Cell counting with inverse distance kernel and self-supervised learning

Yue Guo(✉), David Borland, Carolyn McCormick, Jason Stein, Guorong Wu, and Ashok Krishnamurthy

University of North Carolina at Chapel Hill, Chapel Hill, NC, USA
yueguo@cs.unc.edu

Abstract. We present a solution to image-based cell counting with dot annotations for both 2D and 3D cases. Current approaches have two major limitations: 1) inability to provide precise locations when cells overlap; and 2) reliance on costly labeled data. To address these two issues, we first adopt the inverse distance kernel, which yields separable density maps for better localization. Second, we take advantage of unlabeled data by self-supervised learning with focal consistency loss, which we propose for our pixel-wise task. These two contributions complement each other. Together, our framework compares favorably against state-of-the-art methods, including methods using full annotations on 2D and 3D benchmarks, while significantly reducing the amount of labeled data needed for training. In addition, we provide a tool to expedite the labeling process for dot annotations. Finally, we make the source code and labeling tool publicly available.

Keywords: Cell counting · Self-supervised · Distance transform.

1 Introduction

Our work focuses on cell counting from 2D images and 3D volumes, which is critical to a wide range of research in biology, medicine, and bioinformatics, among other fields. By casting this problem into an object detection or image segmentation problem, significant progress has been achieved with the help of recent successes in deep learning [3,9]. However, one drawback of these approaches is that they typically rely on full annotations of cells (whole-cell), the acquisition of which is a time-consuming and laborious process. To alleviate the burden of manual labeling, others propose using dot-annotations, which represent cells as a single pixel or voxel in the centroid of a cell, and attain competitive results [17,5,4]. Since dot annotations are too sparse for training, such methods typically construct a density map by "smoothing out" dot annotations via a Gaussian kernel and then train a deep model to learn a mapping between the inputs and the density maps. The final cell counting can be inferred via post-processing techniques (e.g., peak detection or connected component analysis) on the resulting density maps.

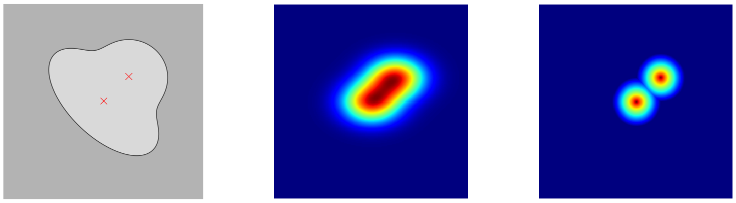

Fig. 1: An example of 2 cells with dot annotations marked by red crosses (left). Due to cell overlapping, the resulting density map with Gaussian kernel (middle) becomes a blob, whereas the inverse distance kernel (right) remains separable.

One remaining challenge is handling overlapping cells, as shown in Fig. 1, where the resulting density map becomes an inseparable blob, rendering post-processing techniques ineffective. An intuitive solution is to introduce a heuristic approach for finding the optimal width of the Gaussian kernel to avoid these blobs. However, it would be overwhelmingly difficult to design such an approach when there is a large number of cells. Inspired by recent success in crowd counting [11], we propose to replace the Gaussian kernel with the inverse distance kernel to address the overlapping issue. This distance transform uses the inverse distance of the nearest neighbor for each dot annotation to build the density maps and is able to separate overlapping cells effectively, as illustrated in Fig. 1.

Another challenge is that current deep learning-based methods are data-hungry, which is particularly problematic in the field of cell counting due to the lack of large-scale training datasets like ImageNet. Even with dot annotations, manual annotation of hundreds of cells still poses a significant challenge. On the other hand, recent progress in utilizing unlabeled data via self-supervised learning has been impressive [18,16,15]. One significant component of self-supervised learning is consistency regularization, which forces the model to yield similar outputs between the original and perturbed (e.g., adding noise) inputs for better generalization. We adopt the same idea for cell counting to reduce the reliance on manual annotations. However, we did not observe noticeable improvement during our initial implementation. Since these methods were designed for tasks like classification, with image-level labels, it is natural to apply perturbations on the same level. In contrast, our task involves pixel-level regression, with most of the image being noisy background. Applying perturbations on the whole image will mislead the model to learn image artifacts instead of the cells. Therefore, we propose a focal consistency loss, which will help the model “focus” on cells rather than the noisy background.

