Class-agnostic Few-shot Object Counting

Shuo-Diao Yang, Hung-Ting Su, Winston H. Hsu, Wen-Chin Chen
National Taiwan University, Taipei, Taiwan

Abstract

Object counting which aims to calculate the number of total instances of the given class is a classic but crucial task that can be applied to many applications. Most of the prior works only focus on counting certain classes of objects such as people, cars, animals, etc. However, in recent years, there are lots of applications that need to get the count of the unseen class of objects such as a mechanical arm commanded to grab the novel object. In this paper, we present an effective object counting network, Class-agnostic Few-shot Object Counting Network (CFOCNet), that supports counting arbitrary classes of object unseen during training stage. Instead of counting a pre-defined class, our model is able to count instances based on input ***reference*** *images and reduces the huge cost of data collection, training and parameter tuning for each new object class. Our model utilizes not only the similarity between* ***query*** *image and* ***reference*** *images but self attending the* ***query*** *image to learn the self-repeatedness. Using a two-stream Resnet that matches features in different scales, our network can automatically learn to aggregate different scales of the matching scores. We evaluate our method on the subset of the COCO dataset that contains 80 classes of objects and many diverse scenes. In the experiments, our network outperforms other methods including detection and some previous works by a large margin. To the best of our knowledge, we are the first that mainly focuses on few-shot object counting in the class-agnostic manner.*

1. Introduction

Object counting has become a popular research area in recent years. There is a large number of works that propose different object counting methods to get the total amount of certain category such as people [2, 22, 27], cars [12], animals [19, 21], grape [24], etc. The application of counting objects is becoming increasingly popular in various aspects. To count cars, for example, one needs to first collect thousands, even tens of thousands of images with manually annotated positions of each car, followed by a learning process to find a desirable solution. These kinds of methods usually perform well in counting the particular class of objects. However, a counter can only count the certain class that it is trained on, unable to apply to other categories. Besides, collecting tremendous labeled data may not be always possible due to its cost and difficulty of annotating especially in some areas like medical and bioscience. Furthermore, the time consumed on training a model is also a concern. Although state-of-the-art models can output a fairly excellent result, most of them need to be trained for a long time, from one day to several weeks, depending on the network architecture.

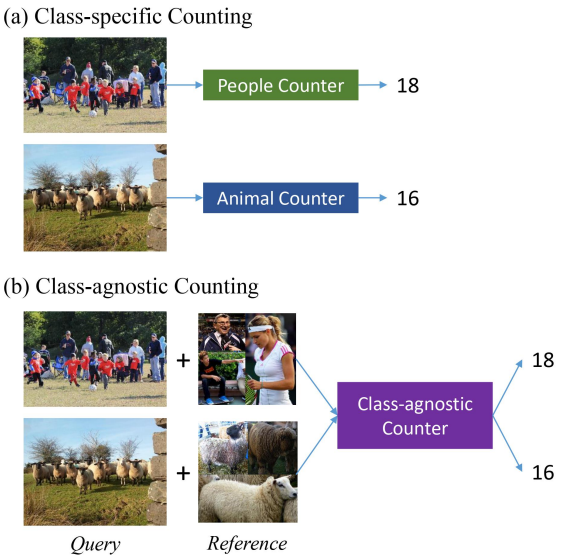

Figure 1. Class-specific counting versus class-agnostic counting. (a) In the scenario of class-specific counting, a counter is only responsible for counting a certain object that is seen in training. (b) In this paper we propose a class-agnostic counter that can count arbitrary object provided a *query* image and few *reference* images. Once trained, the counter can count **unseen** novel object without any retraining or fine-tuning.

There are several ways to get the total count of a certain object, including directly applying object detector on the input image [7, 10], learning a mapping from patches to a number [3], or summing over the predicted density map [2]. In this paper, we choose the last one due to its capability

