From Open Set to Closed Set: Counting Objects by Spatial Divide-and-Conquer*

Haipeng Xiong†, Hao Lu‡, Chengxin Liu†, Liang Liu†, Zhiguo Cao†, Chunhua Shen‡
†Huazhong University of Science and Technology, China
‡The University of Adelaide, Australia
{hpxiong,zgcao}@hust.edu.cn, hao.lu@adelaide.edu.au

Abstract

Visual counting, a task that predicts the number of objects from an image/video, is an open-set problem by nature, i.e., the number of population can vary in $[0, +\infty)$ in theory. However, the collected images and labeled count values are limited in reality, which means only a small closed set is observed. Existing methods typically model this task in a regression manner, while they are likely to suffer from an unseen scene with counts out of the scope of the closed set. In fact, counting is decomposable. A dense region can always be divided until sub-region counts are within the previously observed closed set. Inspired by this idea, we propose a simple but effective approach, Spatial Divide-and-Conquer Network (S-DCNet). S-DCNet only learns from a closed set but can generalize well to open-set scenarios via S-DC. S-DCNet is also efficient. To avoid repeatedly computing sub-region convolutional features, S-DC is executed on the feature map instead of on the input image. S-DCNet achieves the state-of-the-art performance on three crowd counting datasets (ShanghaiTech, UCF_CC_50 and UCF-QNRF), a vehicle counting dataset (TRANCOS) and a plant counting dataset (MTC). Compared to the previous best methods, S-DCNet brings a 20.2% relative improvement on the ShanghaiTech Part_B, 20.9% on the UCF-QNRF, 22.5% on the TRANCOS and 15.1% on the MTC. Code has been made available at: https://github.com/xhp-hust-2018-2011/S-DCNet.

1. Introduction

The task of visual counting in Computer Vision is to infer the number of objects (people, cars, maize tassels, etc.) from an image/video. It has wide applications, such as automatic crowd management [15, 16, 17, 37, 38], traffic monitoring [14, 25], and crop yield estimation [10, 13, 23]. Extensive attention has been received in recent years.

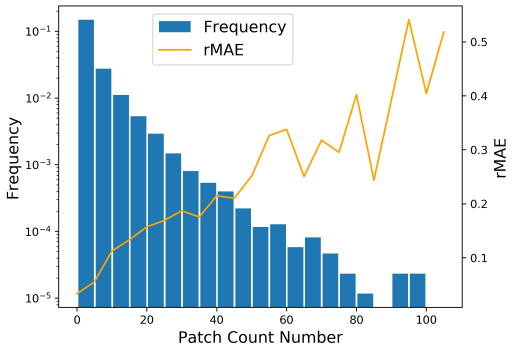

Figure 1. The histogram of count values of 64×64 local patches on the test set of ShanghaiTech Part_A dataset [38]. The orange curve denotes the relative mean absolute error (rMAE) of CSRNet [20] on local patches.

Counting is an open-set problem by nature as a count value can range from 0 to $+\infty$ in theory. It is thus typically modeled in a regression manner. Benefiting from the success of convolutional neural networks (CNNs), state-of-the-art deep counting networks often adopt a multi-branch architecture to enhance the feature robustness to dense regions [2, 4, 38]. However, the observed patterns in datasets are limited in practice, which means networks can only learn from a closed set. *Are these counting networks still able to generate accurate predictions when the number of objects is out of the scope of the closed set?* Meanwhile, observed local counts exhibit a long-tailed distribution shown in Figure 1. Extremely dense patches are rare while sparse patches take up the majority. As what can be observed, the relative mean absolute error (rMAE) increases significantly with increased local density. *Is it necessary to set the working range of CNN-based regressors to the maximum count value observed, even with a majority of samples are sparse such that the regressor works poorly in this range?*

In fact, counting has an unique property—spatially decomposable. The above problem can be largely alleviated with the idea of spatial divide-and-conquer (S-DC). Suppose that a network has been trained to accurately predict a closed set of counts, say $0 \sim 20$. When facing an image with extremely dense objects, one can keep dividing the

*Haipeng Xiong and Hao Lu contributed equally. Zhiguo Cao is the corresponding author.

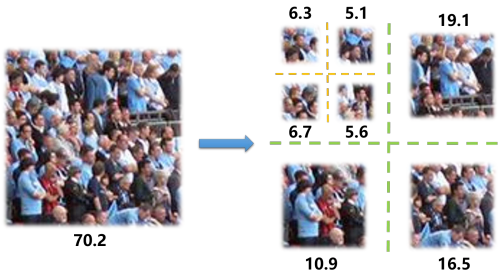

Figure 2. An illustration of spatial divisions. Suppose that the closed set of counts is $[0, 20]$. In this example, dividing the image for one time is inadequate to ensure that all sub-region counts are within the closed set. For the top left sub-region, it needs a further division.

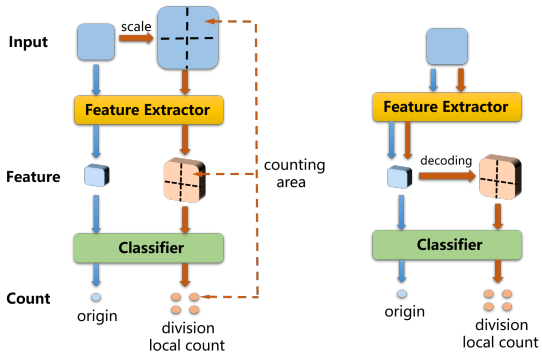

Figure 3. Spatial divisions on the input image (left) and the feature map (right). Spatially dividing the input image is straightforward. The image is upsampled and fed to the same network to infer counts of local areas. The orange dashed line is used to connect the local feature map, the local count and the sub-image. S-DC on the feature map avoids redundant computations and is achieved by upsampling, decoding and dividing the feature map of high resolution.

image into sub-images until all sub-region counts are less than 20. Then the network can accurately count these sub-images and sum over all local counts to obtain the global image count. Figure 2 graphically depicts the idea of S-DC. A follow-up question is how to spatially divide the count. A naive way is to upsample the input image, divide it into sub-images and process sub-images with the same network. This way, however, is likely to blur the image and lead to exponentially-increased computation cost and memory consumption when repeatably extracting the feature map. Inspired by RoI pooling [12], we show that it is feasible to achieve S-DC on the feature map, as conceptually illustrated in Figure 3. By decoding and upsampling the feature map, the later prediction layers can focus on the feature of local areas and predict sub-region counts accordingly.

