Learn from Object Counting: Crowd Counting with Meta-learning

Changtong Zan¹,² | Baodi Liu¹ | Weili Guan³ | Kai Zhang¹ | Weifeng Liu¹,² ©

¹ China University of Petroleum (East China), Qingdao, China
² State Key Laboratory of Integrated Services Networks (Xi'an University), Xi'an, China
³ Faculty of Information Technology, Monash University, Clayton Campus, Australia

Correspondence
Weifeng Liu, College of Control Science and Engineering, China University of Petroleum (East China), Qingdao 266000, China.
Email: liuwf@upc.edu.cn

Founding information:
Project supported by the National Natural Science Foundation of China (Grant No. 61671480); the Major Scientific and Technological Projects of CNPC under Grant ZD2019-183-008; This work was supported by the Open Project Program of the National Laboratory of Pattern Recognition (NLPR) (Grant No. 202000009)

Abstract
The objective of crowd counting is to learn a counter that can estimate the number of people in a single image. So far, most of the proposed work evaluates the crowd density by fitting the constructed density map corresponding to the sample. The performance of those algorithms depends on a large amount of carefully prepared data. However, a significant problem with crowd data sets is the difficulty of labeling. To address such a situation, utilizing object counting data in few-shot scenes is considered and an efficient algorithm to extract the meta-information is proposed, thus improving the accuracy and convergence rate of the crowd counting tasks. Specifically, the counting network is trained with only object counting tasks constructed on different domains during the meta-training phase. Then, the meta-counter is testing on crowd counting tasks in the meta-testing stage. Experimentally, it is demonstrated that the above way improves the converge rate and accuracy of crowd counting tasks on three crowd counting datasets when meta-training on ten-type object counting tasks.

1 | INTRODUCTION

With the increase of the urban population, people quickly gather in many public places, such as shopping centers, train stations on holidays, and crossings during peak hours. This situation has led to a sharp increase in local population density, which brings public safety risks. In these cases, useful crowd density estimation through surveillance equipment is significant for public safety and management [1]. As a proper method, the CNN-based crowd counting method [2, 3] can effectively improve the counting accuracy by extracting features from CNN, which has attracted more and more attention from the research community.

Many crowd counting relevant work have been proposed, such as dataset construction [3, 4], survey [1], CNN-based crowd counting algorithms [5, 6]. The crowd counting algorithms can roughly be divided into two categories[1], including the traditional crowd counting approach [7–9], and CNN-based modern crowd counting approach [2, 3, 10–14].

The traditional crowd counting approach includes the detection-based method, the regression-based method, and the density estimate method. Detection-based methods [7] often use a sliding window in a picture to detect the target and calculate the total number of people. However, these algorithms are easily affected by object occlusion. The regression-based methods [5] can effectively deal with the occlusion and background clutter problems by directly learning the mapping from image patches to counting results. The density estimate methods [6] consider the spatial information, fit the mapping relationship between image and density map, and predict a more accurate count.

Recently, thanks to the powerful image information representation capability of CNN. The CNN-based modern crowd counting method that effectively improves the counting accuracy by extracting features through CNN has attracted more and more research communities. Researchers used raw CNN to fit the crowd image’s density map [10, 11]. With the development of crowd density estimation methods, FCN has become
the mainstream network structure of existing crowd estimation methods through its sound effects [2, 3, 12–14]. In recent years, the cross-domain crowd counting method has gradually attracted people’s attention. SE CycleGAN [3] introduced SSIM [15] into the objective function of traditional Cycle-GAN, and the difference between the synthetic domain and the real domain is reduced by domain transformation. FSC [12] distinguishes the crowd and background in the semantic domain by introducing a pre-trained segmentation model and uses adversarial learning to effectively reduce errors in the ground by aligning features in the semantic space.

