Learning To Count Everything

Viresh Ranjan[1] Udbhav Sharma[1] Thu Nguyen[2] Minh Hoai[1,2]
[1]Stony Brook University, USA
[2]VinAI Research, Hanoi, Vietnam

Abstract

Existing works on visual counting primarily focus on one specific category at a time, such as people, animals, and cells. In this paper, we are interested in counting everything, that is to count objects from any category given only a few annotated instances from that category. To this end, we pose counting as a few-shot regression task. To tackle this task, we present a novel method that takes a query image together with a few exemplar objects from the query image and predicts a density map for the presence of all objects of interest in the query image. We also present a novel adaptation strategy to adapt our network to any novel visual category at test time, using only a few exemplar objects from the novel category. We also introduce a dataset of 147 object categories containing over 6000 images that are suitable for the few-shot counting task. The images are annotated with two types of annotation, dots and bounding boxes, and they can be used for developing few-shot counting models. Experiments on this dataset shows that our method outperforms several state-of-the-art object detectors and few-shot counting approaches. Our code and dataset can be found at `https://github.com/cvlab-stonybrook/LearningToCountEverything`.

1. Introduction

Humans can count objects from most of the visual object categories with ease, while current state-of-the-art computational methods [29, 48, 55] for counting can only handle a limited number of visual categories. In fact, most of the counting neural networks [4, 48] can handle a single category at a time, such as people, cars, and cells.

There are two major challenges preventing the Computer Vision community from designing systems capable of counting a large number of visual categories. First, most of the contemporary counting approaches [4, 48, 55] treat counting as a supervised regression task, requiring thousands of labeled images to learn a fully convolutional regressor that maps an input image to its corresponding density map, from which the estimated count is obtained by summing all the density values. These networks require dot annotations for millions of objects on several thousands of training images, and obtaining this type of annotation is a costly and laborious process. As a result, it is difficult to scale these contemporary counting approaches to handle a large number of visual categories. Second, there are not any large enough unconstrained counting datasets with many visual categories for the development of a general counting method. Most of the popular counting datasets [14–16, 43, 49, 55] consist of a single object category.

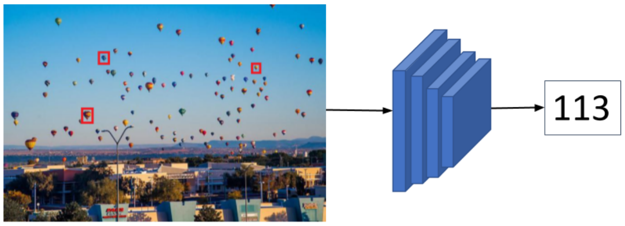


Figure 1: **Few-shot counting—the objective of our work.** Given an image from a novel class and a few exemplar objects from the same image delineated by bounding boxes, the objective is to count the total number of objects of the novel class in the image.

In this work, we address both of the above challenges. To handle the first challenge, we take a detour from the existing counting approaches which treat counting as a typical fully supervised regression task, and pose counting as a few shot regression task, as shown in Fig. 1. In this few-shot setting, the inputs for the counting task are an image and few examples from the same image for the object of interest, and the output is the count of object instances. The examples are provided in the form of bounding boxes around the objects of interest. In other words, our few shot counting task deals with counting instances within an image which are similar to the exemplars from the same image. Following the convention from the few-shot classification task [9, 20, 46], the classes at test time are completely different from the ones seen during training. This makes few-shot counting very different from the typical counting task, where the training and test classes are the same. Unlike the typical counting task, where hundreds [55] or thousands [16] of labeled examples are available for training, a few-shot counting method needs to generalize to completely novel classes using only the input image and a few exemplars.

We propose a novel architecture called Few Shot Adaptation and Matching Network (FamNet) for tackling the few-shot counting task. FamNet has two key components: 1) a feature extraction module, and 2) a density prediction module. The feature extraction module consists of a general feature extractor capable of handling a large number of visual categories. The density prediction module is designed to be agnostic to the visual category. As will be seen in our experiments, both the feature extractor and density prediction modules can already generalize to the novel categories at test time. We further improve the performance of FamNet by developing a novel few-shot adaptation scheme at test time. This adaptation scheme uses the provided exemplars themselves and adapts the counting network to them with a few gradient descent updates, where the gradients are computed based on two loss functions which are designed to utilize the locations of the exemplars to the fullest extent. Empirically, this adaptation scheme improves the performance of FamNet.

Finally, to address the lack of a dataset for developing and evaluating the performance of few-shot counting methods, we introduce a medium-scale dataset consisting of more than 6000 images from 147 visual categories. The dataset comes with dot and bounding box annotations, and is suitable for the few-shot counting task. We name this dataset Few-Shot Counting-147 (FSC-147).

In short, the main contributions of our work are as follows. First, we pose counting as a few-shot regression task. Second, we propose a novel architecture called FamNet for handling the few-shot counting task, with a novel few-shot adaptation scheme at test time. Third, we present a novel few-shot counting dataset called FSC-147, comprising of over 6000 images with 147 visual categories.