To realize the above idea, we propose a simple but effective Spatial Divide-and-Conquer Network (S-DCNet). S-DCNet learns from a closed set of count values but is able to generalize to open-set scenarios. Specifically, S-DCNet adopts a VGG16 [30]-based encoder and an UNet [27]-like decoder to generate multi-resolution feature maps. All feature maps share the same counting predictor. Inspired by [19], in contrast to the conventional density map regression, we discretize continuous count values into a set of intervals and design the counting predictor to be a classifier. Further, a division decider is designed to decide which sub-region should be divided and to merge different levels of sub-region counts into the global image count. We show through a controlled toy experiment that, even given a closed training set, S-DCNet effectively generalizes to the open test set. The effectiveness of S-DCNet is further demonstrated on three crowd counting datasets (ShanghaiTech [38], UCF_CC_50 [15] and UCF-QNRF [16]), a vehicle counting dataset (TRANCOS [14]), and a plant counting dataset (MTC [23]). Results show that S-DCNet indicates a clear advantage over other competitors and sets the new state-of-the-art across five datasets.

The main contribution of this work is that we propose to transform open-set counting into a closed-set problem. We show through extensive experiments that a model learned in a closed set can effectively generalize to the open set with the idea of S-DC.

2. Related Work

Current CNN-based counting approaches are mainly built upon the framework of local regression. According to their regression targets, they can be categorized into two categories: density map regression and local count regression. We first review these two types of regression. Since S-DCNet learns to classify counts, some works that reformulate the regression problem are also discussed.

Density Map Regression The concept of density map was introduced in [18]. The density map contains the spatial distribution of objects, thus can be smoothly regressed. Zhang *et al*. [37] first adopted a CNN to regress local density maps. Then almost all subsequent counting networks followed this idea. Among them, a typical network architecture is multi-branch. MCNN [38] and Switching-CNN [2] used three columns of CNNs with varying receptive fields to depict objects of different scales. SANet [4] adopted Inception [34]-liked modules to integrate extra branches. CP-CNN [32] added two extra density-level prediction branches to combine global and local contextual information. AC-SCP [28] inserted a child branch to match cross-scale consistency and an adversarial branch to attenuate the blurring effect of the density map. ic-CNN [26] incorporated two branches to generate high-quality density maps in a coarse-to-fine manner. IG-CNN [1] and D-ConvNet [29] drew inspirations from ensemble learning and trained a series of networks or regressors to tackle different scenes. DecideNet [21] attempted to selectively fuse the results of

density map estimation and object detection for different scenes. Unlike multi-branch approaches, Idrees *et al*. [16] employed a composition loss and simultaneously solved several counting-related tasks to assist counting. CSR-Net [20] benefited from dilated convolution which effectively expanded the receptive field to capture contextual information.

Existing deep counting networks aim to generate high-quality density maps. However, density maps are actually in the open set as well. Detailed discussion of the open set problem in density maps is provided in the Supplement.

Local Count Regression Local count regression directly predicts count values of local image patches. This idea first appeared in [7] where a multi-output regression model was used to regress region-wise local counts simultaneously. [9] and [23] introduced such an idea into deep counting. Local patches were first densely sampled in a sliding-window manner with overlaps, and a local count was then assigned to each patch by the network. Inferred redundant local counts were finally normalized and fused to the global count. Stahl *et al*. [33] regressed the counts for object proposals generated by Selective Search [36] and combined local counts using an inclusion-exclusion principle. Inspired by subitizing, the ability for a human to quickly counting a few objects at a glance, Chattopadhyay *et al*. [5] transferred their focus to the problem of counting objects in everyday scenes. The main challenge thus became large intra-class variances rather than the occlusions and perspective distortions in crowded scenes.

While some above methods [5, 33] also leverage the idea of spatial divisions, they still regress the open-set counts. Although local-region patterns are easier to be modelled than the whole image, the observed local patches are still limited. Since only finite local patterns (a closed set) can be observed, new scenes in reality have a high probability including objects out of the range (an open set). Moreover, dense regions with large count values are rare (Figure 1) and the networks may suffer from sample imbalance. In this paper, we show that a counting network is able to learn from a closed set with a certain range of counts, say $0 \sim 20$, and then generalizes to an open set (including counts > 20) via S-DC.

Beyond Naive Regression Regression is a natural way to estimating continuous variables, such as age and depth. However, some literatures suggest that regression is encouraged to be reformulated as an ordinal regression problem or a classification problem, which enhances performance and benefits optimization [6, 11, 19, 24]. Ordinal regression is usually implemented by modifying well-studied classification algorithms and has been applied to the problem of age estimation [24] and monocular depth prediction [11]. Li *et al*. [19] further showed that directly reformulating regression to classification was also a good choice. Since count values share a similar property like age and depth, it motivates us to follow such a reformulation. In this work, S-DCNet follows [19] to discretize local counts and classify count intervals. Indeed, we observe in experiments that classification with S-DC works better than direct regression.

classifier	division decider
2×2 AvgPool, s 2	2×2 AvgPool, s 2
1×1 Conv, 512, s 1	1×1 Conv, 512, s 1
1×1 Conv, $class\ num$, s 1	1×1 Conv, 1, s 1
–	Sigmoid

Table 1. The architecture of *classifier* and *division decider*. $AvgPool$ denotes average pooling. Convolutional layers are defined in the format: $Conv\ size \times size$, $output\ channel$, $s\ stride$. Each convolutional layer is followed by a ReLU function except the last layer. In particular, a *sigmoid* function is employed at the end of *division decider* to generate soft division masks.