In addition, we notice a lack of cell labeling tools tailored for dot annotations. Current cell labeling tools are typically designed either for general labeling tasks, e.g., ImageJ, or specifically for segmentation with full annotations, e.g., Segmentor [2]. To address this issue, we developed modifications to Segmentor to directly support dot annotations. These include adjustable pre-processing to provide initial results for easy correction to reduce user workload, and visualizations to help track progress, which is especially useful for 3D volume labeling.

In summary, the contributions of this paper are four-fold:

1. A simple yet effective way to address the challenge of overlapping cells in image-based cell counting. Applying the inverse distance kernel instead of the Gaussian kernel on dot annotations enables separable density maps and outperforms state-of-the-art methods in a fully supervised setting.
2. A self-supervised approach with our novel focal consistency loss to exploit unlabeled data. It is effective when labeled data is scarce, even against methods with full annotations.
3. A labeling tool for dot annotations based on Segmentor [2], enabling a pipeline for image-based cell counting from labeling to modeling.
4. Source code and the interactive labeling tool are released to the community at `https://github.com/mzlr/cell_counting_ssl`, hoping to spur further investigation in this field.

2 Related Work

The relevant work related to this paper can be divided into two categories: image-based cell counting and self-supervised learning.

There has been significant progress in image-based cell counting, especially after the success of deep learning in various domains [7,6,5]. Since cell counting belongs to a broad topic of object counting, generic detection or segmentation based methods have been explored for 2D images [3] and 3D volumes [9]. Still, the reliance on time-consuming full (whole-cell) annotations hinders their impact. Alternatively, dot annotations have become increasingly popular for their simplicity. Notably, Xie et al. [17] applied the Gaussian kernel to dot annotations and used U-Net to learn a mapping between input images and resulting density maps. The final cell count is inferred by the integration of the density maps. Lu et al. [12] designed a Generic Matching Network to learn the exact mapping, which is capable of leveraging large object detection datasets for pre-training and adapting to target datasets. Guo et al. [4] later expanded the framework for 3D volumes and developed a unified network structure, SAU-Net, for various cell types, focusing on the universal nature of the method. However, these approaches rely on the Gaussian kernel to build the density maps and are subject to the issue of inseparable density maps for overlapping cells.

Self-supervised learning describes a class of algorithms that seek to learn from unlabeled data with auxiliary (pretext) tasks. It is a generic learning framework and can be applied in either unsupervised settings to learn useful representations for downstream tasks [8] or semi-supervised settings for a specific task [16]. This paper focuses on the latter for its simplicity. Given labeled data x and its labels y, and unlabeled data x', self-supervised learning generally has an objective function of the following form:

$$\underbrace{\mathcal{L}_l(f_\theta(x), y)}_{\text{primary task}} + \underbrace{w\mathcal{L}_u(f_\theta(x'))}_{\text{auxiliary (pretext) task}} , \tag{1}$$

where $\mathcal{L}_l$ is the supervised loss for primary tasks, e.g., Cross-Entropy (CE) loss for classification tasks or mean squared error (MSE) for regression tasks. $\mathcal{L}_u$ is the unsupervised loss for auxiliary tasks (e.g., consistency loss), w is a weight ratio between the two losses, and θ represents the parameters for model f. For example, Xie et al. [16] designed an auxiliary task by enforcing consistency constraints, i.e., the model is expected to generate consistent outputs given perturbations, and used advanced data augmentation methods as the perturbations. Sohn et al. [15] later extended this idea by considering a pair of weakly-augmented and strongly-augmented unlabeled samples. Ouali et al. [14] observed that injecting perturbations into deep layers of the network rather than the inputs is more effective for segmentation tasks and built an encoder-decoder network based on this observation. Despite numerous advances for self-supervised learning in multiple domains, its application to image-based cell counting has rarely been explored.

Inspired by recent crowd counting work [11], we propose using the inverse distance kernel to address the issue of inseparable density maps. Furthermore, we aim to take advantage of recent success in self-supervised learning for image-based cell counting and leverage unlabeled data to reduce the reliance on costly labeled data.