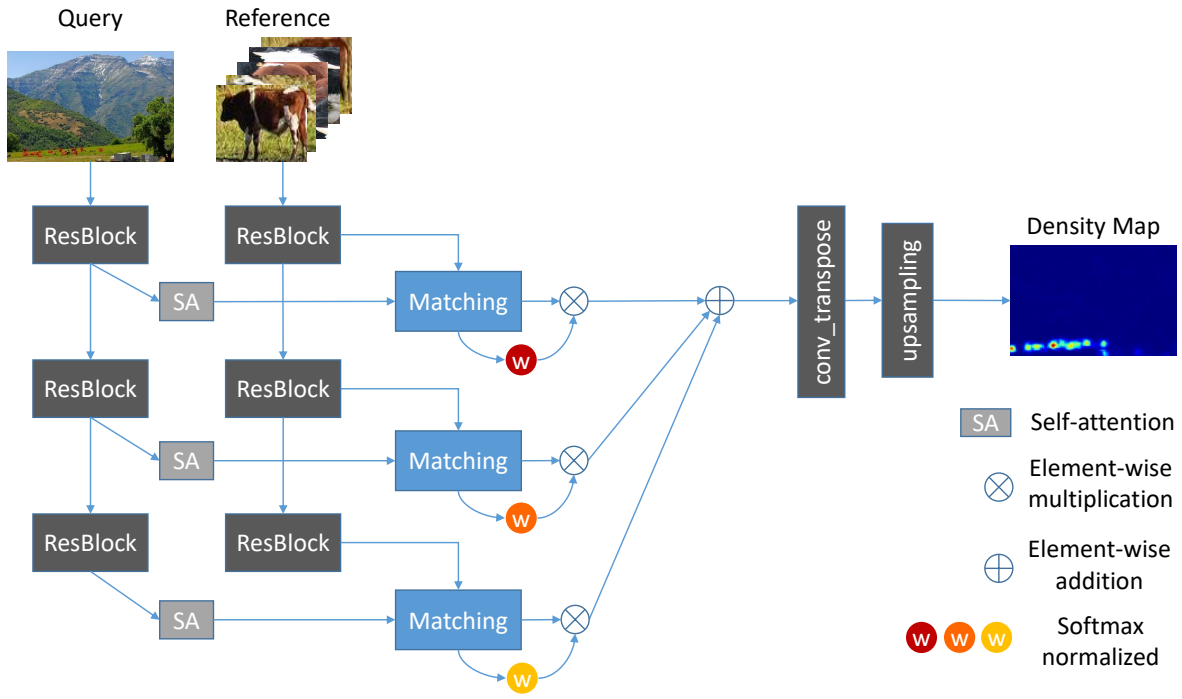

Figure 2. Overall architecture of the proposed method (CFOCNet). The input of this method is a *query* image and several *reference* images where the number of *reference* images needs not to be fixed (details can be found in section 3.2). Our network is mainly built upon two stream Resnet with matching mechanism (details in section 3.2.2) applied on the output of each Resblock followed by a learnable weighted fusion (details in section 3.2.3) that combines the matching score map of different scales. The final count is calculated by taking integration of the predicted density map.

to count those occluded, scale-varied and congested scenes like crowd counting methods [27].

To tackle those aforementioned problems, we introduce an effective Class-agnostic Few-shot Object Counting Network (CFOCNet) which can not only count arbitrary class of object but also requires only a few reference images. The input of our method is a *query* image and several *reference* images. To achieve class-agnostic manner, instead of learning a bunch of object classifier like Faster-RCNN [20], we match the *query* image and *reference* images in Resnet[11]-extracted feature space, provided none of category information, forcing our network to extract class-independent feature. Therefore, our model has only to be trained one time but has the ability to generalize to unseen classes.

Since there is no available dataset that aims at class-agnostic few-shot counting, we evaluate our method on a subset of COCO [16] dataset. During testing, categories of the object to be count are all **unseen** during training stage. The experiments show that our CFOCNet surpasses recent works such as One-Shot Instance Segmentation [18], which is mainly built on object detection method and Class-Agnostic Counting [17], which is similar to our method but has been fine-tuned on testing class. Ablation studies also show the effectiveness of each part of the proposed network architecture.

To summarize, our main contributions of this paper are as follows:

- Different from those object counters that are only designed for counting the specific category, we argue the importance of class-agnostic few-shot counting. To the best of our knowledge, there is no previous work that focuses on this scenario.
- We propose an effective network architecture that calculates the similarity of *query* image and *reference* images in different scales, which reduces the counting problem to a matching problem.
- Compared to similar works, our CFOCNet has the lowest MAE/MSE on the subset of the COCO dataset. Visualization (figure 4) also shows that our method can successfully predict various objects.

2. Related work

There is a large number of methods introduced in recent years that focus on few-shot learning and object counting. In this section, we discuss each task respectively.

2.1. Few-shot learning

Human is good at recognizing a novel class of object through only a few of examples. Few-shot learning aims to mimic human's ability of generalization that can rapidly apply a model to new classes not originally trained on. Formally speaking, in the few-shot scheme, C categories are sampled from the training set, each class contains K examples, and the model is required to classify those C categories from $C \times K$ examples. Such setting is denoted as *C-way, K-shot* learning problem.

