Zero-Shot Object Counting

Jingyi Xu[1], Hieu Le[2], Vu Nguyen[1], Viresh Ranjan[*3], and Dimitris Samaras[1]

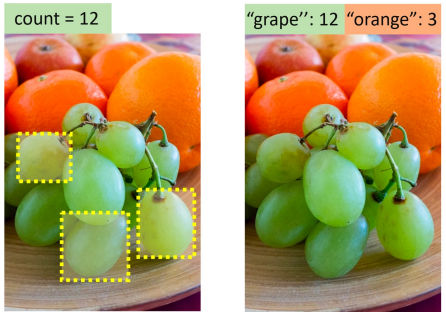
[1]Stony Brook University [2]EPFL [3]Amazon

Abstract

Class-agnostic object counting aims to count object instances of an arbitrary class at test time. Current methods for this challenging problem require human-annotated exemplars as inputs, which are often unavailable for novel categories, especially for autonomous systems. Thus, we propose zero-shot object counting (ZSC), a new setting where only the class name is available during test time. Such a counting system does not require human annotators in the loop and can operate automatically. Starting from a class name, we propose a method that can accurately identify the optimal patches which can then be used as counting exemplars. Specifically, we first construct a class prototype to select the patches that are likely to contain the objects of interest, namely class-relevant patches. Furthermore, we introduce a model that can quantitatively measure how suitable an arbitrary patch is as a counting exemplar. By applying this model to all the candidate patches, we can select the most suitable patches as exemplars for counting. Experimental results on a recent class-agnostic counting dataset, FSC-147, validate the effectiveness of our method. Code is available at https://github.com/cvlab-stonybrook/zero-shot-counting.

1. Introduction

Object counting aims to infer the number of objects in an image. Most of the existing methods focus on counting objects from specialized categories such as human crowds [37], cars [29], animals [4], and cells [46]. These methods count only a single category at a time. Recently, class-agnostic counting [28, 34, 38] has been proposed to count objects of arbitrary categories. Several human-annotated bounding boxes of objects are required to specify the objects of interest (see Figure 1a). However, having humans in the loop is not practical for many real-world applications, such as fully automated wildlife monitoring systems or visual anomaly detection systems.



(a) Few-shot Counting (b) Zero-Shot Counting

Figure 1. Our proposed task of zero-shot object counting (ZSC). Traditional few-shot counting methods require a few exemplars of the object category (a). We propose zero-shot counting where the counter only needs the class name to count the number of object instances. (b). Few-shot counting methods require human annotators at test time while zero-shot counters can be fully automatic.

A more practical setting, exemplar-free class-agnostic counting, has been proposed recently by Ranjan *et al.* [33]. They introduce RepRPN, which first identifies the objects that occur most frequently in the image, and then uses them as exemplars for object counting. Even though RepRPN does not require any annotated boxes at test time, the method simply counts objects from the class with the highest number of instances. Thus, it can not be used for counting a specific class of interest. The method is only suitable for counting images with a single dominant object class, which limits the potential applicability.

Our goal is to build an exemplar-free object counter where we can specify what to count. To this end, we introduce a new counting task in which the user only needs to provide the name of the class for counting rather than the exemplars (see Figure 1b). In this way, the counting model can not only operate in an automatic manner but also allow the user to define what to count by simply providing the class name. Note that the class to count during test time can be arbitrary. For cases where the test class is completely unseen to the trained model, the counter needs to adapt to the unseen class without any annotated data. Hence, we

[*]Work done prior to joining Amazon

name this setting zero-shot object counting (ZSC), inspired by previous zero-shot learning approaches [6, 57].

To count without any annotated exemplars, our idea is to identify a few patches in the input image containing the target object that can be used as counting exemplars. Here the challenges are twofold: 1) how to localize patches that contain the object of interest based on the provided class name, and 2) how to select *good* exemplars for counting. Ideally, good object exemplars are visually representative for most instances in the image, which can benefit the object counter. In addition, we want to avoid selecting patches that contain irrelevant objects or backgrounds, which likely lead to incorrect object counts.

To this end, we propose a two-step method that first localizes the class-relevant patches which contain the objects of interest based on the given class name, and then selects among these patches the optimal exemplars for counting. We use these selected exemplars, together with a pre-trained exemplar-based counting model, to achieve exemplar-free object counting.

In particular, to localize the patches containing the objects of interest, we first construct a class prototype in a pre-trained embedding space based on the given class name. To construct the class prototype, we train a conditional variational autoencoder (VAE) to generate features for an arbitrary class conditioned on its semantic embedding. The class prototype is computed by taking the average of the generated features. We then select the patches whose embeddings are the k-nearest neighbors of the class prototype as the class-relevant patches.

After obtaining the class-relevant patches, we further select among them the optimal patches to be used as counting exemplars. Here we observe that the feature maps obtained using *good* exemplars and *bad* exemplars often exhibit distinguishable differences. An example of the feature maps obtained with different exemplars is shown in Figure 2. The feature map from a *good* exemplar typically exhibits some repetitive patterns (e.g., the dots on the feature map) that center around the object areas while the patterns from a *bad* exemplar are more irregular and occur randomly across the image. Based on this observation, we train a model to measure the goodness of an input patch based on its corresponding feature maps. Specifically, given an arbitrary patch and a pre-trained exemplar-based object counter, we train this model to predict the counting error of the counter when using the patch as the exemplar. Here the counting error can indicate the goodness of the exemplar. After this error predictor is trained, we use it to select those patches with the smallest predicted errors as the final exemplars for counting.