3 Method

Given dot annotations D, the resulting density map M traditionally can be seen as the sum of Gaussian kernels $\mathcal{N}$ with width σ centered on each individual dot annotation d: $M = \sum_{d \in D} \mathcal{N}(d, \sigma^2)$. With the corresponding image I and the resulting density map M, the goal is to train a model f with parameters θ for the mapping between the image and density map : $f_\theta(I) \rightarrow M$.

3.1 Inverse Distance Kernel

As shown in Fig. 1, Gaussian kernels suffer the problem of inseparable density maps. We propose to use the inverse distance kernel from [11]. Mathematically, we have

$$\forall i \in I, M_{dis}(i) = \frac{1}{L_2(i)^\gamma + C}, \tag{2}$$

where i denotes a pixel, and $L_2(i)$ is the Euclidean distance between pixel i and its nearest dot annotation, i.e., $\min_{d \in D} \|i - d\|_2$. Here, C is a constant to avoid dividing by zero, and γ is a decay factor to control the response rate between dot annotations and background. In practice, we set $C = 1$ and $\gamma = 0.02 L_2(i) + 0.75$, following [11].

3.2 Focal Consistency Loss

Following the general framework for self-supervised learning described above, our learning objective consists of two tasks: a primary task with a supervised

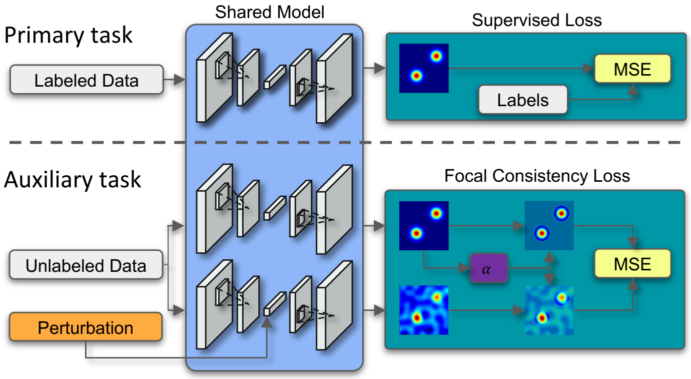

Fig. 2: Overview of the proposed method. α is the focal weight to "focus" on predicted cells instead of background noise, and MSE denotes mean square error.

loss on n labeled images I, $\frac{1}{n}\sum_I \|f_\theta(I) - M_{dis}\|_2^2$, and an auxiliary task with the consistency loss on n' unlabeled images I', $\frac{w}{n'}\sum_{I'} \|f_\theta(\delta(I')) - f_{\widehat{\theta}}(I')\|_2^2$. Here, we use the pixel-wise mean squared error (MSE) for both tasks. f_θ denotes the shared model with parameters θ, and $\widehat{\theta}$ is a fixed copy of the current θ, which is not updated via back propagation and can be viewed as containing pseudo-labels, as suggested in [16]. δ denotes the transform for perturbations. In practice we found that the vanilla framework did not generalize well to our task. Since our task is pixel-level, rather than an image-level task such as image classification for which the framework was originally proposed, adding perturbations to the whole image will inevitably learn the image artifacts in the background. Therefore, we propose a focal consistency loss, $\frac{w}{n'}\sum_{I'} \alpha\|f_\theta(\delta(I')) - f_{\widehat{\theta}}(I')\|_2^2$ with focal weight $\alpha = \sum_{d' \in D'} \mathcal{N}(d', \sigma^2)$, where D' is the set of the local maximums in the current predicted density maps $f_{\widehat{\theta}}(I')$, which are considered as predicted cells and can be inferred by a maximum filter. α ensures that we only calculate the consistency loss in the vicinity of the local maximum, i.e., focusing on cells and ignoring background. We also considered several commonly used perturbations, e.g., random contrast/sharpness/brightness/noise, etc., and found directly injecting noise into the last layer of the encoder yields the best performance, as suggested in [14]. Our final loss function is

$$\mathcal{L}_{\text{final}} = \frac{1}{n}\sum_I \|f_\theta(I) - M_{dis}\|_2^2 + \frac{w}{n'}\sum_{I'} \alpha\|f_{\theta_{\text{dec}}}(\delta(f_{\theta_{\text{enc}}}(I'))) - f_{\widehat{\theta}}(I')\|_2^2, \quad (3)$$

where $f_{\theta_{\text{enc}}}$ and $f_{\theta_{\text{dec}}}$ are the encoder and decoder of the network, respectively. The overview of the proposed method is shown in Fig. 2.