[9] proposes a model-agnostic meta-learning algorithm that can be applied to various tasks trained with gradient-based method. This meta-learning algorithm is trained on various tasks so that it can solve novel tasks using just a small number of examples. [23] introduces prototypical networks for few-shot classification. Prototypical network computes the distance of each class on a metric space that can be used on classifying.

In contrast to typical few-shot learning methods which the number of input examples should be fixed, our network can receive an arbitrary number of examples during training and testing, resulting in flexible usage. Besides, prior few-shot learning works mostly focus on classification. In this work, to the best of our knowledge, we are the first one that mainly aims to counting problem.

2.2. Object counting

There are many works contributing to counting objects. Strategies used to compute the total count of the certain object can be mainly divided into two categories: detection method and density map method. Detection methods such as [12, 4] acquire the count of object by running a detection model. These kinds of methods can output satisfiable prediction in regular cases. However, in some extreme cases where instances of each object are occluded, overlapped and distorted like crowd counting, detection method may fail because of the aforementioned difficulties.

Instead of counting by detecting, density map based methods output a map representing the density distribution of input image, leading to a more fine-grained counting result. [5] proposes an image-level supervised density map estimation for object counting and shows its performance over state-of-the-art methods. [26] proposes a multi-view CNN crowd counting network which can make use of multiple camera views.

Nevertheless, these aforementioned works only focus on counting objects that have been seen during training stage. In this paper, we introduce a network capable of counting arbitrary **unseen** classes of objects in an agnostic manner.

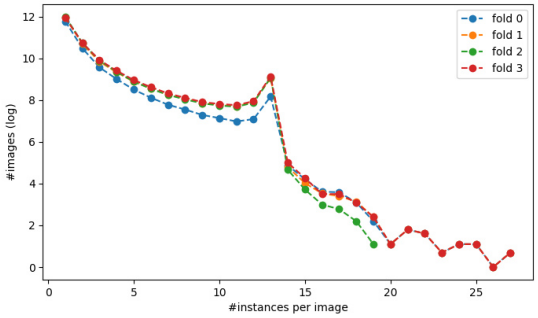

Figure 3. Distribution of the number of instances per image on 4-fold training data. Due to its extremely skewed distribution, we empirically choose images that have five or more instances of training categories to make the training process stable.

3. Method

In this section, we formally define the class-agnostic few-shot problem in section 3.1, followed by proposed network architecture in 3.2.

3.1. Problem definition

The problem definition of this paper is defined as follows: given a *query* image that contains several different objects and several *reference* images of the same category, our model should output a number that represents the number of instances of the *reference* images contained in the *query* image. The *query* image can consist of arbitrary objects which in this work is selected from the COCO dataset. Each *reference* image contains only one object of the desired category.

The number of *reference* images can be one or more, providing more *reference* images leading to better performance (details can be found at figure 5). If the number of *reference* images is decreased to one, the problem is degraded to one-shot learning. In order to input arbitrary number of *reference* images, we design a flexible network architecture to satisfy such circumstances which is discussed in section 3.2.

Besides few-shot manner, our method also aims at class-agnostic learning, which is, in other words, an ability to generalize to the unseen objects. During training stage, only *query* image and *reference* images are provided to our network, without any label information, forcing our network to extract class-independent information. In testing stage, although the categories of *reference* images does not appear in training data, our method can still estimate reasonable result.

COCO Dataset (one-shot) (≥5 instances)								
	fold 0		fold 1		fold 2		fold 3	
Method	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
Segment [18]	4.61±0.04	5.91±0.14	4.53±0.08	5.93±0.13	4.10±0.10	5.56±0.23	4.32±0.13	5.61±0.27
Cac [17]	5.01±0.04	5.92±0.04	4.22±0.03	5.66±0.51	3.97±0.08	4.98±0.09	3.77±0.14	4.73±0.19
CFOCNet	**3.18**±0.05	**4.04**±0.08	**3.69**±0.07	**4.71**±0.17	**3.24**±0.11	**4.40**±0.21	**3.19**±0.14	**4.40**±0.36

Table 1. Result on each of four folds conditioned on one *query* image (#instances≥5) and one *reference* image. We repeat each experiment 5 times and report the average MAE (Mean Absolute Error) and MSE (Mean Square Error) along with standard deviation. Our CFOCNet outperforms previous works by a large margin without any fine-tuning on testing categories.