Experiments on the FSC-147 dataset show that our method outperforms the previous exemplar-free counting method [33] by a large margin. We also provide analyses to show that patches selected by our method can be used in other exemplar-based counting methods to achieve exemplar-free counting. In short, our main contributions can be summarized as follows:

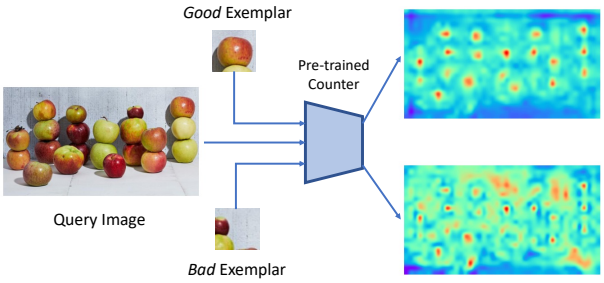


Figure 2. Feature maps obtained using different exemplars given a pre-trained exemplar-based counting model. The feature maps obtained using good exemplars typically exhibit some repetitive patterns while the patterns from bad exemplars are more irregular.

- We introduce the task of zero-shot object counting that counts the number of instances of a specific class in the input image, given only the class name and without relying on any human-annotated exemplars.
- We propose a simple yet effective patch selection method that can accurately localize the optimal patches across the query image as exemplars for zero-shot object counting.
- We verify the effectiveness of our method on the FSC-147 dataset, through extensive ablation studies and visualization results.

2. Related Work

2.1. Class-specific Object Counting

Class-specific object counting focuses on counting predefined categories, such as humans [1, 15, 24, 26, 37, 39, 40, 42, 47, 52, 53, 55, 56], animals [4], cells [46], or cars [14, 29]. Generally, existing methods can be categorized into two groups: detection-based methods [8, 14, 18] and regression-based methods [7, 10, 11, 27, 41, 53, 56]. Detection-based methods apply an object detector on the image and count the number of objects based on the detected boxes. Regression-based methods predict a density map for each input image, and the final result is obtained by summing up the pixel values. Both types of methods require abundant training data to learn a good model. Class-specific counters can perform well on trained categories. However, they can not be used to count objects of arbitrary categories at test time.

2.2. Class-agnostic Object Counting

Class-agnostic object counting aims to count arbitrary categories given only a few exemplars [3, 13, 25, 28, 31, 34, 38, 50, 51]. GMN [28] uses a shared embedding module to

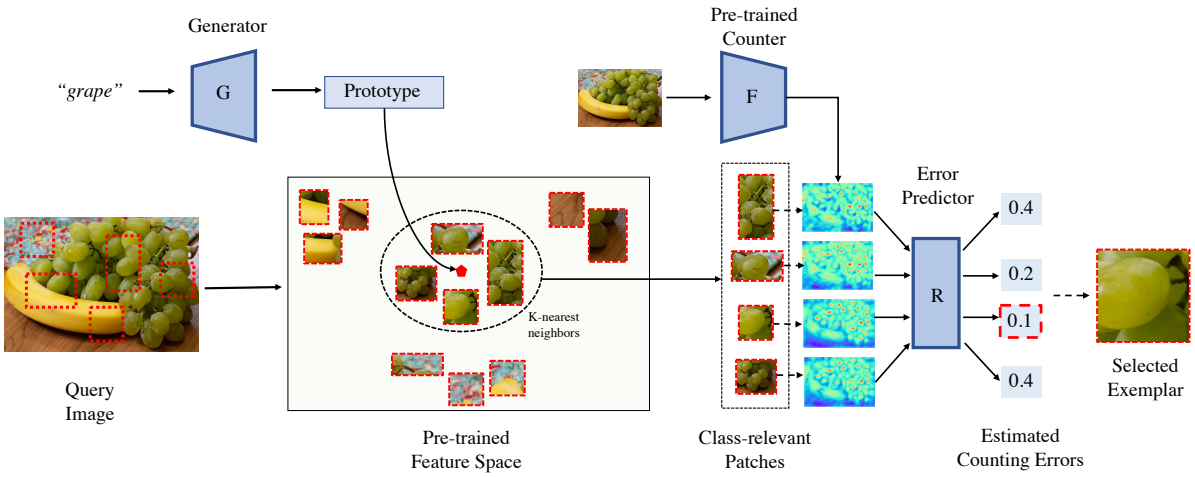


Figure 3. Overview of the proposed method. We first use a generative model to obtain a class prototype for the given class (e.g. grape) in a pre-trained feature space. Then given an input query image, we randomly sample a number of patches of various sizes and extract the corresponding feature embedding for each patch. We select the patches whose embeddings are the nearest neighbors of the class prototype as class-relevant patches. Then for each of the selected class-relevant patches, we use a pre-trained exemplar-based counting model to obtain the intermediate feature maps. Our proposed error predictor then takes the feature maps as input and predicts the counting error (here we use normalized counting errors). We select the patches with the smallest predicted errors as the final exemplar patches and use them for counting.

extract feature maps for both query images and exemplars, which are then concatenated and fed into a matching module to regress the object count. FamNet [34] adopts a similar way to do correlation matching and further applies test-time adaptation. These methods require human-annotated exemplars as inputs. Recently, Ranjan *et al.* have proposed RepRPN [33], which achieves exemplar-free counting by identifying exemplars from the most frequent objects via a Region Proposal Network (RPN)-based model. However, the class of interest can not be explicitly specified for the RepRPN. In comparison, our proposed method can count instances of a specific class given only the class name.

2.3. Zero-shot Image Classification

Zero-shot classification aims to classify unseen categories for which data is not available during training [5, 9, 12, 16, 19, 21, 23, 35, 36]. Semantic descriptors are mostly leveraged as a bridge to enable the knowledge transfer between seen and unseen classes. Earlier zero-shot learning (ZSL) works relate the semantic descriptors with visual features in an embedding space and recognize unseen samples by searching their nearest class-level semantic descriptor in this embedding space [17, 36, 43, 54]. Recently, generative models [20, 22, 48, 49] have been widely employed to synthesize unseen class data to facilitate ZSL [30, 44, 45]. Xian *et al.* [44] use a conditional Wasserstein Generative Adversarial Network (GAN) [2] to generate unseen features which can then be used to train a discriminative classifier for ZSL. In our method, we also train a generative model conditioned on class-specific semantic embedding. Instead of using this generative model to hallucinate data, we use it to compute a prototype for each class. This class prototype is then used to select patches that contain objects of interest.