3. Spatial Divide-and-Conquer Network

In this section, we describe the transformation from quantity to interval which transfers count values into a closed set. We also explain in detail our proposed S-DCNet.

3.1. From Quantity to Interval

Instead of regressing an open set of count values, we follow [19] to discretize local counts and classify count intervals. Specifically, we define an interval partition of $[0, +\infty)$ as $\{0\}, (0, C_1], (C_2, C_3], ..., (C_{M-1}, C_M]$ and $(C_M, +\infty)$. These $M+1$ sub-intervals are labeled to the 0-th to the M-th classes, respectively. For example, if a count value is within $(C_2, C_3]$, it is labeled as the 2-th class. In practice, C_M should be not greater than the max local count observed in the training set.

The median of each sub-interval is adopted when recovering the count from the interval. Notice that, for the last sub-interval $(C_M, +\infty]$, C_M will be used as the count value if a region is classified into this interval. It is clear that *adopting C_M for the last class will cause a systematic error*, but the error can be mitigated via S-DC as what we will show in experiments.

3.2. Single-Stage Spatial Divide-and-Conquer

As shown in Figure 4, S-DCNet includes a VGG16 [30] feature encoder, an UNet [27]-like decoder, a count-interval classifier and a division decider. The structure of the classifier and the division decider are shown in Table 1. Notice that, the first average pooling layer in the classifier has a stride of 2, so the final prediction has an output stride of 64.

The feature encoder removes fully-connected layers from the pre-trained VGG16. Suppose that the input patch is of size 64×64. Given the feature map F_0 (extracted

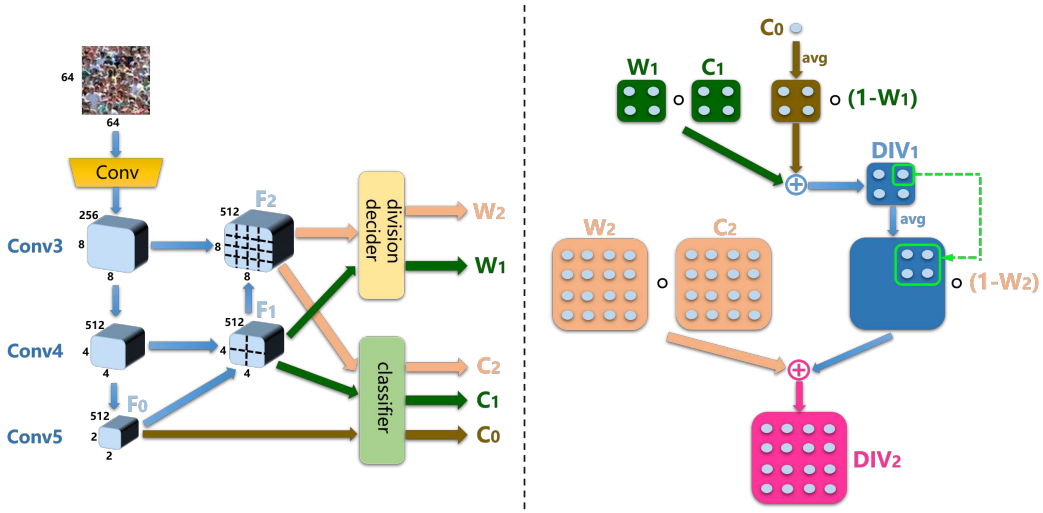

Figure 4. The architecture of S-DCNet (left) and a two-stage S-DC process (right). S-DCNet adopts all convolutional layers in VGG16 [30] while the first two convolutional blocks are simplified as $Conv$ in the figure. An UNet [27]-like decoder is employed to upsample and divide the feature map as per Figure 3. A shared classifier and a division decider receive divided feature maps, and respectively, generate division counts C_is and division masks W_is, for $i = 1, 2, ...$ After obtaining these results, C_i and W_i are merged to the i-th division count DIV_i shown in the right sub-figure. Specially, we average each count of low resolution into the corresponding 2×2 area of high resolution before merging (avg shown in the figure). "$\circ$" denotes the Hadamard product. Note that, the 64×64 local patch is only used as an example for readers to understand the pipeline of S-DCNet. Since S-DCNet is a fully convolutional network, it can process images of arbitrary sizes $M \times N$ and return DIV_2s of size $\frac{M}{64} \times \frac{N}{64}$. The structures for the classifier and the division decider are presented in Table 1.

from the Conv5 layer) with $\frac{1}{32}$ resolution of the input image, the classifier predicts the class label of the count interval CLS_0 conditioned on F_0. The local count C_0, which denotes the count value of the 64×64 input patch, can be recovered from CLS_0. Note that C_0 is the local count without S-DC, which is also the final output of previous approaches [5, 9, 23].

We execute the first-stage S-DC on the fused feature map F_1. F_1 is divided and sent to the shared classifier to produce the division count $C_1 \in \mathbb{R}^{2\times2}$. Concretely, F_0 is upsampled by $\times 2$ in an UNet-like manner to F_1. Given F_1, the classifier fetches the local features that correspond to spatially divided sub-regions, and predicts the first-level division counts C_1. Each of the 2×2 elements in C_1 denotes a sub-count of the corresponding 32×32 sub-region.