Although these methods can effectively calculate the crowd’s density in the picture, there are still some limitations. (1) The CNN-based crowd counting method needs to learn recognition features from a large number of labeled data, and large-scale well-prepared data is significant. However, accurate labeling of crowd images requires much money and human resources. Therefore, it is substantial to learn general meta-information from other object-labeled image datasets (e.g. object counting dataset). (2) Almost all proposed crowd counting methods only consider crowd images. However, considering object counting and designing a crowd counting method can help cross the differences between different object counting domains and propose a unified counting framework. Thus it is a meaningful attempt.

This paper aims to extract the meta-information shared by the object counting and the crowd counting through meta-learning and obtain the meta-information to help the crowd counting. We have some counting tasks that do not include crowd counting to meta-train the crowd counting model, and all these tasks are few-shot scenes. We make a good initialization for the model, which converges faster in crowd counting and has better generalization performance. As illustrated in Figure 1, first, we adopt the meta-learning algorithm to extract helpful meta-information from the object counting task in the meta-training stage. Specifically, we performed a good initialization for the counting model, which can converge faster and count more accurately with meta-information from different domains. To achieve that, we utilize the two-level training paradigm MAML [16] for the CNN-based crowd counting model. We construct k-shot tasks in ten object counting task domains during the meta-training phase and perform generalization performance testing with the tasks sampled on the crowd counting domain in the meta-testing stage.

We extensively evaluated our scheme’s effectiveness when extracting meta-information from counting tasks about ten kinds of object, including cell, Maize Tassel, vehicle (count the number of all traffic vehicles), car, bus, bicycle, motor, tricycle, van, truck. It is worth mentioning that the last eight types of counting images are all sampled from Visdrone-19. The network consists of the feature extractor of VGG-16 and an additional density map estimator. The crowd counting tasks are sampled from the Shanghai-Tech dataset and the Beijing-BRT dataset during the meta-testing phase. All the related information about the dataset is described in the experimental result part. We test the model with three baseline methods described in Section 4.2 on 100 crowd counting tasks and present the result.

Overall, we focus on the problem that extracts the universal meta-information for crowd counting performance improvement, using data from other domains, and the contributions are as follows:

1) By introducing MAML, we proposed a cross-domain from object counting to crowd counting few-shot crowd counting algorithm, effectively extracting meta-information common to technical tasks.
2) Our method can effectively improve the crowd counting task’s accuracy and speed up the convergence speed.
3) We conducted comprehensive experiments on the ShanghaiTech dataset and Beijing-BRT dataset and conducted an empirical analysis of our method. The experimental results verify the effectiveness of our approach.

This article is organized as follows. We will introduce related other work in Section 2. Our method will be explained in Section 3. Section 4 presents the results of the experiment and analyses. Section 5 makes a conclusion.

2 RELATED WORK

In this section, we discuss the published work that is most relevant to our method. It mainly includes three parts, the cross-domain crowd counting method (2.1), the meta-learning method (2.2), and the object counting method (2.3).

2.1 Cross-domain crowd counting methods

Cross-domain crowd counting helps improve the crowd counting task by considering the similarity between different domains, including the differences between other crowd data domains, the differences between object data domains. To the best that we know, PPPD [8] and CAC [9] are rare jobs that consider a combination of crowd counting tasks and object counting tasks. PPPD [8] is a patch-based multi-domain object counting network. This work trains an additional domain-specific normalization layer and scaling layer, which can complete various object counting tasks that are not limited to crowd counting. Moreover, CAC [9] regards calculating tasks as a form of matching task. Based on the instances’ self-similarity in an image, a general matching network independent of the class is proposed. However, both the above two algorithms do not consider the few-shot scene. Few-shot SACC [14] focuses on the scene adaptive crowd counting problem with a few-shot restrain, and the purpose is to adapt the model to a new scenario with a small amount of labeled data.

2.2 The meta-learning method

Meta-learning, namely learning-to-learn, attempts to extract meta-information from the meta-training process, which can help the model converge faster and better in training new tasks during meta-testing. The research can be traced back to some early works [17, 18]. Recently, more researches have focused on the optimization-based meta-learning method, which attempts to provide a good initialization condition for the global shared meta-learner, and it can quickly complete convergence when a new task arrives. MAML [16] is a typical optimization-based meta-learning method. The meta-learner parameters are updated by gradients obtained on the query set, which can effectively initialize any model.