3. Method

Figure 3 summarizes our proposed method. Given an input query image and a class label, we first use a generative model to construct a class prototype for the given class in a pre-trained feature space. We then randomly sample a number of patches of various sizes and extract the feature embedding for each patch. The class-relevant patches are those patches whose embeddings are the nearest neighbors of the class prototype in the embedding space. We further use an error predictor to select the patches with the smallest predicted errors as the final exemplars for counting. We use the selected exemplars in an exemplar-based object counter to infer the object counts. For the rest of the paper, we denote this exemplar-based counter as the "base counting model". We will first describe how we train this base counting model and then present the details of our patch selection method.

3.1. Training Base Counting Model

We train our base counting model using abundant training images with annotations. Similar to previous works [34, 38], the base counting model uses the input image and the exemplars to obtain a density map for object counting. The model consists of a feature extractor F and a counter C. Given a query image I and an exemplar B of an arbitrary class c, we input I and B to the feature extractor to obtain the corresponding output, denoted as $F(I)$ and $F(B)$ re-

spectively. $F(I)$ is a feature map of size $d * h_I * w_I$ and $F(B)$ is a feature map of size $d * h_B * w_B$. We further perform global average pooling on $F(B)$ to form a feature vector b of d dimensions.

After feature extraction, we obtain the similarity map S by correlating the exemplar feature vector b with the image feature map $F(I)$. Specifically, if $w_{ij} = F_{ij}(I)$ is the channel feature at spatial position (i, j), S can be computed by:

$$S_{ij}(I, B) = w_{ij}^T b. \tag{1}$$

In the case where n exemplars are given, we use Eq. 1 to calculate n similarity maps, and the final similarity map is the average of these n similarity maps.

We then concatenate the image feature map $F(I)$ with the similarity map S, and input them into the counter C to predict a density map D. The final predicted count N is obtained by summing over the predicted density map D:

$$N = \sum_{i,j} D_{(i,j)}, \tag{2}$$

where $D_{(i,j)}$ denotes the density value for pixel (i, j). The supervision signal for training the counting model is the L_2 loss between the predicted density map and the ground truth density map:

$$L_{\text{count}} = \|D(I, B) - D^*(I)\|_2^2, \tag{3}$$

where D^* denotes the ground truth density map.

3.2. Zero-shot Object Counting

In this section, we describe how we count objects of any unseen category given only the class name without access to any exemplar. Our strategy is to select a few patches in the image that can be used as exemplars for the base counting model. These patches are selected such that: 1) they contain the objects that we are counting and 2) they benefit the counting model, i.e., lead to small counting errors.

3.2.1 Selecting Class-relevant Patches

To select patches that contain the objects of interest, we first generate a class prototype based on the given class name using a conditional VAE model. Then we randomly sample a number of patches across the query image and select the class-relevant patches based on the generated prototype.

Class prototype generation. Inspired by previous zero-shot learning approaches [44, 45], we train a conditional VAE model to generate features for an arbitrary class based on the semantic embedding of the class. The semantic embedding is obtained from a pre-trained text-vision model [32] given the corresponding class name. Specifically, we train the VAE model to reconstruct features in a pre-trained ImageNet feature space. The VAE is composed of an Encoder E, which maps a visual feature x to a latent code z, and a decoder G which reconstructs x from z. Both E and G are conditioned on the semantic embedding a .The loss function for training this VAE for an input feature x can be defined as:

$$\begin{aligned} L_V(x) = &\text{KL}\left(q(z|x, a)||p(z|a)\right) \\ &-\text{E}_{q(z|x,a)}[\log p(x|z, a)]. \end{aligned} \tag{4}$$

The first term is the Kullback-Leibler divergence between the VAE posterior $q(z|x, a)$ and a prior distribution $p(z|a)$. The second term is the decoder's reconstruction error. $q(z|x, a)$ is modeled as $E(x, a)$ and $p(x|z, a)$ is equal to $G(z, a)$. The prior distribution is assumed to be $\mathcal{N}(0, I)$ for all classes.

We can use the trained VAE to generate the class prototype for an arbitrary target class for counting. Specifically, given the target class name y, we first generate a set of features by inputting the respective semantic vector a^y and a noise vector z to the decoder G:

$$\mathbb{G}^y = \{\hat{x}|\hat{x} = G(z, y), z \sim \mathcal{N}(0, I)\}. \tag{5}$$

The class prototype p^y is computed by taking the mean of all the features generated by VAE:

$$\text{p}^y = \frac{1}{|\mathbb{G}^y|} \sum\nolimits_{\hat{x} \in \mathbb{G}^y} \hat{x} \tag{6}$$

Class-relevant patch selection. The generated class prototype can be considered as a class center representing the distribution of features of the corresponding class in the embedding space. Using the class prototype, we can select the class-relevant patches across the query image. Specifically, we first randomly sample M patches of various sizes $\{b_1, b_2, ..., b_m\}$ across the query image and extract their corresponding ImageNet features $\{f_1, f_2, ..., f_m\}$. To select the class-relevant patches, we calculate the L_2 distance between the class prototype and the patch embedding, namely $d_i = \|f_i - \text{p}^y\|_2$. Then we select the patches whose embeddings are the k-nearest neighbors of the class prototype as the class-relevant patches. Since the ImageNet feature space is highly discriminative, i.e., features close to each other typically belong to the same class, the selected patches are likely to contain the objects of the target class.