COCO Dataset (five-shot) (≥5 instances)								
	fold 0		fold 1		fold 2		fold 3	
Method	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
Segment [18]	4.40±0.05	5.67±0.05	4.36±0.07	5.68±0.10	3.80±0.02	5.18±0.13	4.16±0.05	5.31±0.07
Cac [17]	3.63±0.03	4.58±0.05	4.07±0.04	6.13±0.23	3.05±0.08	4.04±0.09	3.46±0.06	4.64±0.15
CFOCNet	**2.98**±0.02	**3.74**±0.03	**3.46**±0.09	**4.53**±0.14	**3.03**±0.05	**4.04**±0.05	**2.82**±0.07	**3.77**±0.10

Table 2. Result on each of four folds conditioned on one *query* image (#instances≥5) and five *reference* image. We repeat each experiment 5 times and report the average MAE and MSE along with standard deviation. In this table, the number of *reference* images increases from one to five which provides the model more matching capability. Again, our CFOCNet outperforms previous works by a large margin without any fine-tuning on testing categories.

3.2. Network architecture

Inspired by [15] that shows great success in object tracking, our network mainly consists of a two-stream Resnet encoder that calculates the matching score in different scales followed by a decoder that generates the predicted density map. Encoder has two stream that extracts the feature of *query* image and *reference* images respectively, coupled with a correlation operation (section 3.2.2) to embed two branches' information. Decoder learns to fuse the score maps generated from encoder by a trainable weighted sum mechanism. Each component is discussed in the following section.

3.2.1 Resnet encoder

As shown in figure 2, the encoder has two streams, one is *query* stream and the other is *reference* stream. In this work, we use the first three blocks of Resnet-50 due to its powerful feature representation. The *query* stream is the standard Resnet that is composed of three Resblock, the output of each block denoted as follows:

$$Res_i^{query}, i \in \{1, 2, 3\} \tag{1}$$

where i represents the i-th Resblock's output. The *reference* stream is also the standard Resnet which has multiple input images, output of each block denoted as follows:

$$Res_{i,j}^{ref}, i \in \{1, 2, 3\}, j \in \{1, 2, ..., k\} \tag{2}$$

where k represents the number of *reference* images (*k-shot*). Then Res_i^{query} and $Res_{i,j}^{ref}$ are used as the input of next Resblock or the input of correlation operation.

3.2.2 Matching mechanism

Recent counting methods learn a detector per object category which shows satisfiable results in counting certain object class. However, in the class-agnostic setting, we should find a way that makes our model not be restricted in specific categories. Thus in our network, instead of providing label information into our network to train a class-specific detector, we leverage the matching mechanism widely used in traditional methods [8, 13].

To calculate the matchingness of *query* image and *reference* objects, we first aggregate the feature of *reference* images extracted from encoder by pooling operation along the j dimension:

$$Res_i^{ref} = max\_pool(Res_{i,j}^{ref}), i \in \{1, 2, 3\} \tag{3}$$

where providing more *reference* images leads to better performance. Then we can calculate the matching score between R_i^{query} and R_i^{ref} by:

$$input = self\_attn(Res_i^{query}) \tag{4}$$

$$kernel = max\_pool(Res_i^{ref}, r) \tag{5}$$

$$M_i = Conv(input, kernel), i \in \{1, 2, 3\} \tag{6}$$

where max_pool here denotes the max pooling operation along the spatial dimension (height and width) to reduce the spatial resolution of extracted feature map to $r \times r$ due

COCO Dataset (one-shot) ($\geq$1 instances)								
	fold 0		fold 1		fold 2		fold 3	
Method	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
Segment [18]	2.91±0.02	4.20±0.03	2.47±0.04	3.67±0.05	2.64±0.02	3.79±0.06	2.82±0.02	4.09±0.03
Cac [17]	2.97±0.02	4.02±0.02	3.39±0.03	4.56±0.11	3.00±0.04	3.94±0.04	3.30±0.03	4.40±0.09
CFOCNet	**2.24**±0.01	**3.50**±0.03	**1.78**±0.02	**2.90**±0.03	**2.66**±0.03	**3.82**±0.07	**2.16**±0.03	**3.27**±0.07

Table 3. Result on each of four folds conditioned on one query image (#instances$\geq$1) and one *reference* image evaluated on the full set of each fold. The experiment is repeated 5 times to reduce the randomness. Our method outperforms others by a large margin in all folds of the full set.

to computational cost. We empirically set $r = 4$ to achieve the best result. $Conv$ denotes the convolution operation that computes the similarity of each spatial location of feature map from *query* stream. Since the $kernel$ here is derived from the feature map of *reference* stream, it does not require more trainable variables, resulting in a parameterless convolution. In addition, to calculate the matchingness between two branches, self-attention mechanism [25] is also applied on *query* branch to encourage the model focusing on self-similarity incurred by repeatedness of the same objects. After above matching operation, now we have three different maps of matching score M_1, M_2, M_3 representing the similarities in various scales.