MT-GAN [19] focuses on the domain translation task by designing the cycle-consistency meta-objective function in GAN’s meta-training process and ensuring the model’s generalization performance with useful initialization.

2.3 Object counting

Object counting attempts to count the number of a particular object from a single image sample and has applications in many fields, such as agriculture [20], environmental protection [21], and medicine [22, 23]. Penguin counting [21] adds foreground–background segmentation and local uncertainty estimation to the density assessment method. TasselNet [20] realizes partial counting regression through a deep convolutional network and realizes effective counting of maize tassels. Crowd counting [3, 4, 14] is the unique type of counting problem in object counting because its importance in public safety has attracted widespread attention. Although the vast difference between the human body and other counting objects makes it impossible to transfer directly between different counting tasks, the common features like structural features and distribution patterns also indicate the possibility of knowledge transfer between different tasks.

3 METHOD

In this section, an idea that spans the domain difference between object counting and crowd counting is shown. We introduce the problem settings in Section 3.1. And for the specific description, the algorithms, that are explained in Section 3.2 are designed. The full algorithm of our method is outlined in Algorithm 1 in a general case.

3.1 Problem formulation

The purpose of the object counting is to count how many objects of a particular type in each image, such as human (crowd counting [10, 19]), wild animals (wildlife counting [21]), and vehicles (vehicles counting [24–27]). The crowd counting task, which we focused on, becomes extremely important because of
its special significance to public social security. Among the existing crowd counting methods, the density estimation method based on CNN uses CNN or FCN with different structures to fit the mapping function between image X and corresponding density map y and has achieved good results. However, they rely on the difficult to construct crowd counting dataset and cannot obtain the meta-information from other object counting tasks when performing crowd counting tasks. To tackle this problem, we analyze the problem of crowd counting from the perspective of meta-learning. The goal of our method is to address the generalization problem in existing counting algorithms. For the certain similarity between object counting tasks and crowd counting tasks, meta-learning is introduced to capture the prior-information in object counting, thereby improving the training convergence speed and the counting accuracy of crowd counting tasks.

In the meta-learning environment, a k-shot task format represents both crowd counting tasks and object counting tasks. We are committed to learning some priori from the processing of object counting, which is helpful to crowd counting. Suppose we have object counting task distribution \( P(T) = \{ P_o(T), P_c(T) \} \), where \( P_o(T) \) and \( P_c(T) \) represent the crowd counting task distribution and other object counting task distribution such as cell counting, vehicle counting. Specifically, the \( N + M \) k-shot task \( \{ T \}^{N+M} \) extracted from \( P(T) \) is composed of \( k \) tuples composed of samples \( x \in X \) and corresponding density maps \( y \in Y \), respectively, and is divided into two parts: support set \( S_{T} = \{ x_{j}, y_{j} \mid j=1, \ldots, k \} \) and query set \( Q_{T} = \{ x_{i}, y_{i} \mid i=1, \ldots, k \} \) where \( N, M \) are the number of tasks used in meta-training and metatesting separately, \( x \) is the image to be counting, and \( y \) is the corresponding density map, \( k \) refers to the number of tuples in the support set of k-shot task, and \( l \) represents the number for query set.

During meta-training, we have got \( N \) tasks from \( P_o(T) \). This method can be regarded as a two-level optimization algorithm, including inner loop and outer loop. The inner loop evaluates the performance of meta-learners in the task. The outer loop updates the parameters of the meta-learners according to the query loss of the inner loop. Specifically, each inner loop replica of the meta-learner performs several gradient descents on the support set to obtain the task-specific parameters. It verifies the generalization performance of the obtained parameters in the query set. After that, the meta-learner performs one step gradient descent of the outer loop according to the query data's generalization loss. During meta-testing, we have got \( M \) tasks from \( P_c(T) \). And then, we freeze the meta-learner parameters, use replicas for fine-tuning on different support sets in parallel, and represent the result with the average query accuracy of multiple tasks.