3.3 Implementation Details

This work uses SAU-Net [4] as the base model, favoring its versatile nature for both 2D and 3D. We use the Adam optimizer with a cosine decaying learning rate, and the initial value for the schedule is 0.001. For the weight w in Eq. 3 we follow the ramp-up schedule in [14], which exponentially increases from 0 to 1, avoiding noisy signals in the early stage of training. All the hyper-parameters are shared across 2D and 3D experiments, highlighting the versatility of our method.

4 Labeling Tool

To streamline the labeling process for dot annotations we added a dot annotation mode to the Segmentor 3D annotation tool [2]. We first apply Otsu's method and connected component analysis for cell detection, and use the centroids of each component as initial results for user refinement, e.g., identifying merged cells and making the appropriate corrections. In the 2D slice view, dot annotations in nearby z-axis slices are marked with separate visualizations to ease the effort of tracking completed regions. In an adjacent 3D view, volume rendering can be used to verify the placement of the dot annotations.

To quantify the decrease in labeling time for dot annotations vs. full (whole-cell) annotations, we asked four users to dot-annotate four 3D volumes with 156 cells in total and compared the time to the full annotation time previously reported in [2]. The average speed for dot annotation is ∼0.3 minutes per cell, whereas full annotations take ∼8 minutes per cell, a speedup of ∼27x. In addition to improved labeling efficiency, our experiments in the next section show that our dot-annotation method outperforms a state-of-the-art full-annotation method.

5 Experiments

We used two benchmarks to evaluate the proposed method, the 2D VGG dataset [10] and a 3D light-sheet dataset [9]. VGG is a synthetic dataset that contains 200 fluorescence microscopy cell images with an even split between training and test sets, and each image containing 174 ± 64 cells. The 3D dataset comprises light-sheet images of the mouse cortex with 16 training volumes and 5 test volumes. Each volume includes 861 ± 256 cells. This dataset contains full (whole-cell) annotations, and we convert them into dot annotations by using the centroids of the full annotations. Note that we only use a fraction of the training data to study reliance of the proposed method on labeled data; nonetheless, we always evaluate our method on the **whole** test set. Previous works [16,15] suggest self-supersized learning benefits from a large amount of unlabeled data. Following this, we use an additional 55 unlabeled samples for the 3D experiment on the light-sheet dataset, which are all the available unlabeled data from the original work. For the 2D experiment on the VGG dataset, we use 8 out of 100 training data for the supervised training and treat the remaining 92 of the unused training data as unlabeled data for self-supervized training.

We use two metrics to evaluate the proposed method: the mean absolute error (MAE) between the predicted and ground-truth cell counts, and the F_1 score for cell detection. We follow the post-processing techniques from [9,4], i.e., perform

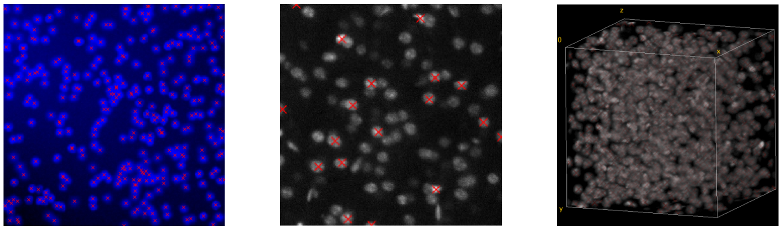

Fig. 3: Sample images of 2D VGG dataset (left) and 3D light sheet dataset with 2D (middle) and 3D (right) view. Dot annotations, typically in the centroids of cells, are marked with red cross overlays.