2. Related Works

In this work, we are interested in counting objects of interest in a given image with a few labeled examples from the same image. Most of the previous counting methods are for specific types of objects such as people [2, 5, 6, 23, 26, 27, 29, 32–34, 39, 42, 47, 50, 54, 55], cars [30], animals [4], cells [3, 18, 53], and fruits [31]. These methods often require training images with tens of thousands or even millions of annotated object instances. Some of these works [34] tackle the issue of costly annotation cost to some extent by adapting a counting network trained on a source domain to any target domain using labels for only few informative samples from the target domain. However, even these approaches require a large amount of labeled data in the source domain.

The proposed FamNet works by exploiting the strong similarity between a query image and the provided exemplar objects in the image. To some extent, it is similar the decade-old self-similarity work of Shechtman and Irani [41]. Also related to this idea is the recent work of Lu and Zisserman[28], who proposed a Generic Matching Network (GMN) for class-agnostic counting. GMN was pre-trained with tracking video data, and it had an explicit adaptation module to adapt the network to an image domain of interest. GMN has been shown to work well if several dozens to hundreds of examples are available for adaptation. Without adaptation, GMN does not perform very well on novel classes, as will be seen in our experiments.

Related to few-shot counting is the few-shot detection task (e.g., [8, 17]), where the objective is to learn a detector for a novel category using a few labeled examples. Few-shot counting differs from few-shot detection in two primary aspects. First, few-shot counting requires dot annotations while detection requires bounding box annotations. Second, few-shot detection methods can be affected by severe occlusion whereas few-shot counting is tackled with a density estimation approach [22, 55], which is more robust towards occlusion than the detection-then-counting approach because the density estimation methods do not have to commit to binarized decisions at an early stage. The benefits of the density estimation approach has been empirically demonstrated in several domains, especially for crowd and cell counting.

Also related to our work is the task of few-shot image classification [9, 19, 21, 35, 40, 46]. The few-shot classification task deals with classifying images from novel categories at test time, given a few training examples from these novel test categories. The Model Agnostic Meta Learning (MAML) [9] based few-shot approach is relevant for our few-shot counting task, and it focuses on learning parameters which can adapt to novel classes at test time by means of few gradient descent steps. However, MAML involves computing second order derivatives during training which makes it expensive, even more so for the pixel level prediction task of density map prediction being considered in our paper. Drawing inspiration from these works, we propose a novel adaptation scheme which utilizes the exemplars available at test time and performs a few steps of gradient descent in order to adapt FamNet to any novel category. Unlike MAML, our training scheme does not require higher order gradients at training time. We compare our approach with MAML, and empirically show that it leads to better performance and is also much faster to train.

3. Few-Shot Adaptation & Matching Network

In this section, we describe the proposed FamNet for tackling the few-shot counting task.

3.1. Network architecture

Fig. 2 depicts the pipeline of FamNet. The input to the network is an image $X \in \Re^{H \times W \times 3}$ and a few exemplar

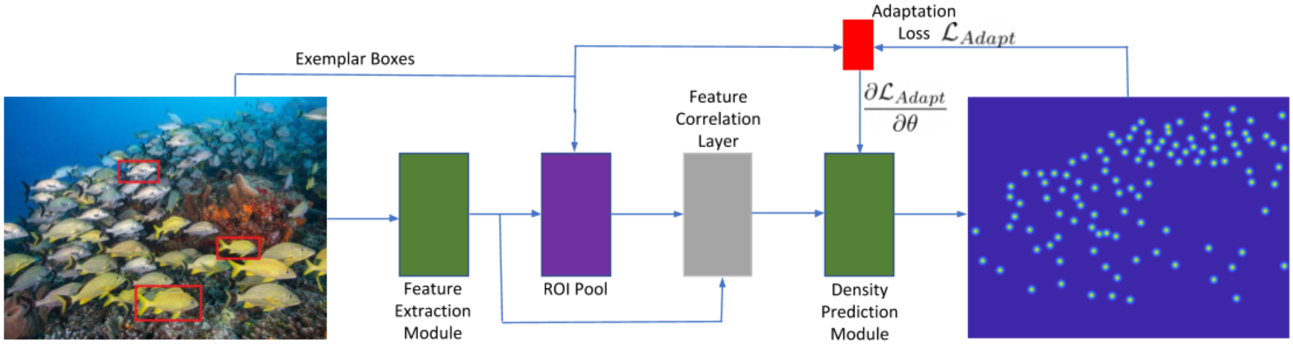

Exemplar Boxes
Adaptation Loss $\mathcal{L}_{Adapt}$
$\frac{\partial \mathcal{L}_{Adapt}}{\partial \theta}$
Feature Correlation Layer
Feature Extraction Module
ROI Pool
Density Prediction Module

Figure 2: **Few-shot adaptation & matching Network** takes as input the query image along with few bounding boxes depicting the object of interest, and predicts the density map. The count is obtained by summing all the pixel values in the density map. The adaptation loss is computed based on the bounding box information, and the gradients from this loss are used to update the parameters of the density prediction module. The adaptation loss is only used during test time.

bounding boxes depicting the object to be counted from the same image. The output of the network is the predicted density map $Z \in \Re^{H \times W}$, and the count for the object of interest is obtained by summing over all density values.