3.2 Meta-learning for crowd counting

The purpose of our method is to extract meta-information familiar to counting tasks through meta-training in a variety of object counting tasks so that the training speed and predic-

tion accuracy of the network can be accelerated in the meta-testing stage.

Formally, we use the feature extractor of VGG-16, plus a density estimator composed of two layers of convolution as the base counting network \( C_{\Theta} \) which is parameterized by \( \Theta \). Before meta-training, we initialize the VGG-16 part with the pre-trained parameters of classifying tasks on ImageNet. In each batch of meta-training, we randomly select \( B \) object counting tasks from the task distribution \( P_o(T) \). Here, we need a good diversity of counting tasks. In this paper, ten types of counting tasks are taken as examples, that is, cell counting tasks, Maize Tassel counting tasks, vehicle counting tasks, car counting tasks, bus counting tasks, bicycle counting tasks, motor counting tasks, tricycle counting tasks, van counting tasks, truck counting tasks.

For a particular task, there are \( k \) tuples \( T = \{ S_{T}, Q_{T} \} \) composed of the object images and density maps. We first obtain a one-time-use copy network \( C_{\Theta'} \) based on the parameters of the meta-counting network \( C_{\Theta} \). In the inner loop, we use a target function that can be used uniformly between object counting and crowd counting with the density estimate method. Precisely, the loss that generates gradient is calculated according to the corresponding label matrix \( y_{i} \) and the output density map matrix \( y'_{i} \) of the network after forwarding propagation task-special type object image \( x_{i} \):

\[
L_{\text{inner}} = \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{z=1}^{W} \left( y'_{ijz} - y_{ijz} \right)^{2},
\]

\[
\Theta' = \Theta - \eta \nabla_{\Theta} L_{\text{inner}}(T),
\]

where \( j \) and \( z \) are the indexes of the element in a matrix, \( \eta \) is the inner task learning rate and \( \Theta' \) is the copy of \( \Theta \). After several gradient update steps are completed, we evaluate the replication network’s generalization performance based on the same objective function in the query set.

For the entire meta-training, we calculate the gradient with the loss of the replica network corresponding to each task on the query set, which we will demonstrate below. And, thanks to the broad applicability of the density estimate method, we can consider different types of object counting tasks \( T^k \) sampled from \( P(T) = \{ P_o(T), P_c(T) \} \). The meta-optimization process can be written in the following form:

\[
\Theta = \Theta - \gamma \nabla_{\Theta} \sum_{T \sim P(T) = P(T) \cap \{ P_o(T), P_c(T) \}} L_{T}(C_{\Theta'}),
\]

where \( \gamma \) is the learning rate, in the project; we used the Adam optimization method instead of straightforward gradient descent here. After the meta-training is over, we will test the final effect of the meta-parameters on the k-shot crowd counting task drawn from \( P_c(T) \).

4 EXPERIMENTAL EVALUATION

In the experimental evaluation, we first describe the related information about the object counting dataset and crowd
counting dataset used in the experiment (Section 4.1). The specific experimental settings, comparison method, and evaluation criteria are present (Section 4.2), and finally, we present the results of the experiment (Section 4.3).

4.1 Datasets

In recent years, crowd counting has developed rapidly, and many related datasets have been established. In this paper, the Shanghai-Tech dataset and Beijing-BRT dataset are selected to construct the meta-testing stage’s evaluation task. At the same time, we choose three datasets of MTC, DCC, and Visdrone2019 to build the meta-training data.

Shanghai-Tech [28] is one of the most widely used crowd counting datasets so far, including 1198 images and corresponding labels, with a total of 330165 annotations. Simultaneously, according to the considerable difference of data distribution, the entire dataset is divided into two parts: Shanghai-Tech part_A (SH_A), Shanghai-Tech part_B (SH_B). The data of the former (SH_A) is randomly obtained from the Internet. There are 482 images, 241677 instances, and the average crowd density in each sample is 501. The latter (SH_B) is taken from a busy street in Shanghai, with 716 images and 88488 instances. In comparison, the average density of people in each sample is smaller than SH_A.