3.2.3 Density map decoder

Due to the scale variation of *query* image ranging from bus to donut, a scale-aware fusing mechanism is required. In this work, we propose a learnable weighted sum fusing mechanism to let the model automatically attend on the desired scale according to the matching score generated from the encoder.

To calculate the weighted sum of three matching score maps, we first compute the weight of each matching score maps by:

$$S_i = Sum(Conv(M_i)), i \in \{1, 2, 3\} \tag{7}$$

$$W = Softmax(S), W \in R^3 \tag{8}$$

where $Conv$ denotes the 1×1 convolution that reduces the channels to be one and Sum is summing operation that produces a scalar. $Softmax$ is used to normalize the weight corresponding to different matching score maps. Then the fused matching score map can be computed as:

$$F = \sum_{i=1}^{3} W_i \times M_i, i \in \{1, 2, 3\} \tag{9}$$

Finally, because the spatial resolution of F is 1/8 of original image, the output density map is simply generated by applying transpose convolution and bilinear upsampling.

4. Experiments

In this section, we first introduce the dataset used for training and testing in section 4.1, followed by implementation details in section 4.2. Evaluation protocal and main result are discussed in section 4.3 and section 4.4 respectively.

4.1. Dataset

Since there is no available dataset that aims at evaluating the generalization of the class-agnostic setting, we manually choose a subset of COCO [16] dataset to meet such requirement. The COCO dataset is widely used in object detection, instance segmentation and many other works due to its rich object categories (80 classes) and various scenes. In this paper, we follow the setting in [18] that uses 4-fold validation which split 80 object categories into 60/20 for training and testing. Therefore the 20 categories of objects in testing stage are unseen in training stage which can benchmark the model ability of generalization to novel classes.

4.2. Implementation details

4.2.1 Training example generation

A training example consists of one *query* image and *k reference* images. *Query* image is randomly sampled from the dataset which one or more training class is contained in. To make the training converge faster, we further choose the images that have five or more instances of training categories that can provide more updating signal to the model (distribution in figure 3). Once the *query* image is selected, it may have several objects of training classes. Hence we again randomly sample one desired class contained in *query* image to decide the class of *reference* images.

To get the *reference* images of the desired class, we crop all the instances of each object from training set, saving them to the disk indexed by their category index for later usage. Thus, in the training stage, we can rapidly sample *k* image of the desired category. In this paper, we set $k = 5$ for all experiments.

COCO Dataset (five-shot) (≥1 instances)								
	fold 0		fold 1		fold 2		fold 3	
Method	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
Segment [18]	2.86±0.01	4.08±0.01	2.46±0.02	3.58±0.03	2.65±0.01	3.73±0.02	2.91±0.01	4.08±0.01
Cac [17]	3.93±0.02	5.12±0.04	3.67±0.02	4.83±0.05	4.33±0.02	5.14±0.40	3.31±0.03	4.26±0.05
CFOCNet	**2.15**±0.01	**3.43**±0.02	**1.73**±0.01	**2.79**±0.02	**2.39**±0.02	**3.39**±0.04	**1.90**±0.01	**2.80**±0.03

Table 4. Result on each of four folds conditioned on one query image (#instances≥1) and five *reference* images evaluated on the full set of each fold. Our network's MAE/MSE decreases as the number of *reference* images increases (cf. table 3) while others have no performance gain, proving the effective design of our network.

4.2.2 Ground truth generation

The output of the network is a density map that represents the distribution of *reference* class. Since the COCO dataset has only bounding box annotation, converting box annotation to density map is required. We follow previous work [22] that generates the ground truth density map as follows. First, we obtain a binary map:

$$B_{i,j} = \begin{cases} 1 & if\ (i,j)\ is\ annotated \\ 0 & else \end{cases} \tag{10}$$

The $annotated$ here means that (i, j) is the center of the bounding box. Then we convolve B by a Gaussian kernel with standard deviation empirically set to 10.

4.2.3 Network settings

During training, the *query* image is randomly cropped to 256×256 with randomly flipped of probability 0.5, while *reference* images are resized to 64×64 with padding to keep the aspect ratio. Besides, the channel of feature maps used to compute the matching score is reduced to 256 by applying a 1×1 convolution. In this paper, we let $k = 5$, resulting in a 5-shot matching problem.