3.2.2 Selecting Exemplars for Counting

Given a set of class-relevant patches and a pre-trained exemplar-based object counter, we aim to select a few exemplars from these patches that are optimal for counting. To do so, we introduce an error prediction network that predicts the counting error of an arbitrary patch when the patch is used as the exemplar. The counting error is calculated from the pre-trained counting model. Specifically, to train this error predictor, given a query image $\bar{I}$ and an arbitrary patch

$\bar{B}$ cropped from $\bar{I}$, we first use the base counting model to get the image feature map $F(\bar{I})$, similarity map $\bar{S}$, and the final predicted density map $\bar{D}$. The counting error of the base counting model can be written as:

$$\epsilon = |\sum_{i,j} \bar{D}_{(i,j)} - \bar{N}^*|, \tag{7}$$

where $\bar{N}^*$ denotes the ground truth object count in image $\bar{I}$. ϵ can be used to measure the goodness of $\bar{B}$ as an exemplar for $\bar{I}$, i.e., a small ϵ indicates that $\bar{B}$ is a suitable exemplar for counting and vice versa.

The error predictor R is trained to regress the counting error produced by the base counting model. The input of R is the channel-wise concatenation of the image feature map $F(\bar{I})$ and the similarity map $\bar{S}$. The training objective is the minimization of the mean squared error between the output of the predictor $R(F(\bar{I}), \bar{S})$ and the actual counting error produced by the base counting model ϵ.

After the error predictor is trained, we can use it to select the optimal patches for counting. The candidates for selection here are the class-relevant patches selected by the class prototype in the previous step. For each candidate patch, we use the trained error predictor to infer the counting error when it is being used as the exemplar. The final selected patches for counting are the patches that yield the top-s smallest counting errors.

3.2.3 Using the Selected Patches as Exemplars

Using the error predictor, we predict the error for each candidate patch and select the patches that lead to the smallest counting errors. The selected patches can then be used as exemplars for the base counting model to get the density map and the final count. We also conduct experiments to show that these selected patches can serve as exemplars for other exemplar-based counting models to achieve exemplar-free class-agnostic counting.

4. Experiments

4.1. Implementation Details

Network architecture For the *base counting model*, we use ResNet-50 as the backbone of the feature extractor, initialized with the weights of a pre-trained ImageNet model. The backbone outputs feature maps of 1024 channels. For each query image, the number of channels is reduced to 256 using an 1×1 convolution. For each exemplar, the feature maps are first processed with global average pooling and then linearly mapped to obtain a 256-d feature vector. The counter consists of 5 convolutional and bilinear upsampling layers to regress a density map of the same size as the query image. For the *feature generation model*, both the encoder and the decoder are two-layer fully-connected (FC) networks with 4096 hidden units. LeakyReLU and ReLU are the non-linear activation functions in the hidden and output layers, respectively. The dimensions of the latent space and the semantic embeddings are both set to be 512. For the *error predictor*, 5 convolutional and bilinear upsampling layers are followed by a linear layer to output the counting error.

Dataset We use the FSC-147 dataset [34] to train the base counting model and the error predictor. FSC-147 is the first large-scale dataset for class-agnostic counting. It includes 6135 images from 147 categories varying from animals, kitchen utensils, to vehicles. The categories in the training, validation, and test sets do not overlap. The feature generator is trained on the MS-COCO detection dataset. Note that the previous exemplar-free method [33] also uses MS-COCO to pre-train their counter.

Training details Both the base counting model and the error predictor are trained using the AdamW optimizer with a fixed learning rate of 10^{-5}. The base counting model is trained for 300 epochs with a batch size of 8. We resize the input query image to a fixed height of 384, and the width is adjusted accordingly to preserve the aspect ratio of the original image. Exemplars are resized to 128×128 before being input into the feature extractor. The feature generation model is trained using the Adam optimizer and the learning rate is set to be 10^{-4}. The semantic embeddings are extracted from CLIP [32]. To select the class-relevant patches, we randomly sample 450 boxes of various sizes across the input query image and select 10 patches whose embeddings are the 10-nearest neighbors of the class prototype. The final selected patches are those that yield the top-3 smallest counting errors predicted by the error predictor.

4.2. Evaluation Metrics

We use Mean Average Error (MAE) and Root Mean Squared Error (RMSE) to measure the performance of different object counters. Besides, we follow [31] to report the Normalized Relative Error (NAE) and Squared Relative Error (SRE). In particular, MAE = $\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$; RMSE = $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$; NAE = $\frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i}$; SRE = $\sqrt{\frac{1}{n}\sum_{i=1}^{n}\frac{(y_i - \hat{y}_i)^2}{y_i}}$ where n is the number of test images, and y_i and $\hat{y}_i$ are the ground truth and the predicted number of objects for image i respectively.

4.3. Comparing Methods

We compare our method with the previous works on class-agnostic counting. RepRPN-Counter [33] is the only previous class-agnostic counting method that does not require human-annotated exemplars as input. In order to make other exemplar based class-agnostic methods including GMN (General Matching Network [28]), FamNet (Few-shot adaptation and matching Network [34]) and BMNet

Method	Exemplars	Val Set				Test Set			
		MAE	RMSE	NAE	SRE	MAE	RMSE	NAE	SRE
GMN [28]	GT	29.66	89.81	-	-	26.52	124.57	-	-
	RPN	40.96	108.47	-	-	39.72	142.81	-	-
FamNet+ [34]	GT	23.75	69.07	0.52	4.25	22.08	99.54	0.44	6.45
	RPN	42.85	121.59	0.75	6.94	42.70	146.08	0.74	7.14
BMNet [38]	GT	19.06	67.95	0.26	4.39	16.71	103.31	0.26	3.32
	RPN	37.26	108.54	0.42	5.43	37.22	143.13	0.41	5.31
BMNet+ [38]	GT	15.74	58.53	0.27	6.57	14.62	91.83	0.25	2.74
	RPN	35.15	106.07	0.41	5.28	34.52	132.64	0.39	5.26
RepRPN-Counter [33]	-	30.40	98.73	-	-	27.45	129.69	-	-
Ours (Base)	GT	18.55	61.12	0.30	3.18	20.68	109.14	0.36	7.63
	RPN	32.19	99.21	0.38	4.80	29.25	130.65	0.35	4.35
	Patch-Selection	**26.93**	**88.63**	**0.36**	**4.26**	**22.09**	**115.17**	**0.34**	**3.74**

Table 1. Quantitative comparisons on the FSC-147 dataset. "GT" denotes using human-annotated boxes as exemplars. "RPN" denotes using the top-3 RPN proposals with the highest objectness scores as exemplars. "Patch-Selection" denotes using our selected patches as exemplars.