Beijing-BRT [29] is a relatively small crowd count dataset, which is composed with 1280 images and 16795 instances, and the average crowd density is only 13. All the data were taken at the Bus Rapid Transit in Beijing.

MTC [20] (maize tassels counting dataset) contains 361 images for wild maize tassels. The data are collected during 2012–2015, from 4 different places in China. The number of instances in each image is from 0 to 100.

DCC [30] (Dublin cell counting dataset) consists of 177 cell images, covering many tissues and species. All the data are collected with the microscope and real cell organization, which guarantee the actual validity. DCC data have an average count of 34.1 and a standard deviation of 21.8.

Visdrone2019 [31] is an object detection dataset, and there is a corresponding challenge on ECCV2020. Considering that all the labels are of the bounding box type, we use the bounding box’s center point. As the labels contain different object types, we could consider constructing more types of object counting tasks.

In the experiment, for meta-training, we randomly selected cell counting tasks, maize tassels counting tasks from MTC and DCC separately, and extracted the following eight categories of tasks from and Visdrone2019: vehicle counting, car counting, bus counting, bicycle counting, motor counting, tricycle counting, van counting, truck counting. We respectively performed meta-testing on the above crowd counting datasets and generated the k-shot crowd counting tasks to evaluate our method’s effect. The experiments are performed using a Tesla V100 GPU with a batch size of 4 and implemented using the C3 framework [32] based on Pytorch.

4.2 Experimental setup

4.2.1 Evaluation criteria

Following convention, we also use MSE (Mean Square Error) and MAE (Mean Absolute error) to evaluate the method’s counting effectiveness. The smaller their value, the better the performance of the model. The corresponding calculation formula is shown below:

\[
MAE = \frac{1}{L} \sum_{i=1}^{L} |y_i - y'_i|, \quad (4)
\]

\[
MSE = \sqrt{\frac{1}{L} \sum_{i=1}^{L} (y_i - y'_i)^2}, \quad (5)
\]

where \(L\) is the number of query image in a task, \(y_i\) and \(y'_i\) are the real count and predicted count of the sample separately. Especially, we make a 10 step gradient descent on the meta-testing task with the initialization of meta-trained parameters. Then, we take the average MAE and MSE of 100 tasks as the final evaluation criteria.

4.2.2 Parameter setting

During the meta-training, the learning rate of inner loop \(\eta\) in Equation (2) and the weight parameters \(\gamma\) in Equation (3) are set to \(10^{-3}\) and \(10^{-5}\), respectively. The Adam algorithm is utilized for training the meta-networks. In each meta-batch, we randomly select different types of tasks: vehicle counting, car counting, bus counting, bicycle counting, awning-tricycle counting, motor counting, tricycle counting, van counting, cell counting, and wild maize tassels counting. And the batch size is 4. Then we construct the used tasks for meta-training. All the images are cropped into 576 × 768. We only consider the 10-shot task, the 5-shot task, and the 1-shot task in which the query set number is 5.

The parameters of front 10 convolution layers of VGG-16 with batch normalization pre-trained on ImageNet are utilized for feature extraction’s initialization. The density estimate estimator is random initialized. After 510 iterations, we stop the updating of the meta-network.

4.2.3 Comparison method

In the few-shot scene, we compared our approach with three baseline methods that use the same counting network. All the methods used for experiments include four kinds: (1) our meta-learning approach that meta-training with ten types of object counting tasks. (2) the scratch method of directly using the object counting network to count people. Specifically, we trained the model on Visdrone-19 vehicles counting and fine-tuning on crowd counting tasks. (3) another method is directly training on the crowd counting tasks. (4) Finally, we used the
Table 1 shows the tests’ results on three crowd datasets under 1-shot, 5-shot, and 10-shot conditions. We noticed that our method achieved the lowest MAE and MSE among the three datasets in 5-shot tasks and 10-shot tasks after meta-trained through the object counting tasks. For the 1-shot task, our method achieved the best results in the BRT dataset, while the results were comparable in SH_A and SH_B. Our method uses object counting datasets for meta-training thus is sensitive to objects in the background when there are few crowd counting samples. The results show that due to the significant inter-domain differences between objects and people, it is infeasible to directly use the network trained in the object counting task, which will bring meaningless resolution. Moreover, it is also not suggested to treat the few-shot task directly as the lack of information. In summary, our method effectively extracts the general meta-information in the counting task and improves the counting accuracy of the crowd counting task by using it.