With local counts C_0 and C_1, the next question is to decide where to divide. We learn such decisions with another network module, division decider, as depicted in the right part of Figure 4. At the first stage of S-DC, the division decider generates a soft division mask W_1 of the same size as C_1 conditioned on F_1 such that for any $w \in W_1, w \in [0, 1]$. $w = 0$ means no division is required at this position, and the value in C_0 is used. $w = 1$ implies that here the initial prediction should be replaced with the division count in C_1. Since W_1 and C_1 are both 2 times larger than C_0, C_0 is upsampled by $\times 2$ to $\hat{C}_0$, and the count is averaged into the 2×2 local area in $\hat{C}_0$. The first-stage division result DIV_1 can thus be computed as

$$DIV_1 = (\mathbb{1} - W_1) \circ avg(C_0) + W_1 \circ C_1 \,, \qquad (1)$$

where $\mathbb{1}$ denotes a matrix filled with 1 and is with the same size of W_1. "$\circ$" denotes the Hadamard product. avg is an averaging re-distribution operator (equally dividing a count value into a 2×2 region).

Algorithm 1: Multi-Stage S-DC

Input: Image I and division time N
Output: Image count C

Extract F_0 from I;
Generate CLS_0 given F_0 with the classifier, and recover C_0 from CLS_0;
Initialize $DIV_0 = C_0$;
for $i \leftarrow 1$ ***to*** N **do**
  Decode F_{i-1} to F_i;
  Process F_i with the classifier and the division decider to obtain CLS_i and the division mask W_i;
  Recover C_i from CLS_i;
  Update DIV_i as per Eq. 2 ;
Integrate over DIV_N to obtain the image count C;
return C

3.3. Multi-Stage Spatial Divide-and-Conquer

S-DCNet can execute multi-stage S-DC by further decoding, dividing the feature map until reaching the output of the first convolutional block. In this sense, the maximum

division time is 4 in VGG16 (actually we show later in experiments that a two-stage division is adequate to guarantee satisfactory performance). In multi-stage S-DC, DIV_i ($i \geq 2$) is merged in a recursive manner as

$$DIV_i = (\mathbb{1} - W_i) \circ avg(DIV_{i-1}) + W_i \circ C_i \,. \quad (2)$$

We employ two types of standard loss functions to train S-DCNet: several cross-entropy losses L_C^is that correspond to different classification outputs CLS_is, and a ℓ_1 loss L_R^N for the final division output DIV_N (N denotes the division time). S-DCNet is learned in a multi-task manner where the overall loss L is a summation of all losses, i.e., $L = \sum_{i=0}^{N} L_C^i + L_R^N$. Note that, L_R^N is essential to provide an implicit supervision signal for learning W_is. Multi-stage S-DCNet is summarized in Algorithm 1.

4. Open Set or Closed Set? A Toy-Level Justification

As aforementioned, counting is an open-set problem while the model is learned in a closed set. *Can a closed-set counting model really generalize to open-set scenarios?* Here we show through a controlled toy experiment that, the answer is *no*. Inspired by [18], we synthesize a cell counting dataset to explore the counting performance outside a closed training set.

Synthesized Cell Counting Dataset We first generate 500 256×256 images with 64×64 sub-regions containing only $0 \sim 10$ cells to construct the training set (a closed set). To generate an open testing set, we further synthesize 500 images with sub-region counts evenly distributed in the range of $[0, 20]$.

Baselines and Protocols We adopt three approaches for comparisons, they are: **i**) a regression baseline with pre-trained VGG16 as the backbone and the *classifier* module used in S-DCNet as the backend except that the output channel is modified to 1. ℓ_1 loss is used. This approach directly regresses the open-set counts; **ii**) a classification baseline with the same VGG16 and the *classifier* settings as S-DCNet, without S-DC; **iii**) our proposed S-DCNet, which learns from a closed set but adapts to the open set via S-DC.

Regarding the discretization of count intervals, we choose 0.5 as the step because cells can be partially presented in local patches. As a consequence, we have a partition of $\{0\}$, $(0.0.5]$,$(0.5, 1]$, ... ,$(9.5, 10]$ and $(10, +\infty)$. All approaches are trained with standard stochastic gradient descent (SGD). The learning rate is initially set to 0.001 and is decreased by $\times 10$ when the training error stagnates.

Observations According to Figure 5, it can be observed that both regression and classification baselines work well in the range of the closed set ($0 \sim 10$), but the counting error increases rapidly when counts are larger than 10.

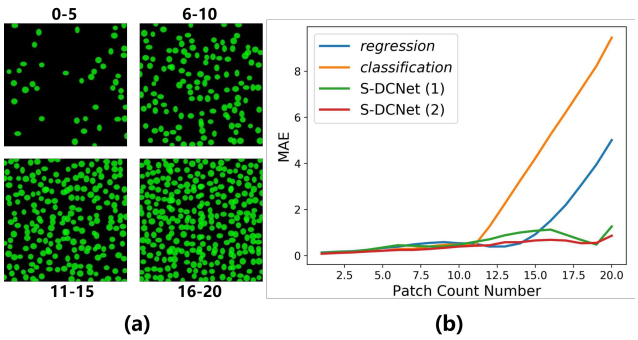

Figure 5. A toy-level justification. **(a)** Some 256×256 images in the simulated cell counting dataset. The numbers denote the range of local counts of 64×64 sub-regions. **(b)** The mean absolute error (MAE) of different methods versus 64×64 sub-region counts. S-DCNet(N) means N-stage S-DCNet.

This suggests a conventional counting model learned in a closed set cannot generalize to the open set. However, S-DCNet can achieve accurate predictions even on the open set, which confirms the advantage of S-DC.

5. Experiments on Real-World Datasets

Extensive experiments are further conducted to demonstrate the effectiveness of S-DCNet on real-world datasets. We first describe some essential implementation details. After that, an ablation study is conducted on the ShanghaiTech Part_A [38] dataset to highlight the benefit of S-DC. Finally, we compare S-DCNet against current state-of-the-art methods on five public datasets. Mean Absolute Error (MAE) and Root Mean Squared Error (MSE) are used as the evaluation metrics following [38].