(Bilinear Matching Network [38]) work in the exemplar-free setup, we replace the human-provided exemplars with the exemplars generated by a pre-trained object detector. Specifically, we use the RPN of Faster RCNN pre-trained on MS-COCO dataset and select the top-3 proposals with the highest objectness score as the exemplars. We also include the performance of these methods using human-annotated exemplars for a complete comparison.

4.4. Results

Quantitative results. As shown in Table 1, our proposed method outperforms the previous exemplar-free counting method [33] by a large margin, resulting in a reduction of 10.10 *w.r.t.* the validation RMSE and 14.52 *w.r.t.* the test RMSE. We also notice that the performance of all exemplar-based counting methods drops significantly when replacing human-annotated exemplars with RPN generated proposals. The state-of-the-art exemplar-based method BMNet+ [38], for example, shows an 19.90 error increase *w.r.t.* the test MAE and a 40.81 increase *w.r.t.* the test RMSE. In comparison, the performance gap is much smaller when using our selected patches as exemplars, as reflected by a 1.41 increase *w.r.t.* the test MAE and a 6.03 increase *w.r.t.* the test RMSE. Noticeably, the NAE and the SRE on the test set are even reduced when using our selected patches compared with the human-annotated exemplars.

Qualitative analysis. In Figure 4, we present a few input images, the image patches selected by our method, and the corresponding density maps. Our method effectively identifies the patches that are suitable for object counting. The density maps produced by our selected patches are meaningful and close to the density maps produced by human-annotated patches. The counting model with random image patches as exemplars, in comparison, fails to output meaningful density maps and infers incorrect object counts.

5. Analyses

5.1. Ablation Studies

Our proposed patch selection method consists of two steps: the selection of class-relevant patches via a generated class prototype and the selection of the optimal patches via an error predictor. We analyze the contribution of each step quantitatively and qualitatively. Quantitative results are in Table 2. We first evaluate the performance of our baseline, i.e. using 3 randomly sampled patches as exemplars without any selection step. As shown in Table 2, using the class prototype to select class-relevant patches reduces the error rate by 7.19 and 6.07 on the validation and test set of MAE, respectively. Applying the error predictor can improve the baseline performance by 7.22 on the validation MAE and 7.57 on the test MAE. Finally, applying the two components together further boosts performance, achieving 26.93 on the validation MAE and 22.09 on the test MAE.

We provide further qualitative analysis by visualizing the selected patches. As shown in Figure 5, for each input query image, we show 10 class-relevant patches selected using our generated prototype, ranked by their predicted counting error (from low to high). All the 10 selected class-relevant patches exhibit some class specific features. However, not all these patches are suitable to be used as counting exemplars, i.e., some patches only contain parts of the object, and some patches contain some background. By further applying our proposed error predictor, we can identify the most suitable patches with the smallest predicted counting errors.

5.2. Generalization to Exemplar-based Methods

Our proposed method can be considered as a general patch selection method that is applicable to other visual counters to achieve exemplar-free counting. To verify that, we use our selected patches as the exemplars for three

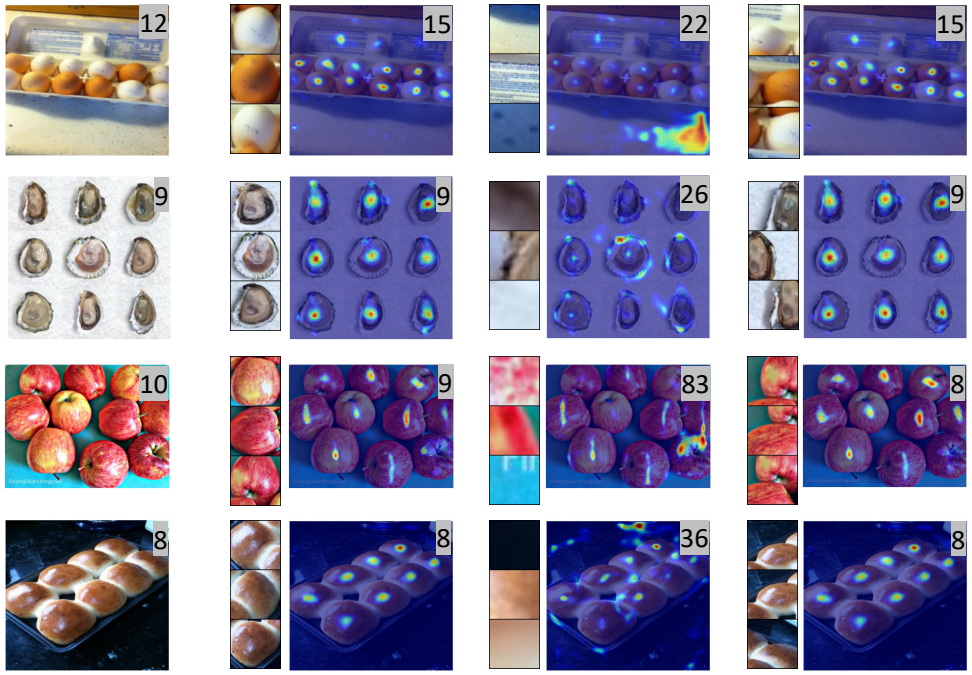


Figure 4. Qualitative results on the FSC-147 dataset. We show the counting exemplars and the corresponding density maps of ground truth boxes, randomly selected patches, and our selected patches respectively. Predicted counting results are shown at the top-right corner. Our method accurately identifies suitable patches for counting and the predicted density maps are close to the ground truth density maps.