4.3.2 Convergence rate evaluation

The above experiments have proved that by training on different types of object counting tasks, our method can extract shared information in the counting task and improve the crowd counting task’s effectiveness. In Figure 2, we show the curve of the result of our method’s predictions as the number of iterations increases for different crowd densities for the 5-shot task. Besides the accuracy of the crowd counting task, we also see that our method could train faster during the meta-testing phase.

4.3.3 Visualization results

We give the density maps corresponding to the pictures predicted by the network with different methods to demonstrate our method’s effectiveness. In Figure 3, we can find that the predicted density map is similar to GT, but there is still noise. When the crowd density increases, the predicted density map is also affected by noise. We believe that the main reason is that the network has performed meta-training in the object counting dataset. When migrating to the crowd counting task, the background objects (such as backpacks, shoes) are also more sensitive.

4.3.4 Ablation experimental results

We provide the results of ablation experiments to analyze the impact of pre-training on the final effect. The results are shown in Table 2. We can find that using the VGG-16 parameters pre-trained on ImageNet, the experiment’s effect has been significantly improved. We think that the main reason is that pre-training effectively reduces the possibility of meta-training falling into a local minimum, thereby improving the final effect.
FIGURE 2  The prediction density curve. In the task of constructing 5-shots in the SH_A dataset, we show the changes in the predicted value of our model and Algorithm 2 for the same image during the fine-tuning process.

FIGURE 3  Visualization results of the predicted density map in the SH_B dataset. “GT” and “Pred” represent the labeled and predicted count, respectively.

5 CONCLUSION

We introduce a meta-learning-based method that can obtain meta-information from object counting tasks, thus improving the performance and convergence rate when facing crowd counting tasks. We are the first to consider the model’s generalization performance from object count to crowd count under a few-shot scene to the best of our knowledge. Our method successfully extracts the shared information between the crowd counting task and the object counting task. Experimental results prove that our approach can still improve crowd counting in the meta-testing phase when only using object images and meta-training tags. We will consider finding a more generalized method to extract meta-information between different counting tasks in future work.

TABLE 2  Average MAE and MSE of 5-shot ablation experimental results. ‘fail’ means almost all of the outputs are meaningless count, that is, a value of 0. The best performers are in bold.

| Algorithm | BRT(MAE/MSE) | SH_A(MAE/MSE) | SH_B(MAE/MSE) |
|-----------|--------------|---------------|---------------|
| With pre-train | 10.92/12.71 | 186.17/246.77 | 34.34/43.92 |
| Without pre-train | 10.51/12.19 (fail) | 288.75/372.23 | 49.61/64.89 |
REFERENCES