5.1. Implementation Details

Interval Partition We generate ground-truth counts of local patches by integrating over the density maps. The counts are usually not integers, because objects can partly present in cropped local patches. We evaluate two different partition strategies. In the first partition, we choose 0.5 as the step and generate partitions as $\{0\}$, $(0.0.5]$,$(0.5, 1]$, ... ,$(C_{max} - 0.5, C_{max}]$ and $(C_{max}, +\infty)$, where C_{max} denotes the maximum count of the closed set. This partition is named as `One-Linear Partition`.

In the second partition, we further finely divide the sub-interval $(0.0.5]$, because this interval contains a sudden change from no object to part of an object, and a large proportion of objects lie in this sub-interval. A small step of 0.05 is used to divide this interval. We call this partition `Two-Linear Partition`.

Data Augmentation We follow the same data augmentation used in [20], except for the UCF-QNRF dataset [16]. In particular, 9 sub-images of $\frac{1}{4}$ resolution are cropped from

the original image. The first 4 sub-images are from four corners, and the remaining 5 are randomly cropped. Random scaling and mirroring are also performed. For the UCF-QNRF dataset [16], we follow the same setting as in [16] and crop the original image into 224×224 sub-images.

Training Details S-DCNet is implemented with `PyTorch`. We train S-DCNet using standard SGD. The encoder in S-DCNet is directly adopted from convolutional layers of VGG16 [30] pretrained on ImageNet, and the other layers employ random Gaussian initialization with a standard deviation of 0.01. The learning rate is initially set to 0.001 and is decreased by $\times 10$ when the training error stagnates. We keep training until convergence. For the ShanghaiTech, UCF_CC_50, TRANCOS and MTC datasets, the batch size is set to 1. For the UCF-QNRF dataset, the batch size is set to 16 following [16].

5.2. Ablation Study on the ShanghaiTech Part_A

Is S-DCNet Robust to C_{max}? When reformulating the counting problem into classification, a critical issue is how to choose C_{max}, which defines the closed set. Hence, it is important that S-DCNet is robust to the choice of C_{max}.

We conduct a statistical analysis on count values of local patches in the training set, and then set C_{max} with the quantiles ranging from 100% to 80% (decreased by 5%). Two-stage S-DCNet is evaluated. Another baseline of classification without S-DC is also used to explore whether counting can be simply modeled in a closed-set classification manner. To be specific, we reserve the VGG16 encoder and the classifier in this classification baseline.

Results are presented in Figure 6. It can be observed that the MAE of the classification baseline increases rapidly with decreased C_{max}. This result is not surprising, because the model is constrained to be visible to count values not greater than C_{max}. This suggests that counting cannot be simply transformed into closed-set classification. However, with the help of S-DC, S-DCNet exhibits strong robustness to the changes of C_{max}. It seems the systematic error brought by C_{max} can somewhat be alleviated with S-DC. Regarding how to choose concrete C_{max}, the maximum count of the training set seems not the best choice, while some small quantiles even deliver better performance. Perhaps a model is only able to count objects accurately within a certain degree of denseness. We also notice `Two-Linear Partition` is slightly better than `One-Linear Partition`, which indicates that the fine division to the $(0, 0.5]$ sub-interval has a positive effect.

According to the above results, S-DCNet is robust to C_{max} in a wide range of values, and C_{max} is generally encouraged to be set less than the maximum count value observed. In addition, there is no significant difference between two kinds of partitions. For simplicity, we set

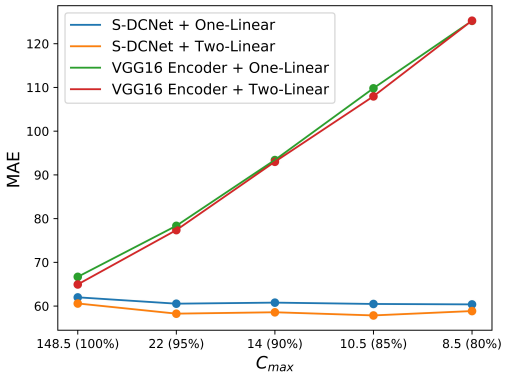

Figure 6. The influence of C_{max} to S-DCNet on the ShanghaiTech Part_A dataset [38]. The numbers in the brackets denote quantiles of the training set, for example, 22 (95%) means the 95% quantile is 22. 'VGG16 Encoder' is the classification baseline without S-DC. 'One-Linear' and 'Two-Linear' are defined in Section 5.1.

C_{max} to be the 95% quantile and adopt `Two-Linear Partition` in the following experiments.

How Many Times to Divide? S-DCNet can apply S-DC up to 4 times, but how many times are sufficient? Here we evaluate S-DCNet with different division stages. Quantitative results are listed in Table 2. It can be observed that applying two-stage S-DC is clearly adequate.

The Effect of S-DC To highlight the effect of S-DC, we compare S-DCNet against several regression and classification baselines. These baselines adopt the same architecture of VGG16 encoder and the classifier in S-DCNet. *classification* is the result of C_0 without S-DC, and C_{max} is set to be the 95% quantile (22). For all regression baselines, we modify the output channel of the classifier to be 1 and employ the ℓ_1 loss. We set three regression baselines. *regression* predicts counts without S-DC. To justify whether S-DC can also work in regression, we adapt the S-DC idea to regression under both open-set and closed-set settings. *open-set regression + S-DC* is straight-forward. We do not limit the output range, and it can vary from 0 to $+\infty$. *closed-set regression + S-DC* indicates that the output range is constrained within $[0, C_{max}]$ (C_{max} is set to 22 for a fair comparison), and large outputs will be clipped to C_{max}.