The network is supervised by the standard L2 loss that compares the difference between two images, defined as:

$$L_E = \frac{1}{N}\sum_{i=1}^{N}\|P_i - GT_i\|^2, \tag{11}$$

supposed there are N pixels in a given image where P_i is the prediction of our model and GT_i is ground truth density map. In addition, SSIM loss is used as well since it shows the advantage on catching local pattern consistency [2], defined as:

$$SSIM = \frac{(2\mu_p\mu_{gt} + c_1)(2\sigma_{p,gt} + c_2)}{(\mu_p^2 + \mu_{gt}^2 + c_1)(\sigma_p^2 + \sigma_{gt}^2 + c_2)} \tag{12}$$

where μ_p and μ_{gt} are mean and σ_p and σ_{gt} are standard deviation of prediction map and ground truth density map, $\sigma_{p,gt}$ denotes the covariance, c_1 and c_2 are small value to avoid division by zero. Aggregating each pixel of prediction, SSIM loss is defined as:

$$L_{SSIM} = 1 - \frac{1}{N}\sum SSIM(\mathbf{x}), \tag{13}$$

where $\mathbf{x}$ is the position of the map. Final loss is calculated by fusing these two loss functions:

$$L = L_E + \lambda L_{SSIM}, \tag{14}$$

where λ is used to balance two loss functions. In this paper we set λ to $1e-5$.

Our network is trained from scratch without any pre-trained weight. Adam [14] optimizer is used because it shows faster converging speed in many tasks, learning rate is set to $1e-4$ without any scheduling and batch size is set to 20. All the experiments are implemented by Tensorflow [1] framework and nVidia V100 GPUs.

4.3. Evaluation protocol

To evaluate the effectiveness of the proposed method, we measure the performance by MAE (Mean Absolute Error) and MSE (Mean Square Error) that are commonly applied on recent works [12, 17], defined as follows:

$$MAE = \frac{1}{M}\sum_{i=1}^{M}|Pred_i - Cnt_i| \tag{15}$$

$$MSE = \sqrt{\frac{1}{M}\sum_{i=1}^{M}|Pred_i - Cnt_i|^2} \tag{16}$$

where M is total images, $Pred_i$ is prediction count integrated from predicted density map and Cnt_i is ground truth count obtained by taking integral of predicted density map.

Since our network is fully convolutional, which can receive arbitrary size of the input image, different from training stage that uses a patch-based input, we empirically find out that during testing directly inputting the whole *query* image to the network leads to better performance.

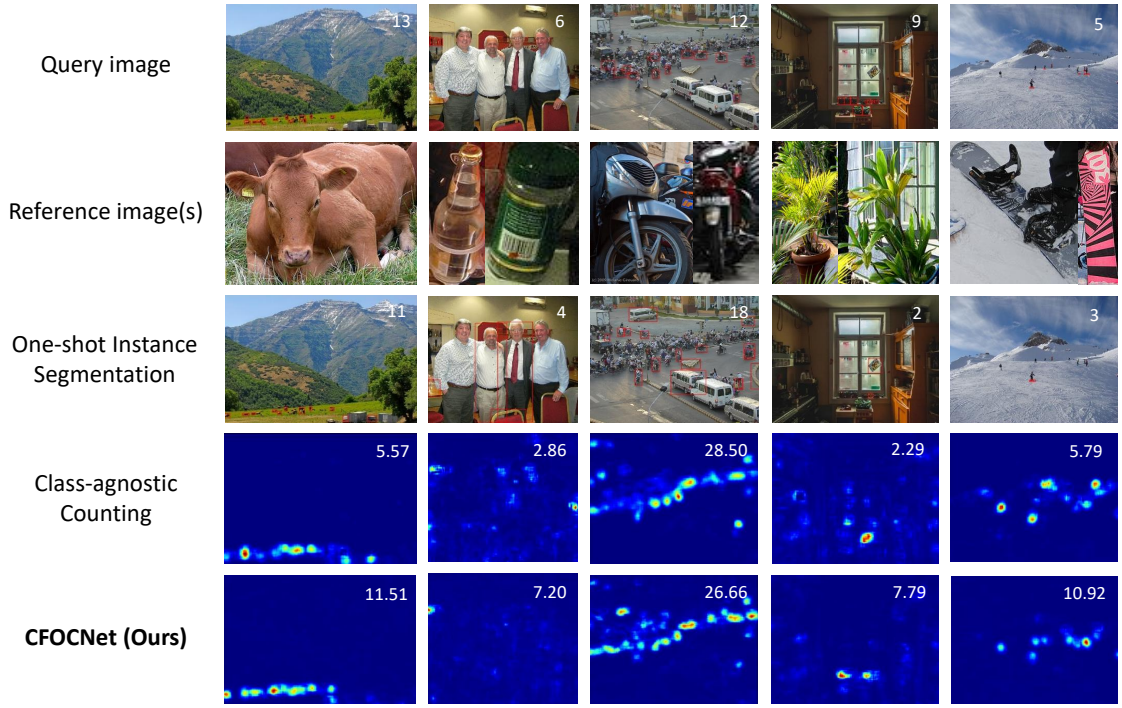

Figure 4. Visualization of our method versus previous works. The first row is the *query* image, the second row is the *reference* images sampled from testing set, and the third, fourth and fifth row shows the predicted result of [18], [17] and CFOCNet (ours). The number in the upper-right corner represents the count. It is obvious that our method has finer prediction than similar works. The analysis can be found in section 4.4.3.