1. Gao, G., et al.: CNN-based density estimation and crowd counting: A survey. arXiv preprint, arXiv:2003.12783, 2020
2. Gao, J., et al.: Feature-aware adaptation and density alignment for crowd counting in video surveillance. IEEE Trans. Cybern. (2020)
3. Wang, Q., et al.: Pixel-wise crowd understanding via synthetic data. International Journal of Computer Vision 129, 225–245 (2021)
4. Wang, Q., et al.: Nwpu-crowd: a large-scale benchmark for crowd counting and localization. IEEE Trans. Pattern Anal. Mach. Intell. (2020)
5. Liu, J., et al.: Decidenet: counting varying density crowds through attention guided detection and density estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5197–5206. IEEE, Piscataway (2018)
6. Gao, J., Wang, Q., Li, X.: Pcc net: perspective crowd counting via spatial convolutional network. IEEE Transactions on Circuits and Systems for Video Technology 30(10), 3486–3498 (2019)
7. Lempitsky, V., Zisserman, A.: Learning to count objects in images. In: Proceedings of the Annual Conference on Advances in Neural Information Processing Systems, pp. 1324–1332. MIT Press, Cambridge (2010)
8. Walach, E., Wolf, I.: Learning to count with cnn boosting. In: Proceedings of the European Conference on Computer Vision, pp. 660–670. Springer, Cham (2016)
9. Bai, H., et al.: Crowd counting on images with scale variation and isolated redundant counting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2547–2554. IEEE, Piscataway (2013)
10. Lempitsky, V., Zisserman, A.: Learning to count objects in images. In: Proceedings of the Annual Conference on Advances in Neural Information Processing Systems, pp. 1324–1332. MIT Press, Cambridge (2010)
11. Wang, C., et al.: Deep people counting in extremely dense crowds. In: Proceedings of the ACM International Conference on Multimedia, pp. 1299–1302. ACM, New York (2015)
12. Cao, X., et al.: Scale aggregation network for accurate and efficient crowd counting. In: Proceedings of the European Conference on Computer Vision, pp. 734–750. Springer, Cham (2018)
13. Liu, Y., et al.: Point in, box out: beyond counting persons in crowds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6469–6478. IEEE, Piscataway (2019)
14. Sam, D.B., et al.: Locate, size and count: accurately resolving people in dense crowds via detection. IEEE Trans. Pattern Anal. Mach. Intell. (2020)
15. Zhang, Y., et al.: Single-image crowd counting via multi-column convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 589–597. IEEE, Piscataway (2016)
16. Liu, J., et al.: Decidenet: counting varying density crowds through attention guided detection and density estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5197–5206. IEEE, Piscataway (2018)
17. Liu, J., et al.: Decidenet: counting varying density crowds through attention guided detection and density estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5197–5206. IEEE, Piscataway (2018)
18. Schmidhuber, J.: Evolutionary principles in self-referential learning, or on learning how to learn: The meta-meta... hook. Ph.D. Dissertation, Technische Universität München (1987)
19. Lin, J., et al.: Learning to transfer: unsupervised domain translation via meta-learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 11507–11514. AAAI Press, Palo Alto (2020)
20. Lu, H., et al.: Tasselnet: counting maize tassels in the wild via local counts regression network. Plant Methods 13, 79 (2017)
21. Arteta, C., et al.: Counting in the wild. In: Proceedings of the European Conference on Computer Vision, pp. 483–498. Springer, Cham (2016)
22. Finn, C., et al.: Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of the International Conference on Machine Learning, pp. 1126–1135. IEEE, Piscataway (2017)
23. Thrun, S., Pratt, L.: Learning to learn: Introduction and overview. In: Learning to Learn, pp. 3–17 (1998)
24. Guerrero-Gómez-Olmedo, R., et al.: Extremely overlapping vehicle counting. In: Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis, pp. 423–431 (2015)
25. Hsieh, M.-R., et al.: Drone-based object counting by fully convolutional neural networks for vehicle counting in city cameras. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3667–3676. IEEE, Piscataway (2017)
26. Bai, H., et al.: Crowd counting on images with scale variation and isolated clusters. In: Proceedings of the International Conference on Computer Vision Workshops (2019)
27. Zhang, Y., et al.: Single-image crowd counting via multi-column convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 359–367. IEEE, Piscataway (2020)
28. Ding, X., et al.: A deeply-recursive convolutional network for crowd counting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 359–367. IEEE, Piscataway (2020)
29. Marsden, M., et al.: People, penguins and petri dishes: adapting object counting models to new visual domains and object types without forgetting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8070–8079. IEEE, Piscataway (2018)
30. Zhu, P., et al.: Vision meets drones: A challenge. arXiv preprint, arXiv:1804.07437, 2018
31. Gao, J., et al.: C³ framework: an open-source pytorch code for crowd counting. arXiv preprint, arXiv:1907.02724, 2019

How to cite this article: Zan, C., et al.: Learn from Object Counting: Crowd Counting with Meta-learning. IET Image Process. 1–8 (2021). https://doi.org/10.1049/ipt2.12241