

Learning Spatial Similarity Distribution for Few-shot Object Counting

Yuanwu Xu, Feifan Song, Haofeng Zhang*

School of Artificial Intelligence, Nanjing University of Science and Technology
 {xuyuanwu, sff, zhanghf}@njjust.edu.cn

Abstract

Few-shot object counting aims to count the number of objects in a query image that belong to the same class as the given exemplar images. Existing methods compute the similarity between the query image and exemplars in the 2D spatial domain and perform regression to obtain the counting number. However, these methods overlook the rich information about the spatial distribution of similarity on the exemplar images, leading to significant impact on matching accuracy. To address this issue, we propose a network learning Spatial Similarity Distribution (SSD) for few-shot object counting, which preserves the spatial structure of exemplar features and calculates a 4D similarity pyramid point-to-point between the query features and exemplar features, capturing the complete distribution information for each point in the 4D similarity space. We propose a Similarity Learning Module (SLM) which applies the efficient center-pivot 4D convolutions on the similarity pyramid to map different similarity distributions to distinct predicted density values, thereby obtaining accurate count. Furthermore, we also introduce a Feature Cross Enhancement (FCE) module that enhances query and exemplar features mutually to improve the accuracy of feature matching. Our approach outperforms state-of-the-art methods on multiple datasets, including FSC-147 and CARPK. Code is available at <https://github.com/CBalance/SSD>.

1 Introduction

Visual object counting aims at counting how many times a certain object occurs in the query image, which has received growing attention in the past years. Existing methods often focus on specific domains, such as crowd counting [Shu *et al.*, 2022; Wang *et al.*, 2020; Abousamra *et al.*, 2021], animal counting [Arteta *et al.*, 2016], or car counting [Hsieh *et al.*, 2017]. These methods typically rely on large amounts of data to train accurate counting models. Furthermore, they are

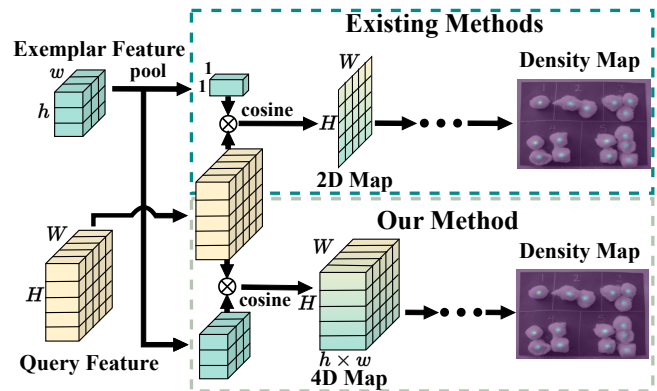


Figure 1: Comparison between existing methods and our method. Compared to the feature similarity computation process in previous methods, our approach preserves the spatial structure of exemplars. Each position is computed with query features, and in the subsequent convolutional regression process, we fully utilize the spatial similarity distribution information between query and exemplar features at a point-to-point level.

limited to counting objects of specific categories and cannot generalize well to novel categories.

To overcome these limitations, a recent approach called Few-shot Object Counting (FSC) has been introduced and gained great attention with the emergence of a dataset [Ranjan *et al.*, 2021]. FSC addresses the challenge of counting objects from arbitrary categories using only a few exemplars. This enables the model to generalize to unseen categories, offering potential for application in various scene categories beyond those encountered during training. By leveraging few exemplars, FSC provides a more flexible and adaptable solution for object counting tasks.

As shown in Fig.1, existing few-shot object counting methods typically follow a general workflow. They first calculate the similarity between query and exemplar features, and then directly regress the similarity matrix or enhance the query features using the similarity matrix and exemplar features before regression. In terms of similarity computation, some methods, as demonstrated in [Ranjan *et al.*, 2021; Yang *et al.*, 2021; You *et al.*, 2023; Đukić *et al.*, 2023], employ exemplar features as fixed kernels to perform convolution with query feature. However, in this approach, the

*Corresponding author.