4.4. Results

4.4.1 Baselines

We compare our method with two similar works, One-shot Instance Segmentation [18] and Class-agnostic Counting [17]. Each of them is briefly described in the following section.

One-shot Instance Segmentation: This paper proposes a Siamese Mask R-CNN network to address the problem of one/few-shot object detection and segmentation. To be compared with our method whose output is a scalar representing the number of instances of the given class of the object, we choose to use the bounding box part of this model. By manual thresholding the confidence score ([0.70, 0.99]) of the bounding boxes, we choose the best one that results in the lowest MAE in each fold respectively. The result of this method is evaluated on the checkpoint released on its Github page.

Class-agnostic Counting: This paper also introduces a siamese-like network to predict a density map that indicates the distribution of the interesting object. It first learns a GMN (Generic Matching Network) on ImageNet [6], followed by a fine-tuning process on the target dataset. Different from this method, our proposed model is only trained on the COCO dataset without any fine-tuning on target categories. Besides, due to the original network architecture only capable of receiving only one *reference* image, we slightly modify the internal layer to make it able to process arbitrary *reference* images along with *query* image at the same time. To be fair, we train this model on the COCO dataset from scratch according to the 4-fold splits.

4.4.2 Comparison

Since the *reference* images are randomly chosen from testing set, the result of each experiment may fluctuate. To alleviate the uncertainty caused by randomness, for each experiment we repeat five times and report the average MAE/MSE and standard deviation.

First, we evaluate each fold on those images having more than five instances of objects which meet the training condition. As shown in table 1, our CFOCNet (last row) reaches the lowest MAE/MSE on all folds, significantly outperforming similar works. Table 2 shows the performance where number of *reference* images increases to five. It is obvious that our model can benefit from having more *reference* images, indicating the power of feature aggregating as mentioned in section 3.2.2. The effect of the amount of *reference* images provided to the network is described in

Ablation Studies (one-shot) (≥1 instances)		
Method	MAE	MSE
w/o weighted fusion	2.40±0.01	3.29±0.02
w/o self-attn	2.45±0.02	3.90±0.08
CFOCNet	**2.16**±0.03	**3.27**±0.07

Table 5. Ablation studies of removing some part of our network conducted on fold 3. As shown in this table, removing each of the components resulting in larger MAE/MSE. Details can be found on section 4.4.4.

section 4.4.4.

Secondly, we test our model on the full set of each fold, that is, *query* image not restricted to the number of object instances. In this case, there are some images that contain less than 5 instances which are not seen during training stage. Again our method performs better than others, indicating that our model can generalize to the distribution that is different from training (table 3). Besides, our network performs better than using only one *reference* image (table 4), showing the effectiveness of network architecture.

4.4.3 Visualization

In addition to the quantitative result, we also visualize the result of the output of our network and previous works. Figure 4 shows the results that are most accurate among all folds. The first row is the *query* image with red bounding boxes annotating the instances of ground truth objects. The second row shows some examples of *reference* images. We visualize the prediction of One-shot Instance Segmentation [18] by manual thresholding the confidence score which resulting in the lowest MAE. Visualization result shows that [18] can successfully regress the bounding box of objects, but with wrong object class. Class-agnostic Counting [17] outputs a similar density map to our CFOCNet. However, the prediction of [17] consists of too much false positive, especially in those regions with similar shape or texture.

4.4.4 Ablation studies

To verify the effectiveness of components of our architecture, we conduct ablation studies on 1) remove the weighted fusion of three matching score map, as described in equation 7 and 8, which three maps are directly added instead and 2) remove the self-attention layer, as described in equation 4. Table 5 shows the result of removing each of the aforementioned components, leading to a performance drop on both MAE/MSE, indicating the importance of the weighted fusion and self-attention mechanism.

Furthermore, we conduct an ablation study on the number of *reference* images, increasing from one to five. As shown in figure 5, providing more *reference* images to our network leads to better performance, which is also a proof of the benefit gained from the pooling operation of *reference* branch.