FamNet consists of two key modules: 1) a multi-scale feature extraction module, and 2) a density prediction module. We design both of these modules so that they can handle novel categories at test time. We use an ImageNet-pretrained network [12] for the feature extraction, since such networks can handle a broad range of visual categories. The density prediction module is designed to be agnostic to the visual categories. The multi-scale feature extraction module consists of the first four blocks from a pre-trained ResNet-50 backbone [12] (the parameters of these blocks are frozen during training). We represent an image by the convolutional feature maps at the third and fourth blocks. We also obtain the multi-scale features for an exemplar by performing ROI pooling on the convolutional feature maps from the third and fourth Resnet-50 blocks.

To make the density prediction module agnostic to the visual categories, we do not use the features obtained from the feature extraction module directly for density prediction. Instead, we only use the correlation map between the exemplar features and image features as the input to the density prediction module. To account for the objects of interest at different scales, we scale the exemplar features to different scales, and correlate the scaled exemplar features with the image features to obtain multiple correlation maps, one for each scale. For all of our experiments, we use the scales of 0.9 and 1.1, along with the original scale. The correlation maps are concatenated and fed into the density prediction module. The density prediction module consists of five convolution blocks and three upsampling layers placed after the first, second, and third convolution layers. The last layer is a 1×1 convolution layer, which predicts the 2D density map. The size of the predicted density map is the same as the size of the input image.

3.2. Training

We train the FamNet using the training images of our dataset. Each training image contains multiple objects of interest, but only the exemplar objects are annotated with bounding boxes and the majority of the objects only have dot annotations. It is, however, difficult to train a density estimation network with the training loss that is defined based on the dot annotations directly. Most existing works for visual counting, especially for crowd counting [55], convolve the dot annotation map with a Gaussian window of a fixed size, typically 15×15, to generate a smoothed target density map for training the density estimation network.

Our dataset consists of 147 different categories, where there is huge variation in the sizes of the objects. Therefore, to generate the target density map, we use Gaussian smoothing with adaptive window size. First, we use dot annotations to estimate the size of the objects. Given the dot annotation map, where each dot is at an approximate center of an object, we compute the distance between each dot and its nearest neighbor, and average these distances for all the dots in the image. This average distance is used as the size of the Gaussian window to generate the target density map. The standard deviation of the Gaussian is set to be a quarter of the window size.

To train FamNet, we minimize the mean squared error between the predicted density map and the ground truth density map. We use Adam optimizer with a learning rate of 10^{-5}, and batch size of 1. We resize each image to a fixed height of 384, and the width is adjusted accordingly to preserve the aspect ratio of the original image.

3.3. Test-time adaptation

Since the two modules of the FamNet are not dependent on any object categories, the trained FamNet can already be used for counting objects from novel categories given a few exemplars. In this section, we describe a novel approach to adapt this network to the exemplars, further improving the accuracy of the estimated count. The key idea is to harness the information provided by the locations of the exemplar bounding boxes. So far, we have only used the bounding boxes of the exemplars to extract appearance features of the exemplars, and we have not utilized their locations to the full extent.

Let B denote the set of provided exemplar bounding boxes. For a bounding box $b \in B$, let Z_b be the crop from the density map Z at location b. To harness the extra information provided by the locations of the bounding boxes B, we propose to consider the following two losses.

Min-Count Loss. For each exemplar bounding box b, the sum of the density values within Z_b should be at least one. This is because the predicted count is taken as the sum of predicted density values, and there is at least one object at the location specified by the bounding box b. However, we cannot assert that the sum of the density values within Z_b to be exactly one, due to possible overlapping between b and other nearby objects of interest. This observation leads to an inequality constraint: $||Z_b||_1 \geq 1$, where $||Z_b||_1$ denotes the sum of all the values in Z_b. Given the predicted density map and the set of provided bounding boxes for the exemplars, we define the following Min-Count loss to quantify the amount of constraint violation:

$$\mathcal{L}_{MinCount} = \sum_{b \in B} \max(0, 1 - ||Z_b||_1). \qquad (1)$$

Perturbation Loss. Our second loss to harness the positional information provided by the exemplar bounding boxes is inspired by the success of tracking algorithms based on correlation filter [13, 44, 51]. Given the bounding box of an object to track, these algorithms learn a filter that has highest response at the exact location of the bounding box and lower responses at perturbed locations. The correlation filter can be learned by optimizing a regression function to map from a perturbed location to a target response value, where the target response value decreases exponentially as the perturbation distance increases, usually specified by a Gaussian distribution.

In our case, the predicted density map Z is essentially the correlation response map between the exemplars and the image. To this end, the density values around the location of an exemplar should ideally look like a Gaussian. Let $G_{h \times w}$ be the 2D Gaussian window of size $h \times w$. We define the perturbation loss as follows:

$$\mathcal{L}_{Per} = \sum_{b \in B} ||Z_b - G_{h \times w}||_2^2. \qquad (2)$$

The combined adaptation Loss. The loss used for test-time adaptation is the weighted combination of the Min-Count loss and the Perturbation loss. The final test time adaptation loss is given as

$$\mathcal{L}_{Adapt} = \lambda_1 \mathcal{L}_{MinCount} + \lambda_2 \mathcal{L}_{Per}, \qquad (3)$$

