2015. 5

[3] Xinkun Cao, Zhipeng Wang, Yanyun Zhao, and Fei Su. Scale aggregation network for accurate and efficient crowd counting. In *ECCV*, 2018. 1, 2, 4, 7, 8

[4] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *CVPR*, 2017. 2, 6, 7

[5] Yang Gao, Oscar Beijbom, Ning Zhang, and Trevor Darrell. Compact bilinear pooling. In *CVPR*, 2016. 3

[6] Rohit Girdhar and Deva Ramanan. Attentional pooling for action recognition. In *NeurIPS*, 2017. 2

[7] Ricardo Guerrero-Gómez-Olmedo, Beatriz Torre-Jiménez, Roberto López-Sastre, Saturnino Maldonado-Bascón, and Daniel Onoro-Rubio. Extremely overlapping vehicle counting. In *IbPRIA*, 2015. 1, 5

[8] Mohammad Hossain, Mehrdad Hosseinzadeh, Omit Chanda, and Yang Wang. Crowd counting using scale-aware attention networks. In *WACV*, 2019. 2

[9] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018. 1, 6

[10] Haroon Idrees, Muhmmad Tayyab, Kishan Athrey, Dong Zhang, Somaya Al-Maadeed, Nasir Rajpoot, and Mubarak Shah. Composition loss for counting, density map estimation and localization in dense crowds. In *ECCV*, 2018. 1, 2, 5, 7, 8

[11] H. Laradji Issam, Rostamzadeh Negar, O. Pinheiro Pedro, Vazquez David, and Schmidt Mark. Where are the blobs: Counting by localization with point supervision. In *ECCV*, 2018. 1, 7

[12] Di Kang and Antoni B. Chan. Crowd counting by adaptively fusing predictions from an image pyramid. In *BMVC*, 2018. 1, 2

[13] Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. In *NeurIPS*, 2010. 1, 2

[14] Yuhong Li, Xiaofan Zhang, and Deming Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *CVPR*, 2018. 1, 2, 5, 7, 8

[15] Zhenyang Li, Kirill Gavrilyuk, Efstratios Gavves, Mihir Jain, and Cees G. M. Snoek. VideoLSTM convolves, attends and flows for action recognition. *CVIU*, 166:41–50, 2018. 2

[16] Tsung-Yi Lin, Piotr Dollár, Ross B Girshick, Kaiming He, Bharath Hariharan, and Serge J Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 4

[17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 3

[18] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *ICCV*, 2015. 3, 4

[19] Jiang Liu, Chenqiang Gao, Deyu Meng, and Alexander G. Hauptmann. Decidenet: Counting varying density crowds through attention guided detection and density estimation. In *CVPR*, 2018. 1, 2

[20] Lingbo Liu, Hongjun Wang, Guanbin Li, Wanli Ouyang, and Liang Lin. Crowd counting using deep recurrent spatial-aware network. In *IJCAI*, 2018. 2

[21] Xialei Liu, Joost van de Weijer, and Andrew D. Bagdanov. Leveraging unlabeled data for crowd counting by learning to rank. In *CVPR*, 2018. 2

[22] Mark Marsden, Kevin McGuinness, Suzanne Little, Ciara E. Keogh, and Noel E. O'Connor. People, penguins and petri dishes: Adapting object counting models to new visual domains and object types without forgetting. In *CVPR*, 2018. 1, 5, 7, 8

[23] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 1(10), 2016. 5

[24] Daniel Onoro-Rubio and Roberto J López-Sastre. Towards perspective-free object counting with deep learning. In *ECCV*, 2016. 2, 4

[25] Viresh Ranjan, Hieu Le, and Minh Hoai. Iterative crowd counting. In *ECCV*, 2018. 2, 4

[26] Deepak Babu Sam, Shiv Surya, and R Venkatesh Babu. Switching convolutional neural network for crowd counting. In *CVPR*, 2017. 2

[27] Zan Shen, Yi Xu, Bingbing Ni, Minsi Wang, Jianguo Hu, and Xiaokang Yang. Crowd counting via adversarial cross-scale consistency pursuit. In *CVPR*, 2018. 2, 4, 7, 8

[28] Zenglin Shi, Le Zhang, Yun Liu, Xiaofeng Cao, Yangdong Ye, Ming-Ming Cheng, and Guoyan Zheng. Crowd counting with deep negative correlation learning. In *CVPR*, 2018. 2, 5

[29] Zenglin Shi, Le Zhang, Yibo Sun, and Yangdong Ye. Multi-scale multitask deep netvlad for crowd counting. *IEEE TII*, 14(11):4953–4962, 2018. 2

[30] Vishwanath A Sindagi and Vishal M Patel. Generating high-quality crowd density maps using contextual pyramid cnns. In *ICCV*, 2017. 2, 4

[31] Panqu Wang, Pengfei Chen, Ye Yuan, Ding Liu, Zehua Huang, Xiaodi Hou, and Garrison Cottrell. Understanding convolution for semantic segmentation. In *WACV*, 2018. 4, 5

[32] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. 6

[33] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and So Kweon. Cbam: Convolutional block attention module. In *ECCV*, 2018. 1, 2, 6, 7

[34] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 2

[35] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *CVPR*, 2016. 1, 2, 5

[36] Fisher Yu, Vladlen Koltun, and Thomas A Funkhouser. Dilated residual networks. In *CVPR*, 2017. 4

[37] Cong Zhang, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. Cross-scene crowd counting via deep convolutional neural networks. In *CVPR*, 2015. 2, 5

[38] Shanghang Zhang, Guanhang Wu, Joao P. Costeira, and Jose M. F. Moura. Understanding traffic density from large-scale web camera data. In *CVPR*, 2017. 4

[39] Shanshan Zhang, Jian Yang, and Bernt Schiele. Occluded pedestrian detection through guided attention in cnns. In *CVPR*, 2018. 2

[40] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *CVPR*, 2016. 1, 2, 4, 5, 7, 8

[41] Zheng Zhu, Wei Wu, Wei Zou, and Junjie Yan. End-to-end flow correlation tracking with spatial-temporal attention. In *CVPR*, 2018. 2