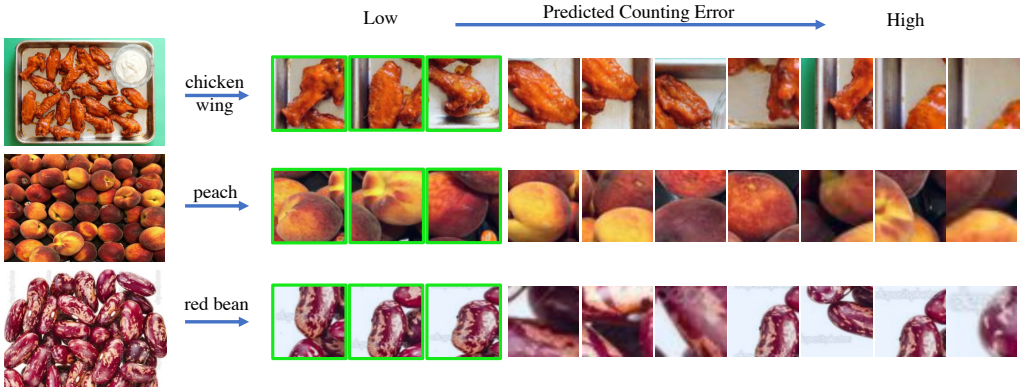


Figure 5. Qualitative ablation analysis. All the 10 selected class-relevant patches exhibit some class-specific attributes. They are ranked by the predicted counting errors and the final selected patches with the smallest errors are framed in green.

Prototype	Predictor	Val Set				Test Set			
		MAE	RMSE	NAE	SRE	MAE	RMSE	NAE	SRE
-	-	35.20	106.70	0.61	6.68	31.37	134.98	0.52	5.92
✓	-	28.01	88.29	0.39	4.66	25.30	**113.82**	0.40	4.88
-	✓	27.98	**88.62**	0.43	4.59	23.80	128.36	0.40	4.43
✓	✓	**26.93**	88.63	**0.36**	**4.26**	**22.09**	115.17	**0.34**	**3.74**

Table 2. Ablation study on each component's contribution to the final results. We show the effectiveness of the two steps of our framework: selecting class-relevant patches via a generated class prototype and selecting optimal patches via an error predictor.

other different exemplar-based methods: FamNet [34], BMNet and BMNet+ [38]. Figure 6 (a) shows the results on the FSC-147 validation set. The baseline uses three randomly sampled patches as the exemplars for the pre-trained exemplar-based counter. By using the generated class prototype to select class-relevant patches, the error rate is reduced by 5.18, 8.59 and 5.60 on FamNet, BMNet and BMNet+, respectively. In addition, as the error predictor is additionally adopted, the error rate is further reduced by 1.76, 1.00 and 1.08 on FamNet, BMNet and BMNet+, respectively. Similarly, Figure 6 (b) shows the results on the FSC-

147 test set. Our method achieves consistent performance improvements for all three methods.

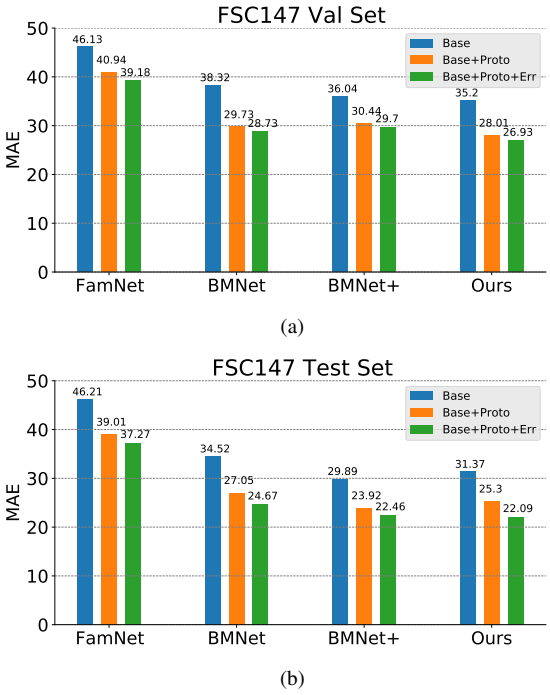


Figure 6. Using our selected patches as exemplars for other exemplar-based class-agnostic counting methods (FamNet, BMNet and BMNet+) on FSC-147 dataset. Blue bars are the MAEs of using three randomly sampled patches. Orange bars are the MAEs of using the class prototype to select class-relevant patches as exemplars. Green bars are the MAEs of using the class prototype and error predictor to select optimal patches as exemplars.

5.3. Multi-class Object Counting

Our method can count instances of a specific class given the class name, which is particularly useful when there are multiple classes in the same image. In this section, we show some visualization results in this multi-class scenario. As seen in Figure 7, our method selects patches according to the given class name and count instances from that specific class in the input image. Correspondingly, the heatmap highlights the image regions that are most relevant to the specified class. Here the heatmaps are obtained by correlating the exemplar feature vector with the image feature map in a pre-trained ImageNet feature space. Note that we mask out the image region where the activation value in the heatmap is below a threshold for counting purpose. We also show the patches selected using another exemplar-free counting method, RepRPN [33]. The class of RepRPN selected patches can not be explicitly specified. It simply selects patches from the class with the highest number of instances in the image according to the repetition score.