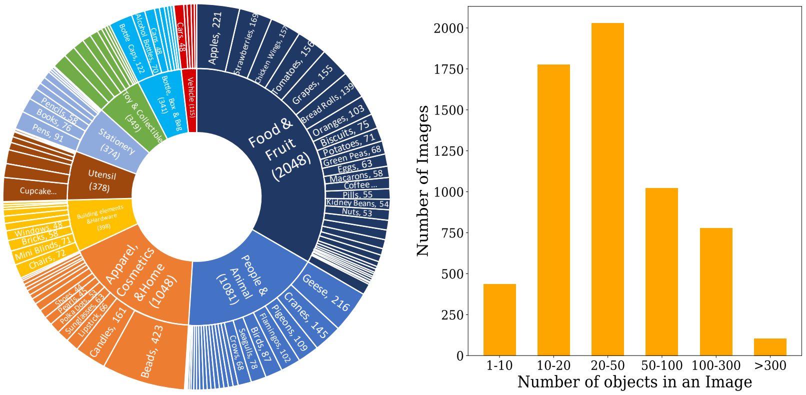
where λ_1 and λ_2 are scalar hyper parameters. At test time, we perform 100 gradient descent steps for each test image, and optimize the joint loss presented in Eq. (3). We use the learning rate 10^{-7}. The values for λ_1 and λ_2 are 10^{-9} and 10^{-4} respectively. The learning rate, the number of gradient steps, λ_1, and λ_2, are tuned based on the performance on the validation set. The values of λ_1, and λ_2 seem small, but this is necessary to make the adaptation loss to have similar magnitude to the training loss. Even though the training loss is not used for test time adaptation, it is important for the losses and their gradients to have similar magnitudes. Otherwise, the gradient update steps of the adaptation process will either do nothing or move away far from the parameters learned during training.

Note that the adaptation loss is only used at test time. During training of FamNet, this loss is redundant because the proposed training loss, based on mean squared errors computed over all pixel locations, already provides stronger supervision signal than the adaptation loss.

4. The FSC-147 Dataset

To train the FamNet, we need a dataset suitable for the few-shot counting task, consisting of many visual categories. Unfortunately, existing counting datasets are mostly dedicated for specific object categories such as people, cars, and cells. Meanwhile, existing multi-class datasets do not contain many images that are suitable for visual counting. For example, although some images from the COCO dataset [25] contains multiple instances from the same object category, most of the images do not satisfy the conditions of our intended applications due to the small number of object instances or the huge variation in pose and appearance of the object instances in each image.

Since there was no dataset that was large and diverse enough for our purpose, we collected and annotated images ourselves. Our dataset consists of 6135 images across a diverse set of 147 object categories, from kitchen utensils and office stationery to vehicles and animals. The object count in our dataset varies widely, from 7 to 3731 objects, with an average count of 56 objects per image. In each image, each object instance is annotated with a dot at its approximate center. In addition, three object instances are selected



(a) Image categories and number of images for each category in our dataset.

(b) Number of images in several ranges of object count.

			Annotation type	
Dataset	Images	Categories	Dot	Bounding Box
UCF CC 50 [15]	50	1	✓	✗
Shanghaitech [55]	1198	1	✓	✗
UCF QNRF [16]	1535	1	✓	✗
NWPU [49]	5109	1	✓	✗
JHU Crowd [43]	4372	1	✓	✓
CARPK [14]	1448	1	✓	✓
Proposed	6135	147	✓	✓

(c) Comparison with popular counting datasets.

Figure 3: **Categories & no. of images per category, object counts, and comparison with other counting datasets**

randomly as exemplar instances; these exemplars are also annotated with axis-aligned bounding boxes. In the following subsections, we will describe how the data was collected and annotated. We will also report the detailed statistics and how the data was split into disjoint training, validation, and testing sets.

4.1. Image Collection

To obtain the set of 6135 images for our dataset, we started with a set of candidate images obtained by keyword searches. Subsequently, we performed manual inspection to filter out images that do not satisfy our predefined conditions as described below.

Image retrieval. We started with a list of object categories, and collected 300–3000 candidate images for each category by scraping the web. We used Flickr, Google, and Bing search engines with the open source image scrappers [7, 45]. We added adjectives such as *many, multiple, lots of*, and *stack of* in front of the category names to create the search query keywords.

Manual verification and filtering. We manually inspected the candidate images and only kept the suitable ones satisfying the following criteria:

1. *High image quality*: The resolution should be high enough to easily differentiate between objects.
2. *Large enough object count*: The number of objects of interest should be at least 7. We are more interested in counting a large number of objects, since humans do not need help counting a small number of objects.
3. *Appearance similarity*: we selected images where object instances have somewhat similar poses, texture, and appearance.
4. *No severe occlusion*: in most cases, we removed candidate images where severe occlusion prevents humans from accurately counting the objects.

4.2. Image Annotation

Images in the dataset were annotated by a group of annotators using the OpenCV Image and Video Annotation Tool [1]. Two types of annotation were collected for each image, dots and bounding boxes, as illustrated in Fig. 4. For images containing multiple categories, we picked only one of the categories. Each object instance in an image was marked with a dot at its approximate center. In case of occlusion, the occluded instance was only counted and annotated if the amount of occlusion was less than 90%. For each image, we arbitrarily chose three objects as exemplar instances and we drew axis-aligned bounding boxes for those instances.

4.3. Dataset split

We divided the dataset into train, validation, and test sets such that they do not share any object category. We randomly selected 89 object categories for the train set, and 29 categories each for the validation and test sets. The train, validation, and test sets consist of 3659, 1286 and 1190 images respectively.