Results are shown in Table 3. We can see that counting by classification without S-DC suffers from the limitation of C_{max} and performs even worse than regression. In addition, regression can also benefit from S-DC, and it is encouraged to limit the output range of the regressor in a closed set. Moreover, with S-DC, S-DCNet significantly reduces the counting error and outperforms both the classification and regression baselines by a large margin. This verifies our argument that it is more effective to reformulate counting in classification than in regression. Perhaps the optimization is easier and less sensitive to sample imbalance in classification than in regression. Whatever, at least one thing is

Division time	MAE	MSE
0	76.0	142.5
1	62.2	103.4
2	**58.3**	**95.0**
3	60.1	99.8
4	61.9	107.2

Table 2. Results of S-DCNet with different S-DC stages. The best performance is boldfaced.

Method	MAE	MSE
classification	77.4	149.3
regression	68.9	112.1
open-set regression + S-DC	66.6	107.9
closed-set regression + S-DC	64.7	105.7
S-DCNet (2)	**58.3**	**95.0**

Table 3. Effect of S-DC. Two classification and regression baselines are compared against S-DCNet. S-DCNet (2) denotes two-stage S-DCNet. The best performance is boldfaced.

$\sum_{i=0}^{2} L_C^i$	L_R^2	MAE	MSE
	✓	301.4	396.9
✓		88.4	128.8
✓	✓	**58.3**	**95.0**

Table 4. Effect of different loss functions. Note that, multi-stage predictions are averaged if L_R^2 is not applied, because the division decider cannot receive supervision signal during training. The best performance is boldfaced.

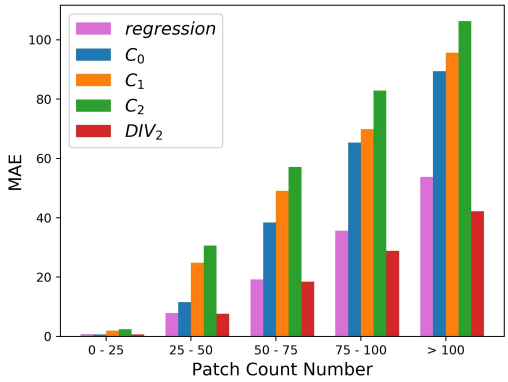

Figure 7. Counting errors of 64×64 local patches on the test set of ShanghaiTech Part_A [38]. *regression* denotes direct local counts regression using VGG16. C_0, C_1 and C_2 are single-branch predictions conditioned on F_0, F_1 and F_2, respectively. DIV_2 denotes two-stage S-DCNet, which fuses the predictions of C_0, C_1 and C_2 with S-DC.

made clear: a counting model can learn from a closed set and generalize well to a open set via S-DC.

We further analyze the counting error of 64×64 local patches in detail. As shown in Figure 7, we observe that the direct single-branch prediction without S-DC (predicting C_0, C_1 and C_2 from $F0$, $F1$ and $F2$, respectively) performs worse than the regression baseline, which can be attributed to the limited C_{max} of the classifier. After embedding the S-DC strategy to divide and merge the count map of multiple resolutions, counting errors significantly reduce. Such a benefit is even much obvious in dense patches with local counts greater than 100. It justifies our argument that, instead of regressing a large count value directly, it is more accurate to count dense patches through S-DC.

Loss Functions Also Matter We further validate the effect of different loss functions used in S-DCNet and report the results in Table 4. S-DCNet works poorly when trained with only L_R^2. This is not surprising, because no supervision signal is provided to multi-stage division results. In addition, it seems necessary for the division decider to decide where to divide, because S-DCNet greatly benefits from the help of merging loss L_R^2.

Through the visualizations of W_is in Fig. 8, we observe that reasonably good divisions can be achieved with the supervision of L_R^2. This has another benefit, the network can learn when to divide not just in counts larger than C_{max}.

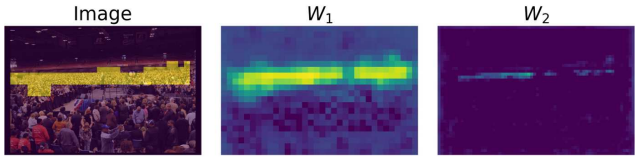

Figure 8. Visualizations of W_is in S-DCNet. The brighter the image is, the greater the values are. In the input image, count values greater than C_{max} are indicated by yellow regions. It is clear that W_i appropriately identifies regions to be divided.

Dataset	C_{max}	max	Gaussian kernel
SH Part_A [38]	22.0	148.5	Geometry-Adaptive
UCF_CC_50 [15]	–	–	
UCF-QNRF [16]	8.0	131.5	
SH Part_B [38]	7.0	83.0	Fixed: $\sigma = 15$
Trancos [14]	5.0	24.5	Fixed: $\sigma = 10$
MTC [23]	3.5	8.0	Fixed: $\sigma = 8$
Partition	Two-Linear		
Type of C_{max}	95% quantile		

Table 5. Overall configurations of S-DCNet. *max* denotes the maximum count of local patch in the training set, while C_{max} is the maximum count set for the closed set in S-DCNet. *Gaussian kernel* is used to generate density maps from dotted annotations. Specially, since UCF_CC_50 adopts 5-fold cross-validation, *max* and C_{max} are set differently for each fold.

5.3. Comparison with State of the Art

According to the ablation study, the final configurations for S-DCNet are summarized in Table 5. Qualitative results are shown in the Supplement.

The ShanghaiTech Dataset The ShanghaiTech crowd counting dataset [38] is consisted of two parts: Part_A and Part_B. Part_A includes 300 images for training and 182 for testing. This part represents highly congested scenes. Part_B contains 716 images in relatively sparse scenes, where 400 images are used for training and 316 for testing. Quantitative results are listed in Table 6. Our method outperforms the previous state-of-the-art SPN [8] and SANet [4] with a 5.5% relative improvement in Part_A and 20.2% in Part_B, respectively. These results suggest S-DCNet is able to adapt to both sparse and crowded scenes.