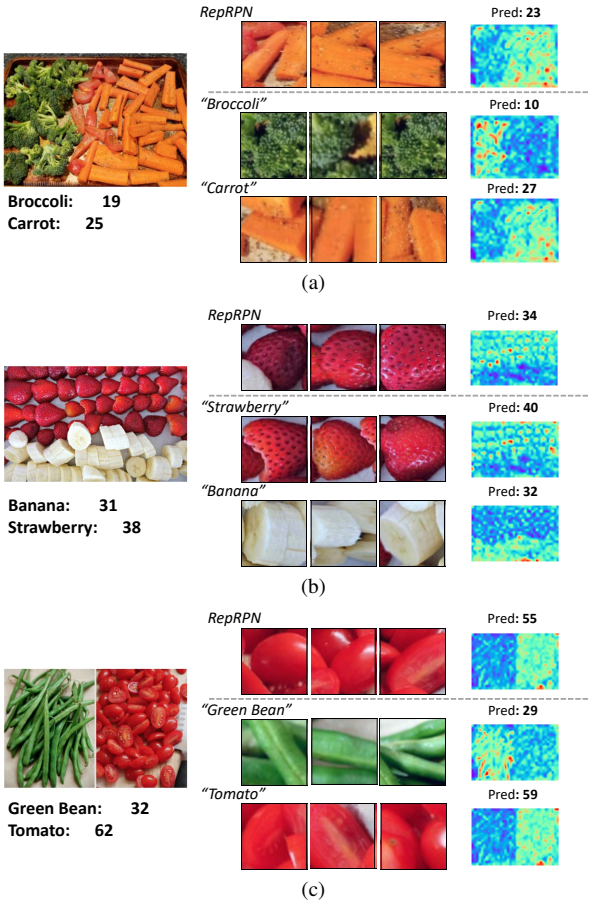


Figure 7. Visualization results of our method in some multi-class examples. Our method selects patches according to the given class name and the corresponding heatmap highlights the relevant areas.

6. Conclusion

In this paper, we proposed a new task, zero-shot object counting, to count instances of a specific class given only the class name without access to any exemplars. To address this, we developed a simple yet effective method that accurately localizes the optimal patches across the query image that can be used as counting exemplars. Specifically, we construct a class prototype in a pre-trained feature space and use the prototype to select patches that contain objects of interest; then we use an error predictor to select those patches with the smallest predicted errors as the final exemplars for counting. Extensive results demonstrate the effectiveness of our method. We also conduct experiments to show that our selected patches can be used for other exemplar-based counting methods to achieve exemplar-free counting.

Acknowledgements. This research was partially supported by NSF grants IIS-2123920 and IIS-2212046 and the NASA Biodiversity program (Award 80NSSC21K1027).

References

[1] Shahira Abousamra, Minh Hoai, Dimitris Samaras, and Chao Chen. Localization in the crowd with topological constraints. In *AAAI*, 2021. 2

[2] Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. In *ICML*, 2017. 3

[3] Carlos Arteta, Victor S. Lempitsky, Julia Alison Noble, and Andrew Zisserman. Interactive object counting. In *ECCV*, 2014. 2

[4] Carlos Arteta, Victor S. Lempitsky, and Andrew Zisserman. Counting in the wild. In *ECCV*, 2016. 1, 2

[5] Yuval Atzmon and Gal Chechik. Adaptive confidence smoothing for generalized zero-shot learning. In *CVPR*, 2019. 3

[6] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-shot object detection. In *ECCV*, 2018. 2

[7] Antoni B. Chan, Zhang-Sheng John Liang, and Nuno Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *CVPR*, 2008. 2

[8] Prithvijit Chattopadhyay, Ramakrishna Vedantam, Ramprasaath R. Selvaraju, Dhruv Batra, and Devi Parikh. Counting everyday objects in everyday scenes. *CVPR*, 2017. 2

[9] Long Chen, Hanwang Zhang, Jun Xiao, W. Liu, and Shih-Fu Chang. Zero-shot visual recognition using semantics-preserving adversarial embedding networks. In *CVPR*, 2018. 3

[10] Hisham Cholakkal, Guolei Sun, Fahad Shahbaz Khan, and Ling Shao. Object counting and instance segmentation with image-level supervision. In *CVPR*, 2019. 2

[11] Hisham Cholakkal, Guolei Sun, Salman Hameed Khan, Fahad Shahbaz Khan, Ling Shao, and Luc Van Gool. Towards partial supervision for generic object counting in natural scenes. volume 44, 2022. 2

[12] Andrea Frome, Gregory S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013. 3

[13] Shenjian Gong, Shanshan Zhang, Jian Yang, Dengxin Dai, and Bernt Schiele. Class-agnostic object counting robust to intraclass diversity. In *ECCV*, 2022. 2

[14] Meng-Ru Hsieh, Yen-Liang Lin, and Winston H. Hsu. Drone-based object counting by spatially regularized regional proposal network. In *ICCV*, 2017. 2

[15] Haroon Idrees, Muhmmad Tayyab, Kishan Athrey, Dong Zhang, Somaya Ali Al-Maadeed, Nasir M. Rajpoot, and Mubarak Shah. Composition loss for counting, density map estimation and localization in dense crowds. In *ECCV*, 2018. 2

[16] Dinesh Jayaraman and Kristen Grauman. Zero-shot recognition with unreliable attributes. In *NIPS*, 2014. 3

[17] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. *CVPR*, 2009. 3

[18] Issam H. Laradji, Negar Rostamzadeh, Pedro H. O. Pinheiro, David Vázquez, and Mark W. Schmidt. Where are the blobs: Counting by localization with point supervision. In *ECCV*, 2018. 2

[19] Hieu Le, Bento Goncalves, Dimitris Samaras, and Heather Lynch. Weakly labeling the antarctic: The penguin colony case. In *CVPR Workshops*, June 2019. 3

[20] Hieu Le and Dimitris Samaras. Physics-based shadow image decomposition for shadow removal. Los Alamitos, CA, USA. IEEE Computer Society. 3

[21] Hieu Le and Dimitris Samaras. From shadow segmentation to shadow removal. In *ECCV*, 2020. 3