4.4. Data Statistics

The dataset contains a total of 6135 images. The average height and width of the images are 774 and 938 pixels, respectively. The average number of objects per image is 56, and the total number of objects is 343,818. The minimum and maximum number of objects for one image are 7 and 3701, respectively. The three categories with the highest number of objects per image are: Lego (303 ob-

Method	Val Set		Test Set	
	MAE	RMSE	MAE	RMSE
Mean	53.38	124.53	47.55	147.67
Median	48.68	129.70	47.73	152.46
FR few-shot detector [17]	45.45	112.53	41.64	141.04
FSOD few-shot detector [8]	36.36	115.00	32.53	140.65
Pre-trained GMN [28]	60.56	137.78	62.69	159.67
GMN [28]	29.66	89.81	26.52	124.57
MAML [9]	25.54	79.44	24.90	112.68
FamNet (Proposed)	**23.75**	**69.07**	**22.08**	**99.54**

Table 1: Comparing FamNet to two simple baselines (Mean, Median) and four stronger baseline (Feature Reweighting (FR) few-shot detector, FSOD few-shot detector, GMN and MAML), these are few-shot methods that have been adapted and trained for counting. FamNet has the lowest MAE and RMSE on both val and test sets.

jects/image), Brick (271), and Marker (247). The three categories with lowest number of objects per image are: Supermarket shelf (8 objects/image), Meat Skewer (8), and Oyster (11). Fig. 3b is a histogram plot for the number of images in several ranges of object count.

5. Experiments

5.1. Performance Evaluation Metrics

We use Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) to measure the accuracy of a counting method. MAE and RMSE are commonly used metrics for counting task [29, 32, 55], and they are defined as follows. $MAE = \frac{1}{n}\sum_{i=1}^{n}|c_i - \hat{c}_i|; RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(c_i - \hat{c}_i)^2}$, where n is the number of test images, and c_i and $\hat{c}_i$ are the ground truth and predicted counts.

5.2. Comparison with Few-Shot Approaches

We compare the performance of FamNet with two trivial baselines and four competing few-shot methods. The two trivial baseline methods are: (1) always output the average object count for training images; (2) always output the median count for the training images. We also implement stronger methods for comparison, by adapting several few-shot methods for the counting task and training them on our training data. Specifically, we adapt the following approaches for counting: the state-of-the-art few-shot detectors [8, 17], the Generic Matching Network (GMN) [28], and Model Agnostic Meta Learning (MAML) [9]. We implement MAML using the higher library [10], which is a meta learning library supporting higher order optimization. The training procedure of MAML involves an *inner optimization loop*, which adapts the network to the specific test classes, and an *outer optimization loop* which learns meta parameters that facilitate faster generalization to novel tasks. At test time, only the inner optimization is performed. We use the $\mathcal{L}_{Adapt}$ loss defined in Eq. (3) for the inner optimization loop, and the MSE loss over the entire dot annotation map for the outer optimization loop.

Method	Val-COCO Set		Test-COCO Set	
	MAE	RMSE	MAE	RMSE
Faster R-CNN	52.79	172.46	36.20	79.59
RetinaNet	63.57	174.36	52.67	85.86
Mask R-CNN	52.51	172.21	35.56	80.00
FamNet (Proposed)	**39.82**	**108.13**	**22.76**	**45.92**

Table 2: **Comparing FamNet with pre-trained object detectors**, on counting objects from categories where there are pre-trained object detectors.

As can be seen in Table 1, FamNet outperforms all the other methods. Surprisingly, the pre-trained GMN does not work very well, even though it is a class agnostic counting method. The GMN model trained on our training data performs better than its pre-trained version; and this demonstrates the benefits of our dataset. The state-of-the-art few-shot detectors [8, 17] perform relatively poor, even when they are trained on our dataset. With these results, we are the first to show the empirical evidence for the inferiority of the detection-then-counting approach compared to the density estimation approach (GMN, MAML, FamNet) for generic object counting. However, this is not new for the crowd counting research community, where the density estimation approach dominates the recent literature [55], thanks to its robustness to occlusion and the freedom of not having to commit to binarized decisions at an early stage. Among the competing approaches, MAML is the best method of all. This is perhaps because MAML is a meta learning method that leverages the advantages of having the FamNet architecture as its core component. The MAML way of training this network leads to a better model than GMN, but it is still inferior to the proposed FamNet together with the proposed training and adaptation algorithms. In terms of training time per epoch, FamNet is around three times faster than MAML, because it does not require any higher order gradient computation like MAML.