The UCF_CC_50 Dataset UCF_CC_50 [15] is a tiny crowd counting dataset with 50 images in extremely

	Part A		Part B	
Method	MAE	MSE	MAE	MSE
Zhang *et al.* [37]	181.8	277.7	32.0	49.8
CP-CNN [32]	73.6	106.4	20.1	30.1
D-ConvNet [29]	73.5	112.3	18.7	26.0
IG-CNN [1]	72.5	118.2	13.6	21.1
DRSAN [22]	69.3	96.4	11.1	18.2
CSRNet [20]	68.2	115.0	10.6	16.0
SANet [4]	67.0	104.5	8.4	13.6
SPN [8]	61.7	99.5	9.4	14.4
S-DCNet	**58.3**	**95.0**	**6.7**	**10.7**

Table 6. Comparison with state-of-the-art approaches on the test set of ShanghaiTech [38] dataset. The best performance is boldfaced.

Method	MAE	MSE
Idree*et al.* [15]	468.0	590.3
Zhang*et al.* [37]	467.0	498.5
IG-CNN [1]	291.4	349.4
D-ConvNet [29]	288.4	404.7
CSRNet [20]	266.1	397.5
SANet [4]	258.4	334.9
DRSAN [22]	219.2	**250.2**
S-DCNet	**204.2**	301.3

Table 7. Comparison with state-of-the-art approaches on the test set of UCF_CC_50 [15] dataset. The best performance is boldfaced.

Method	MAE	MSE
Idree*et al.* [15]	315	508
MCNN [38]	277	426
Encoder-Decoder [3]	270	478
CMTL [31]	252	514
Switching-CNN [2]	228	445
Base Network [16]	163	227
Composition Loss [16]	132	191
S-DCNet	**104.4**	**176.1**

Table 8. Comparison with state-of-the-art approaches on the test set of UCF-QNRF [16] dataset. The best performance is boldfaced.

Method	GAME(0)	GAME(1)	GAME(2)	GAME(3)
CCNN [25]	12.49	16.58	20.02	22.41
Hydra-3s [25]	10.99	13.75	16.69	19.32
CSRNet [20]	3.56	5.49	8.57	15.04
SPN [8]	3.35	4.94	6.47	9.22
S-DCNet	**2.92**	**4.29**	**5.54**	**7.05**

Table 9. Comparison with state-of-the-art approaches on the test set of TRANCOS [14] dataset. The best performance is boldfaced.

Method	MAE	MSE
GlobalReg [35]	19.7	23.3
DensityReg [18]	11.9	14.8
CCNN [25]	21.0	25.5
TasselNet [23]	6.6	9.6
S-DCNet	**5.6**	**9.1**

Table 10. Comparison with state-of-the-art approaches on the test set of MTC [23] dataset. The best performance is boldfaced.

crowded scenes. The number of people within an images varies from 96 to 4633. We follow the 5-fold cross-validation as in [15]. Results are shown in Table 7. Our method surpasses the previous best method, DRSAN [22], with a 6.8% relative improvement in *MAE*.

The UCF-QNRF Dataset UCF-QNRF [16] is a large crowd counting dataset with 1535 high-resolution images and 1.25 million head annotations. There are 1201 training images and 334 test images. It contains extremely congested scenes where the maximum count of an image can reach 12865. We follow the same image processing as in [16] and report results in Table 8. Our method reaches the state-of-the-art performance and surpasses the previous best method with a 20.9% boost in *MAE*. We surprisingly notice that S-DCNet only learn from a closed set with $C_{max} = 8.0$, which is only 6% of the maximum count 131.5 according to Table 5. S-DCNet, however, generalizes to large counts effectively and predicts accurate counts.

The TRANCOS Dataset Aside from crowd counting, we also evaluate S-DCNet on a vehicle counting dataset, TRANCOS [14], to see its generalization ability. TRANCOS contains 1244 images of congested traffic scenes in various perspectives. It adopts the Grid Average Mean Absolute Error (GAME) [14] as the evaluation metric. $GAME(L)$ divides an image into $2^L \times 2^L$ non-overlapping sub-regions and accumulates of the MAE over sub-regions. Larger L implies better local predictions. In particular, $GAME(0)$ downgrades to MAE. Results are listed in Table 9. S-DCNet surpasses other methods under all $GAME(L)$ metrics, and particularly, delivers a 22.5% relative improvement on $GAME(3)$. This suggests S-DCNet not only achieves accurate global predictions but also behaves well in local regions.

The MTC Dataset We further evaluate our method on a plant counting dataset, i.e., the MTC dataset [23]. The MTC dataset contains 361 high-resolution images of maize tassels collected from 2010 to 2015 in the wild field. In contrast to people or vehicles that have similar physical sizes, maize tassels are with heterogeneous physical sizes and are self-changing over time. We think this dataset is suitable for justifying the robustness of S-DCNet to object-size variations. We follow the same setting as in [23] and report quantitative results in Table 10. Although the previous best method, TasselNet [23], already exhibits accurate results, S-DCNet still shows a certain degree of improvement.

6. Conclusion

Counting is an open-set problem in theory, but only a finite closed set can be observed in reality. This is particularly true because any dataset is always a sampling of the real world. Inspired by the decomposable property of counting, we propose to transform the open-set counting into a closed-set problem, and address the problem with the idea of S-DC. We realize S-DC in a deep counting network termed S-DCNet. We show through a toy experiment and extensive evaluations on standard benchmarks that, even given a closed training set, S-DCNet can effectively generalize to open-set scenarios.

For future work, we will test the adaptability of S-DC on other network architectures.

Acknowledgements This work was supported by the Natural Science Foundation of China under Grant No. 61876211.