Connected Component Analysis after thresholding and use the centroids of each component as cell locations. Depending on the type of annotations, a detection is valid when a predicted centroid lies within a full (whole-cell) annotation of a cell or a radius of a dot annotation, where the radius is an empirical radius of cells. MAE has an inherent deficiency since it is unable to distinguish correct cell counts. For a trivial case with only one cell, a method could miss that cell and yield a 0 F_1 score but pick up background noise as a cell and achieve a perfect MAE of 0. Without location information, it would be impossible to determine this error using MAE. Nonetheless, we still report MAE for a direct comparison with previous work.

5.1 Ablation Study

A 3D light-sheet dataset is used for our ablation study since its full annotations enable a comparison with more methods. This work focused on learning using limited data; therefore, only one volume is used for training. All the experiments are conducted with the same baseline SAU-Net, and the model is initialized from the same random weights, which could help us better attribute the source of improvement. Table 1 shows that using the inverse distance kernel already outperforms the baseline and SAU-Net with full annotations, and our proposed self-supervised learning further improves the performance.

Table 1: Ablation study

Method	MAE	F_1(%)
Gaussian kernel (*baseline*)	57.8	93.55
Full annotations*	42.2	94.12
Inverse distance kernel	13.8	95.21
Inverse distance kernel + self-supervised learning (*proposed*)	**12.4**	**95.86**

* *Implementation following [9] except replacing U-Net with SAU-Net for consistency.*

Table 2: 2D VGG dataset

Method	MAE	F_1(%)	N_{train}
Arteta et al. [1]	5.1	93.46	32
GMN [12]	3.6	90.18	32
SAU-Net [4]	**2.6**	94.51	64
Ours	5.6	**96.78**	**8**

Table 3: 3D light sheet dataset

Method	MAE	F_1(%)	N_{train}
NuMorph [9] (*full-annot.*)	50.1	95.37	4
CUBIC [13] (*unsupervised*)	36.2	95.43	-
SAU-Net [4] (*dot-annot.*)	42.2	93.97	4
Ours (*dot-annot.*)	**12.4**	**95.86**	**1**

5.2 Comparison to State-of-the-art

We compare our proposed method with other state-of-the-art approaches on the 2D VGG and 3D light-sheet datasets. To show that our method reduces the reliance on labeled data, we drastically limit the number of labeled data to **25% or less** of the amount used in other state-of-the-art approaches. Table 2 and 3 present the results. For both datasets, our method improves the performance in terms of F_1 score, even improving upon the full-annotation method, while substantially reducing the amount of labeled data needed for training. Our method does not outperform [4], the state-of-the-art Gaussian method, on the MAE metric for the VGG dataset. This is due to the fact that this synthetic dataset contains severe overlapping, as shown in Fig. 3, and the ground-truth is pre-defined; otherwise, it would be considerably challenging for human annotators. For those cases, the cell count can still be obtained by integration of the Gaussian density maps, although they are inseparable. On the other hand, our method attains better performance on the F_1 metric with location information. In our opinion, although widely used for cell counting, MAE provides a limited evaluation for cell counting methods. Since it ignores the location information of individual cells, we also calculate the F_1 score for a comprehensive review. Overall, by utilizing unlabeled data, our method outperforms most state-of-art methods in 2D and 3D cases with a quarter or less of labeled data.

6 Discussion

We introduce a framework for image-based cell counting using dot annotations, including a labeling tool and an improved model with a highly efficient training scheme. Compared to conventional Gaussian kernel methods, our model exploits the inverse distance kernel for separable density maps, enabling post-processing techniques for cell location in addition to cell counts. By leveraging unlabeled data, our self-supervised pipeline with a novel focal consistency loss allows a drastic reduction of labeled data and achieves state-of-the-art or very competitive performance on both 2D and 3D benchmarks, even compared to methods using full annotations. Finally, we add dot annotation specific features to an existing labeling tool to facilitate the annotation process. We notice that, in some works for image classification [16,15], the advantage brought by self-supervised learning against fully supervised learning diminishes as the number of labeled data grows. In the future, we plan to investigate this matter for cell counting.

7 Acknowledgments

This work was supported in part by the National Science Foundation under grants OCI-1153775 and OAC-1649916. The authors would to like thank Carla Escobar and Tala Farah for their participation in the user study of our labeling tool.