5.3. Comparison with Object Detectors

One approach for counting is to use a detector to detect objects and then count. This approach only works for certain categories of objects, where there are detectors for those categories. In general, it requires thousands of examples to train an object detector, so this is not a practical method for general visual counting. Nevertheless, we evaluate the performance of FamNet on a subset of categories from the validation and test sets that have pre-trained object detectors on the COCO dataset. We refer to these

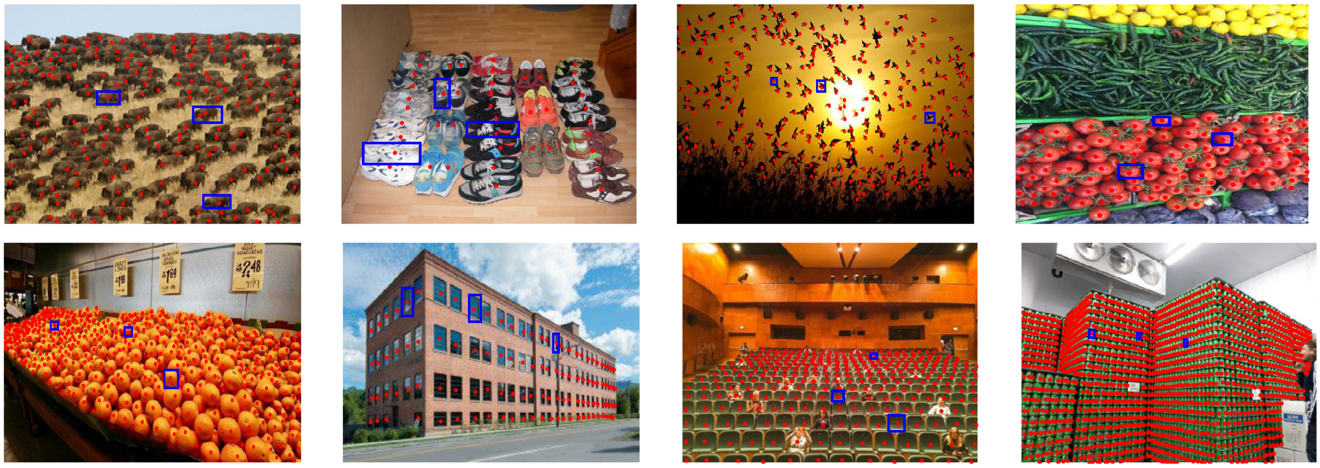


Figure 4: **Few annotated images from the dataset**. Dot and box annotations are shown in red and blue respectively. The number of objects in each image varies widely, some images contain a dozen of objects while some contains thousands.

Number of Exemplars	MAE	RMSE
1	26.55	77.01
2	24.09	72.37
3	23.75	69.07

Table 3: Performance of FamNet on the validation data as the number of exemplars increases. FamNet can provide a reasonable count estimate even with a single exemplar, and the estimate becomes more accurate with more exemplars.

subsets as Val-COCO and Test-COCO, which comprise of 277 and 282 images respectively. Specifically, we compare FamNet with FasterRCNN [37], MaskRCNN [11], and RetinaNet [24]. All of these pretrained detectors are available in the Detectron2 library [52]. Table 2 shows the comparison results. As can be seen, FamNet outperforms the pre-trained detectors, even on object categories where the detectors have been trained with thousands of annotated examples from the COCO dataset.

Components	Combinations			
Multi-scale image feature	✗	✓	✓	✓
Multi-scale exemplar feature	✗	✗	✓	✓
Test time adaptation	✗	✗	✗	✓
MAE	32.70	27.80	24.32	23.75
RMSE	104.31	93.53	70.94	69.07

Table 4: **Analyzing the components of FamNet**. Each of the components of FamNet adds to the performance.

5.4. Ablation Studies

We perform ablation studies on the validation set of FSC-147 to analyze: (1) how the counting performance changes as the number of exemplars increases, and (2) the benefits of different components of FamNet.

In Table 3, we analyze the performance of FamNet as the number of exemplars is varied between one to three during the testing of FamNet. We see that FamNet can work even with one exemplar, and it outperforms all the competing methods presented in Table 1 with just 2 exemplars. Not surprisingly, the performance of FamNet improves as the number of exemplars is increased. This suggests that an user of our system can obtain a reasonable count even with a single exemplar, and they can obtain a more accurate count by providing more exemplars.

In Table 4, we analyze the importance of the key components of FamNet: multi-scale image feature map, the multi-scale exemplar features, and test time adaptation. We train models without few/all of these components on the training set of FSC-147, and report the validation performance. We notice that all of the components of FamNet are important, and adding each of the component leads to improved results.

5.5. Counting category-specific objects

FamNet is specifically designed to be general, being able to count generic objects with only a few exemplars. As such, it might not be fair to demand it to work extremely well for a specific category, such as counting cars. Cars are popular objects that appear in many datasets and this category is the explicit or implicit target for tuning for many networks, so it would not be surprising if our method does not perform as well as other customized solutions. Having said that, we still investigate the suitability of using FamNet to count cars from the CARPK dataset [14], which consists of overhead images of parking lots taken by downward facing drone cameras. The training and test set consists of 989

Method	MAE	RMSE
YOLO [14, 36]	48.89	57.55
Faster RCNN [14, 38]	47.45	57.39
One-look Regression [14, 30]	59.46	66.84
Faster RCNN [14, 38](RPN-small)	24.32	37.62
Spatially Regularized RPN [14]	23.80	36.79
GMN [28]	7.48	9.90
FamNet– (pre-trained)	28.84	44.47
FamNet+ (trained with CARPK data)	18.19	33.66

Table 5: **Counting car performance on the CARPK dataset**. FamNet– is a FamNet model, that is trained without any CARPK images nor images from the car category of FSC-147. Other methods use the entire CARPK train set. Pre-trained FamNet– outperforms three of of the previous approaches. FamNet+, yields even better performance.

and 459 images respectively. There are around 90,000 instances of cars in the dataset.