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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3618–3626, 2018. 2, 8

[2] Deepak Babu Sam, Shiv Surya, and R. Venkatesh Babu. Switching convolutional neural network for crowd counting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5744–5752, 2017. 1, 2, 8

[3] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017. 8

[4] Xinkun Cao, Zhipeng Wang, Yanyun Zhao, and Fei Su. Scale aggregation network for accurate and efficient crowd counting. In *The European Conference on Computer Vision (ECCV)*, pages 734–750, 2018. 1, 2, 7, 8

[5] Prithvijit Chattopadhyay, Ramakrishna Vedantam, Ramprasaath R. Selvaraju, Dhruv Batra, and Devi Parikh. Counting everyday objects in everyday scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1135–1144, 2017. 3, 4

[6] Ke Chen, Shaogang Gong, Tao Xiang, and Chen Change Loy. Cumulative attribute space for age and crowd density estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2467–2474, 2013. 3

[7] Ke Chen, Chen Change Loy, Shaogang Gong, and Tony Xiang. Feature mining for localised crowd counting. In *Proc. British Machine Vision Conference (BMVC)*, 2012. 3

[8] Xinya Chen, Yanrui Bin, Nong Sang, and Changxin Gao. Scale pyramid network for crowd counting. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1941–1950, 2019. 7, 8

[9] Joseph Paul Cohen, Genevieve Boucher, Craig A. Glastonbury, Henry Z. Lo, and Yoshua Bengio. Count-ception: Counting by fully convolutional redundant counting. In *Proc. IEEE International Conference on Computer Vision Workshop (ICCVW)*, pages 18–26, 2017. 3, 4

[10] Jose A. Fernandez-Gallego, Shawn C. Kefauver, Nieves Aparicio Gutiérrez, María Teresa Nieto-Taladriz, and José Luis Araus. Wheat ear counting in-field conditions: high throughput and low-cost approach using RGB images. *Plant Methods*, 14(1):22–33, 2018. 1

[11] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2002–2011, 2018. 3

[12] Ross Girshick. Fast R-CNN. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015. 2

[13] Mario Valerio Giuffrida, Massimo Minervini, and Sotirios Tsaftaris. Learning to count leaves in rosette plants. In *Proc. British Machine Vision Conference Workshops (BMVCW)*, pages 1.1–1.13, 2015. 1

[14] Ricardo Guerrerogómezolmedo, Beatriz Torrejiménez, Roberto Lópezsastre, Saturnino Maldonadobascón, and Daniel Oñororubio. Extremely overlapping vehicle counting. In *Pattern Recognition and Image Analysis*, pages 423–431, 2015. 1, 2, 7, 8

[15] Haroon Idrees, Imran Saleemi, Cody Seibert, and Mubarak Shah. Multi-source multi-scale counting in extremely dense crowd images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2547–2554, 2013. 1, 2, 7, 8

[16] 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 *The European Conference on Computer Vision (ECCV)*, pages 532–546, 2018. 1, 2, 3, 5, 6, 7, 8

[17] Issam H. Laradji, Negar Rostamzadeh, Pedro O. Pinheiro, David Vazquez, and Mark Schmidt. Where are the blobs: Counting by localization with point supervision. In *The European Conference on Computer Vision (ECCV)*, pages 547–562, 2018. 1

[18] Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1324–1332, 2010. 2, 5, 8

[19] Ruibo Li, Ke Xian, Chunhua Shen, Zhiguo Cao, Hao Lu, and Lingxiao Hang. Deep attention-based classification network for robust depth prediction. 2018. 2, 3

[20] Yuhong Li, Xiaofan Zhang, and Deming Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1091–1100, 2018. 1, 3, 5, 8

[21] Jiang Liu, Chenqiang Gao, Deyu Meng, and Alexander G. Hauptmann. Decidenet: Counting varying density crowds through attention guided detection and density estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5197–5206, 2018. 2

[22] Lingbo Liu, Hongjun Wang, Guanbin Li, Wanli Ouyang, and Lin Liang. Crowd counting using deep recurrent spatial-aware network. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2018. 8

[23] Hao Lu, Zhiguo Cao, Yang Xiao, Bohan Zhuang, and Chunhua Shen. TasselNet: counting maize tassels in the wild via local counts regression network. *Plant Methods*, 13(1):79–95, 2017. 1, 2, 3, 4, 7, 8

[24] Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. Ordinal regression with multiple output cnn for age estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4920–4928, 2016. 3

[25] Daniel Oñoro-Rubio and Roberto J. López-Sastre. Towards perspective-free object counting with deep learning. In *The European Conference on Computer Vision (ECCV)*, pages 615–629, 2016. 1, 8

[26] Viresh Ranjan, Hieu Le, and Minh Hoai. Iterative crowd counting. In *The European Conference on Computer Vision (ECCV)*, pages 270–285, 2018. 2

[27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation.

In *International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 234–241, 2015. 2, 3, 4

[28] Zan Shen, Yi Xu, Bingbing Ni, Minsi Wang, Jianguo Hu, and Xiaokang Yang. Crowd counting via adversarial cross-scale consistency pursuit. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5245–5254, 2018. 2

[29] Zenglin Shi, Le Zhang, Yun Liu, Xiaofeng Cao, Yangdong Ye, Ming-Ming Cheng, and Guoyan Zheng. Crowd counting with deep negative correlation learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5382–5390, 2018. 2, 8

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *Computer Science*, 2014. 2, 3, 4, 6

[31] Vishwanath A. Sindagi and Vishal M. Patel. Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting. In *The IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, 2017. 8

[32] Vishwanath A. Sindagi and Vishal M. Patel. Generating high-quality crowd density maps using contextual pyramid cnns. In *The IEEE International Conference on Computer Vision (ICCV)*, pages 1861–1870, 2017. 2, 8

[33] Tobias Stahl, Silvia L Pintea, and Jan C van Gemert. Divide and count: Generic object counting by image divisions. *IEEE Transactions on Image Processing*, 28(2):1035–1044, 2019. 3

[34] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015. 2

[35] Karunya Tota and Haroon Idrees. Counting in dense crowds using deep features, 2015. CRCV. 8

[36] J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013. 3

[37] Cong Zhang, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. Cross-scene crowd counting via deep convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 833–841, 2015. 1, 2, 8

[38] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 589–597, 2016. 1, 2, 5, 6, 7, 8