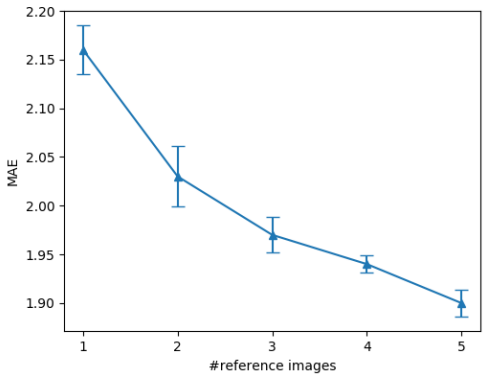

Figure 5. Ablation study on the number of *reference* images providing to the network, tested on fold 3. We repeat this experiment five times and report the average MAE along with standard deviation. It is clear that providing more *reference* images leads to lower MAE.

5. Conclusion

In this paper, we tackle the problem of class-agnostic few-shot learning on object counting tasks. We propose a network, called CFOCNet, that computes the number of instances of certain object class by providing one or more *reference* images to the network. To achieve the class-agnostic manner, we cast the counting problem into a matching problem in order to force the model learning the class-independent feature so that in the testing stage the model can generalize to such unseen objects. The experiments also show that our proposed model can reach lower MAE/MSE in each fold of the COCO dataset.

During doing experiments, we also find out that the object instances that appeared in the dataset are not well-annotated. For example, there are many images containing so large amount of instances that not all of them are annotated. Therefore the ground truth counts are often underestimated, leading to the increase in MAE/MSE. Despite this, our model still shows its matching ability on the unannotated regions, as shown in figure 4.

6. Acknowledgement

This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 109-2634-F-002-032. We benefit from NVIDIA grants and DGX-1 AI Supercomputer and are grateful to the National Center for High-performance Computing.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

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

[3] Antoni B Chan and Nuno Vasconcelos. Bayesian poisson regression for crowd counting. In *2009 IEEE 12th international conference on computer vision*, pages 545–551. IEEE, 2009.

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

[5] Hisham Cholakkal, Guolei Sun, Fahad Shahbaz Khan, and Ling Shao. Object counting and instance segmentation with image-level supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12397–12405, 2019.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[7] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE transactions on pattern analysis and machine intelligence*, 34(4):743–761, 2011.

[8] M-P Dubuisson and Anil K Jain. A modified hausdorff distance for object matching. In *Proceedings of 12th international conference on pattern recognition*, volume 1, pages 566–568. IEEE, 1994.

[9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

[10] Weina Ge and Robert T Collins. Marked point processes for crowd counting. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2913–2920. IEEE, 2009.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] Meng-Ru Hsieh, Yen-Liang Lin, and Winston H Hsu. Drone-based object counting by spatially regularized regional proposal network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4145–4153, 2017.

[13] Anil K. Jain, Yu Zhong, and Sridhar Lakshmanan. Object matching using deformable templates. *IEEE Transactions on pattern analysis and machine intelligence*, 18(3):267–278, 1996.

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[15] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4282–4291, 2019.

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[17] Erika Lu, Weidi Xie, and Andrew Zisserman. Class-agnostic counting. In *Asian Conference on Computer Vision*, pages 669–684. Springer, 2018.

[18] Claudio Michaelis, Ivan Ustyuzhaninov, Matthias Bethge, and Alexander S. Ecker. One-shot instance segmentation. *CoRR*, abs/1811.11507, 2018.

[19] Mohammad Sadegh Norouzzadeh, Anh Nguyen, Margaret Kosmala, Alexandra Swanson, Meredith S Palmer, Craig Packer, and Jeff Clune. Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *Proceedings of the National Academy of Sciences*, 115(25):E5716–E5725, 2018.

[20] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[21] Alberto Rivas, Pablo Chamoso, Alfonso González-Briones, and Juan Corchado. Detection of cattle using drones and convolutional neural networks. *Sensors*, 18(7):2048, 2018.

[22] Miaojing Shi, Zhaohui Yang, Chao Xu, and Qijun Chen. Revisiting perspective information for efficient crowd counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7279–7288, 2019.

[23] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017.

[24] Laura Zabawa, Anna Kicherer, Lasse Klingbeil, Andres Milioto, Reinhard Topfer, Heiner Kuhlmann, and Ribana Roscher. Detection of single grapevine berries in images using fully convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[25] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.

[26] Qi Zhang and Antoni B Chan. Wide-area crowd counting via ground-plane density maps and multi-view fusion cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8297–8306, 2019.

[27] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 589–597, 2016.