We experiment with two variants of FamNet: a pre-trained model and a model trained on CARPK dataset. The pre-trained FamNet model is called FamNet–, which is trained on FSC-147, without using the data from CARPK or the car category from FSC-147. The FamNet model trained with training data from CARPK is called FamNet+, and it is trained as follows. We randomly sample a set of 12 exemplars from the training set, and use these as the exemplars for all of the training and test images. We train FamNet+ on the CARPK training set. Table 5 displays the results of several methods on this CARPK dataset. FamNet+ outperforms all methods except GMN [28]. GMN, unlike all the other approaches, uses extra training data from the ILSVRC video dataset which consists of video sequences of cars. Perhaps this may be why GMN works particularly well on CARPK.

5.6. Qualitative Results

Fig. 5 shows few images and FamNet predictions. The first three are success cases,and the last is a failure case. For the fourth image, FamNet confuses portions of the background as being the foreground, because of similarity in appearance between the background and the object of interest. Fig. 6 shows a test case where test time adaptation improves on the initial count by decreasing the density values in the dense regions.

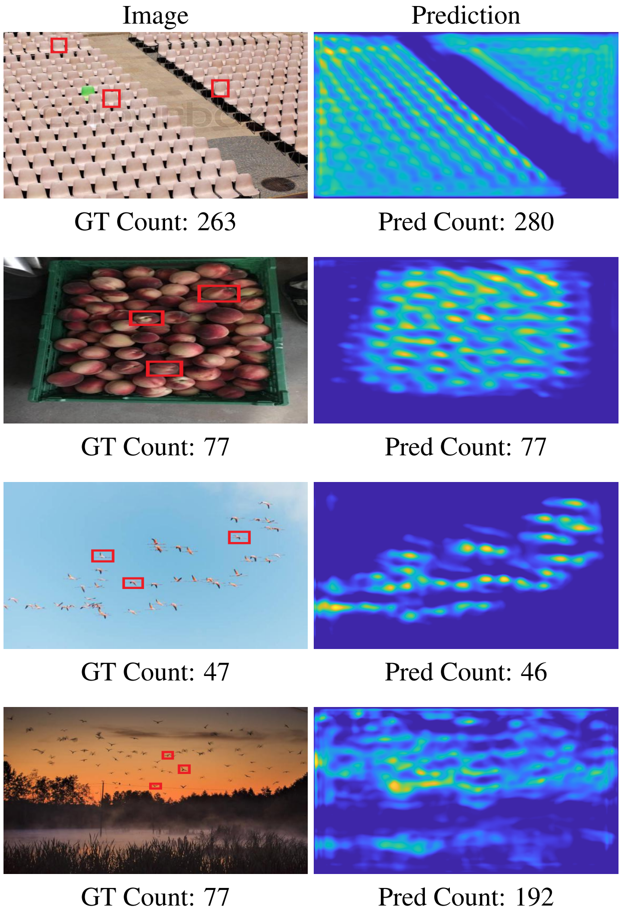


Figure 5: **Predicted density maps and counts of FamNet**.

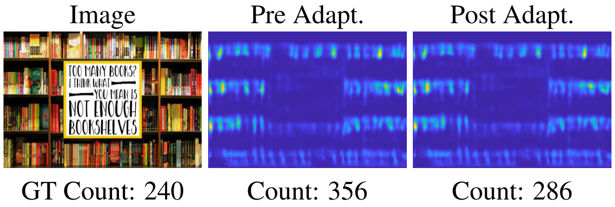


Image
Pre Adapt.
Post Adapt.
TOO MANY BOOKS? I THINK WHAT YOU MEAN IS NOT ENOUGH BOOKSHELVES
GT Count: 240
Count: 356
Count: 286

Figure 6: **Test time adaptation**. Shown are the initial density map (Pre Adapt) and final density map after adaptation (Post Adapt). In case of over counting, adaptation decreases the density values at dense locations.

6. Conclusions

In this paper, we posed counting as a few-shot regression task. Given the non-existence of a suitable dataset for the few-shot counting task, we collected a visual counting dataset with relatively large number of object categories and instances. We also presented a novel approach for density prediction suitable for the few-shot visual counting task. We compared our approach with several state-of-art detectors and few shot counting approaches, and showed that our approach outperforms all of these approaches.

Acknowledgements: This project is partially supported by MedPod, the SUNY2020 Infrastructure Transportation Security Center, and the NSF I/UCRC Center for Visual and Decision Informatics at Stony Brook.

References

[1] Computer vision annotation tool.

[2] Shahira Abousamra, Minh Hoai, Dimitris Samaras, and Chao Chen. Localization in the crowd with topological constraints. In *Proceedings of AAAI Conference on Artificial Intelligence*, 2021.

[3] Carlos Arteta, Victor Lempitsky, J Alison Noble, and Andrew Zisserman. Detecting overlapping instances in microscopy images using extremal region trees. *Medical image analysis*, 27:3–16, 2016.

[4] Carlos Arteta, Victor Lempitsky, and Andrew Zisserman. Counting in the wild. In *Proceedings of the European Conference on Computer Vision*, 2016.

[5] Deepak Babu Sam, Neeraj N Sajjan, R Venkatesh Babu, and Mukundhan Srinivasan. Divide and grow: Capturing huge diversity in crowd images with incrementally growing cnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[6] 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*, 2018.