[22] Hieu Le, Tomas F. Yago Vicente, Vu Nguyen, Minh Hoai, and Dimitris Samaras. A+D Net: Training a shadow detector with adversarial shadow attenuation. In *ECCV*, 2018. 3

[23] Hieu Le, Chen-Ping Yu, Gregory Zelinsky, and Dimitris Samaras. Co-localization with category-consistent features and geodesic distance propagation. In *ICCV Workshop*, 2017. 3

[24] Dongze Lian, Jing Li, Jia Zheng, Weixin Luo, and Shenghua Gao. Density map regression guided detection network for rgb-d crowd counting and localization. *CVPR*, 2019. 2

[25] Chang Liu, Yujie Zhong, Andrew Zisserman, and Weidi Xie. Countr: Transformer-based generalised visual counting. In *BMVC*, 2022. 2

[26] Weizhe Liu, N. Durasov, and P. Fua. Leveraging self-supervision for cross-domain crowd counting. In *CVPR*, 2022. 2

[27] Weizhe Liu, Mathieu Salzmann, and Pascal V. Fua. Context-aware crowd counting. In *CVPR*, 2019. 2

[28] Erika Lu, Weidi Xie, and Andrew Zisserman. Class-agnostic counting. In *ACCV*, 2018. 1, 2, 5, 6

[29] Terrell N. Mundhenk, Goran Konjevod, Wesam A. Sakla, and Kofi Boakye. A large contextual dataset for classification, detection and counting of cars with deep learning. In *ECCV*, 2016. 1, 2

[30] Sanath Narayan, Akshita Gupta, Fahad Shahbaz Khan, Cees G. M. Snoek, and Ling Shao. Latent embedding feedback and discriminative features for zero-shot classification. In *ECCV*, 2020. 3

[31] Thanh Nguyen, Chau Pham, Khoi Nguyen, and Minh Hoai. Few-shot object counting and detection. In *ECCV*, 2022. 2, 5

[32] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 4, 5

[33] Viresh Ranjan and Minh Hoai. Exemplar free class agnostic counting. In *ACCV*, 2022. 1, 2, 3, 5, 6, 8

[34] Viresh Ranjan, Udbhav Sharma, Thua Nguyen, and Minh Hoai. Learning to count everything. In *CVPR*, 2021. 1, 2, 3, 5, 6, 7

[35] Mahdi Rezaei and Mahsa Shahidi. Zero-shot learning and its applications from autonomous vehicles to covid-19 diagnosis: A review. In *Intelligence-Based Medicine*, volume 3, 2020. 3

[36] Bernardino Romera-Paredes and Philip H. S. Torr. An embarrassingly simple approach to zero-shot learning. In *ICML*, 2015. 3

[37] Deepak Babu Sam, Abhinav Agarwalla, Jimmy Joseph, Vishwanath A. Sindagi, R. Venkatesh Babu, and Vishal M. Patel. Completely self-supervised crowd counting via distribution matching. In *ECCV*, 2022. 1, 2

[38] Min Shi, Hao Lu, Chen Feng, Chengxin Liu, and Zhiguo Cao. Represent, compare, and learn: A similarity-aware framework for class-agnostic counting. In *CVPR*, 2022. 1, 2, 3, 6, 7

[39] Vishwanath A. Sindagi, Rajeev Yasarla, and Vishal M. Patel. Pushing the frontiers of unconstrained crowd counting: New dataset and benchmark method. In *ICCV*, 2019. 2

[40] Jia Wan, Ziquan Liu, and Antoni B. Chan. A generalized loss function for crowd counting and localization. In *CVPR*, 2021. 2

[41] Boyu Wang, Huidong Liu, Dimitris Samaras, and Minh Hoai Nguyen. Distribution matching for crowd counting. In *NeurIPS*, 2020. 2

[42] Qi Wang, Junyu Gao, Wei Lin, and Xuelong Li. Nwpu-crowd: A large-scale benchmark for crowd counting and localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43, 2021. 2

[43] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh N. Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. In *CVPR*, 2016. 3

[44] Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41, 2019. 3, 4

[45] Yongqin Xian, Saurabh Sharma, Bernt Schiele, and Zeynep Akata. F-vaegan-d2: A feature generating framework for any-shot learning. In *CVPR*, 2019. 3, 4

[46] 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, 2018. 1, 2

[47] Haipeng Xiong and Angela Yao. Discrete-constrained regression for local counting models. In *ECCV*, 2022. 2

[48] Jingyi Xu and Hieu Le. Generating representative samples for few-shot classification. In *CVPR*, 2022. 3

[49] Jingyi Xu, Hieu Le, Mingzhen Huang, ShahRukh Athar, and Dimitris Samaras. Variational feature disentangling for fine-grained few-shot classification. In *ICCV*, 2021. 3

[50] Shuo Yang, Hung-Ting Su, Winston H. Hsu, and Wen-Chin Chen. Class-agnostic few-shot object counting. In *WACV*, 2021. 2

[51] Zhiyuan You, Kai Yang, Wenhan Luo, Xin Lu, Lei Cui, and Xinyi Le. Few-shot object counting with similarity-aware feature enhancement. In *WACV*, 2023. 2

[52] Anran Zhang, Lei Yue, Jiayi Shen, Fan Zhu, Xiantong Zhen, Xianbin Cao, and Ling Shao. Attentional neural fields for crowd counting. In *ICCV*, 2019. 2

[53] Cong Zhang, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. Cross-scene crowd counting via deep convolutional neural networks. In *CVPR*, 2015. 2

[54] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a deep embedding model for zero-shot learning. In *CVPR*, 2017. 3

[55] Qi Zhang and Antoni Chan. Calibration-free multi-view crowd counting. In *ECCV*, 2022. 2

[56] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *CVPR*, 2016. 2

[57] Ye Zheng, Jiahong Wu, Yongqiang Qin, Faen Zhang, and Li Cui. Zero-shot instance segmentation. In *CVPR*, 2021. 2