References

1. Arteta, C., Lempitsky, V., Noble, J.A., Zisserman, A.: Detecting overlapping instances in microscopy images using extremal region trees. Medical image analysis **27**, 3–16 (2016)
2. Borland, D., McCormick, C.M., Patel, N.K., Krupa, O., Mory, J.T., Beltran, A.A., Farah, T.M., Escobar-Tomlienovich, C.F., Olson, S.S., Kim, M., et al.: Segmentor: a tool for manual refinement of 3d microscopy annotations. BMC bioinformatics **22**(1), 1–12 (2021)
3. Caicedo, J.C., Goodman, A., Karhohs, K.W., Cimini, B.A., Ackerman, J., Haghighi, M., Heng, C., Becker, T., Doan, M., McQuin, C., et al.: Nucleus segmentation across imaging experiments: the 2018 data science bowl. Nature methods **16**(12), 1247–1253 (2019)
4. Guo, Y., Krupa, O., Stein, J., Wu, G., Krishnamurthy, A.: Sau-net: A unified network for cell counting in 2d and 3d microscopy images. IEEE/ACM Transactions on Computational Biology and Bioinformatics (2021)
5. Guo, Y., Stein, J., Wu, G., Krishnamurthy, A.: Sau-net: A universal deep network for cell counting. In: Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics. pp. 299–306 (2019)
6. Guo, Y., Wang, Q., Krupa, O., Stein, J., Wu, G., Bradford, K., Krishnamurthy, A.: Cross modality microscopy segmentation via adversarial adaptation. In: International Work-Conference on Bioinformatics and Biomedical Engineering. pp. 469–478. Springer (2019)
7. Guo, Y., Wrammert, J., Singh, K., Ashish, K., Bradford, K., Krishnamurthy, A.: Automatic analysis of neonatal video data to evaluate resuscitation performance. In: 2016 IEEE 6th International Conference on Computational Advances in Bio and Medical Sciences (ICCABS). pp. 1–6. IEEE (2016)
8. Kolesnikov, A., Zhai, X., Beyer, L.: Revisiting self-supervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1920–1929 (2019)
9. Krupa, O., Fragola, G., Hadden-Ford, E., Mory, J.T., Liu, T., Humphrey, Z., Rees, B.W., Krishnamurthy, A., Snider, W.D., Zylka, M.J., et al.: Numorph: tools for cellular phenotyping in tissue cleared whole brain images. bioRxiv pp. 2020–09 (2021)
10. Lempitsky, V., Zisserman, A.: Learning to count objects in images. Advances in neural information processing systems **23**, 1324–1332 (2010)
11. Liang, D., Xu, W., Zhu, Y., Zhou, Y.: Focal inverse distance transform maps for crowd localization and counting in dense crowd. arXiv preprint arXiv:2102.07925 (2021)
12. Lu, E., Xie, W., Zisserman, A.: Class-agnostic counting. In: Asian conference on computer vision. pp. 669–684. Springer (2018)

13. Matsumoto, K., Mitani, T.T., Horiguchi, S.A., Kaneshiro, J., Murakami, T.C., Mano, T., Fujishima, H., Konno, A., Watanabe, T.M., Hirai, H., et al.: Advanced cubic tissue clearing for whole-organ cell profiling. Nature protocols **14**(12), 3506–3537 (2019)
14. Ouali, Y., Hudelot, C., Tami, M.: Semi-supervised semantic segmentation with cross-consistency training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12674–12684 (2020)
15. Sohn, K., Berthelot, D., Li, C.L., Zhang, Z., Carlini, N., Cubuk, E.D., Kurakin, A., Zhang, H., Raffel, C.: Fixmatch: Simplifying semi-supervised learning with consistency and confidence. arXiv preprint arXiv:2001.07685 (2020)
16. Xie, Q., Dai, Z., Hovy, E., Luong, M.T., Le, Q.V.: Unsupervised data augmentation for consistency training. arXiv preprint arXiv:1904.12848 (2019)
17. Xie, W., Noble, J.A., Zisserman, A.: Microscopy cell counting and detection with fully convolutional regression networks. Computer methods in biomechanics and biomedical engineering: Imaging & Visualization **6**(3), 283–292 (2018)
18. Zhai, X., Oliver, A., Kolesnikov, A., Beyer, L.: S4l: Self-supervised semi-supervised learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1476–1485 (2019)