[7] Del Riccardo Chiaro. python-flickr-image-downloader.

[8] Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai. Few-shot object detection with attention-rpn and multi-relation detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the International Conference on Machine Learning*, 2017.

[10] Edward Grefenstette, Brandon Amos, Denis Yarats, Phu Mon Htut, Artem Molchanov, Franziska Meier, Douwe Kiela, Kyunghyun Cho, and Soumith Chintala. Generalized inner loop meta-learning. *arXiv preprint arXiv:1910.01727*, 2019.

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the International Conference on Computer Vision*, 2017.

[12] 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*, 2016.

[13] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596, 2015.

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

[15] Haroon Idrees, Imran Saleemi, Cody Seibert, and Mubarak Shah. Multi-source multi-scale counting in extremely dense crowd images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[16] Haroon Idrees, Muhmmad Tayyab, Kishan Athrey, Dong Zhang, Somaya Al-Maadeed, Nasir Rajpoot, and Mubarak Shah. Composition loss for counting, density map estimation and localization in dense crowds. In *Proceedings of the European Conference on Computer Vision*, 2018.

[17] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *Proceedings of the International Conference on Computer Vision*, 2019.

[18] Aisha Khan, Stephen Gould, and Mathieu Salzmann. Deep convolutional neural networks for human embryonic cell counting. In *Proceedings of the European Conference on Computer Vision*. Springer, 2016.

[19] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, 2015.

[20] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, December 2015.

[21] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[22] Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. In *Advances in Neural Information Processing Systems*, 2010.

[23] Yuhong Li, Xiaofan Zhang, and Deming Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[24] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the International Conference on Computer Vision*, 2017.

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, 2014.

[26] Weizhe Liu, Mathieu Salzmann, and Pascal Fua. Context-aware crowd counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[27] Xialei Liu, Joost Van De Weijer, and Andrew D Bagdanov. Leveraging unlabeled data for crowd counting by learning to rank. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[28] Erika Lu, Weidi Xie, and Andrew Zisserman. Class-agnostic counting. In *Proceedings of the Asian Conference on Computer Vision*, 2018.

[29] Zhiheng Ma, Xing Wei, Xiaopeng Hong, and Yihong Gong. Bayesian loss for crowd count estimation with point supervision. In *Proceedings of the International Conference on Computer Vision*, 2019.

[30] T Nathan Mundhenk, Goran Konjevod, Wesam A Sakla, and Kofi Boakye. A large contextual dataset for classification, detection and counting of cars with deep learning. In *Proceedings of the European Conference on Computer Vision*, 2016.

[31] Maryam Rahnemoonfar and Clay Sheppard. Deep count: fruit counting based on deep simulated learning. *Sensors*, 17(4):905, 2017.

[32] Viresh Ranjan, Hieu Le, and Minh Hoai. Iterative crowd

counting. In *Proceedings of the European Conference on Computer Vision*, 2018.

[33] Viresh Ranjan, Mubarak Shah, and Minh Hoai Nguyen. Crowd transformer network. *arXiv preprint arXiv:1904.02774*, 2019.

[34] Viresh Ranjan, Boyu Wang, Mubarak Shah, and Minh Hoai. Uncertainty estimation and sample selection for crowd counting. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[35] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

[36] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[37] 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*. 2015.

[38] 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*, 2015.

[39] Deepak Babu Sam, Shiv Surya, and R Venkatesh Babu. Switching convolutional neural network for crowd counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[40] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. One-shot learning with memory-augmented neural networks. 2016.

[41] Eli Shechtman and Michal Irani. Matching local self-similarities across images and videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[42] 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*, 2019.

[43] Vishwanath A Sindagi, Rajeev Yasarla, and Vishal M Patel. Jhu-crowd++: Large-scale crowd counting dataset and a benchmark method. *arXiv preprint arXiv:2004.03597*, 2020.

[44] Jack Valmadre, Luca Bertinetto, Joao Henriques, Andrea Vedaldi, and Philip HS Torr. End-to-end representation learning for correlation filter based tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[45] Hardik Vasa. Google images download.

[46] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, 2016.

[47] Jia Wan and Antoni Chan. Adaptive density map generation for crowd counting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1130–1139, 2019.

[48] Boyu Wang, Huidong Liu, Dimitris Samaras, and Minh Hoai. Distribution matching for crowd counting. In *Advances in Neural Information Processing Systems*. 2020.

[49] Qi Wang, Junyu Gao, Wei Lin, and Xuelong Li. Nwpu-crowd: A large-scale benchmark for crowd counting. *arXiv preprint arXiv:2001.03360*, 2020.

[50] Qi Wang, Junyu Gao, Wei Lin, and Yuan Yuan. Learning from synthetic data for crowd counting in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[51] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[52] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2, 2019.

[53] Weidi Xie, J Alison Noble, and Andrew Zisserman. Microscopy cell counting and detection with fully convolutional regression networks. *Computer methods in biomechanics and biomedical engineering: Imaging & Visualization*, 6(3):283–292, 2018.

[54] Anran Zhang, Lei Yue, Jiayi Shen, Fan Zhu, Xiantong Zhen, Xianbin Cao, and Ling Shao. Attentional neural fields for crowd counting. In *Proceedings of the International Conference on Computer Vision*, 2019.

[55] 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*, 2016.