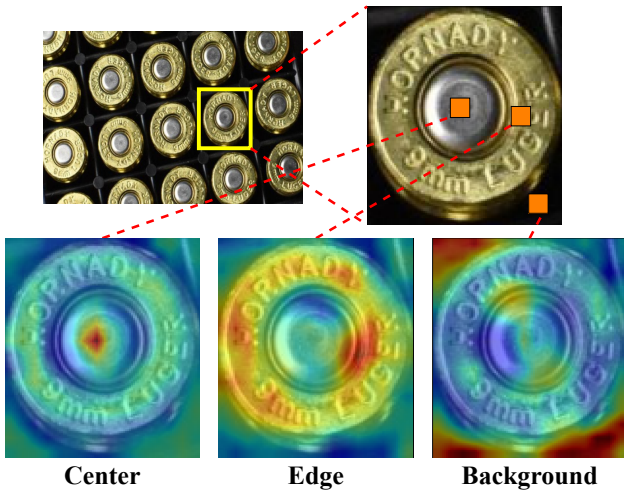


Figure 2: Heatmap depicting the similarity distribution of objects at different positions on the exemplar.

distribution of kernel features remains fixed during convolution matching, limiting its adaptability to different sizes and shapes of object features in the query. Another approach, such as the one used in [Shi *et al.*, 2022; Lin *et al.*, 2022; Liu *et al.*, 2022], involves pooling exemplar features to obtain 1×1 feature prototypes, followed by cosine similarity computation between feature vector of each position in the query and these prototypes. This method disregards the distribution information of query and exemplar features, and counting performance becomes dependent on the performance of previous feature extraction and self-attention mechanisms.

To accurately locate the center of an object and generate an appropriate density distribution, we leverage distinct similarity distribution characteristics between each parts, such as object centers, edges and background, when compared to exemplars. Explicitly, as shown in Fig.2, the similarity distribution of the object center in exemplars gradually diminishes from the central position towards the surrounding regions, while the similarity distribution at the edges exhibits variations across different locations. On the other hand, the background demonstrates generally lower similarity values across all positions except the background area. Taking advantage of these patterns, we propose a novel method that tries to preserve the spatial structure of exemplars during similarity computation, and name it as learning Spatial Similarity Distribution (SSD). Concretely, this method yields a 4D similarity tensor, which allows for flexible extraction of point-to-point similarity distribution information between query and exemplar features using convolution operations in the 4D space. The features obtained through convolution enable precise calculation of density values for each position in the query during regression. In addition, we introduce a Feature Cross Enhancement (FCE) module for query and exemplar features. This employs the similarity matrices as weights to mutually enhance the features, aiming to achieve higher matching accuracy for objects belonging to the given category.

We conduct comprehensive experiments on two renowned public benchmark datasets, *i.e.*, FSC-147 [Ranjan *et al.*,

2021] and CARPK [Hsieh *et al.*, 2017]. The results clearly illustrate that our approach surpasses the performance of current state-of-the-art methods. Our contributions can be summarized as follows:

- We design a model based on learning the 4D spatial similarity distribution between query and exemplar features in Similarity Learning Module (SLM). This model is capable of obtaining accurate counting results after comprehensive integration of similarity distribution information among point pairs and their surroundings.
- Before calculating the similarity between query and exemplar features, we introduce a Feature Cross Enhancement (FCE) module, which enhances the interaction between them, reducing the distance between the target objects and exemplar features to achieve better matching performance.
- Extensive experiments on large-scale counting benchmarks, such as FSC-147 and CARPK, are conducted and the results demonstrate that our method outperforms the state-of-the-art approaches.

2 Related Work

2.1 Class-Specific Object Counting

Class-specific object counting focuses on counting a specific class of objects, such as crowd [Stewart *et al.*, 2016; Liang *et al.*, 2023; Lin and Chan, 2023; Du *et al.*, 2023], animals [Arteta *et al.*, 2016], or cars [Hsieh *et al.*, 2017]. In related methods, the class information can be incorporated into the feature extraction process without additional classification steps. Existing methods can be broadly categorized into detection-based and regression-based approaches.

Detection-based methods detect the positions of objects in an image to perform counting. However, counting accuracy in these methods relies heavily on the performance of the detection process, which introduces errors. This limits the effectiveness of counting tasks in scenarios with densely packed objects. To address this issue, regression-based methods have been proposed to generate a density map, where the sum of the density values represents the predicted object count.

Classic detection-based methods, for example, [Stewart *et al.*, 2016] propose a model that decodes an image into a set of people detections, generating distinct detection hypotheses directly from the input image. On the other hand, recent research in regression-based methods, such as [Cheng *et al.*, 2022], utilizes locally connected multivariate Gaussian kernels as replacements for convolution filters. Moreover, a recent work [Liang *et al.*, 2023] proposes knowledge transfer from a vision-language pre-trained model (CLIP) to unsupervised crowd counting tasks, eliminating the need for density map annotation.

2.2 Few-shot Object Counting

In recent years, few-shot object counting (FSC) has gained significant attention and witnessed a surge of interest. FSC aims to accurately count objects in an image by leveraging only a few exemplars as references. This ability to adapt to

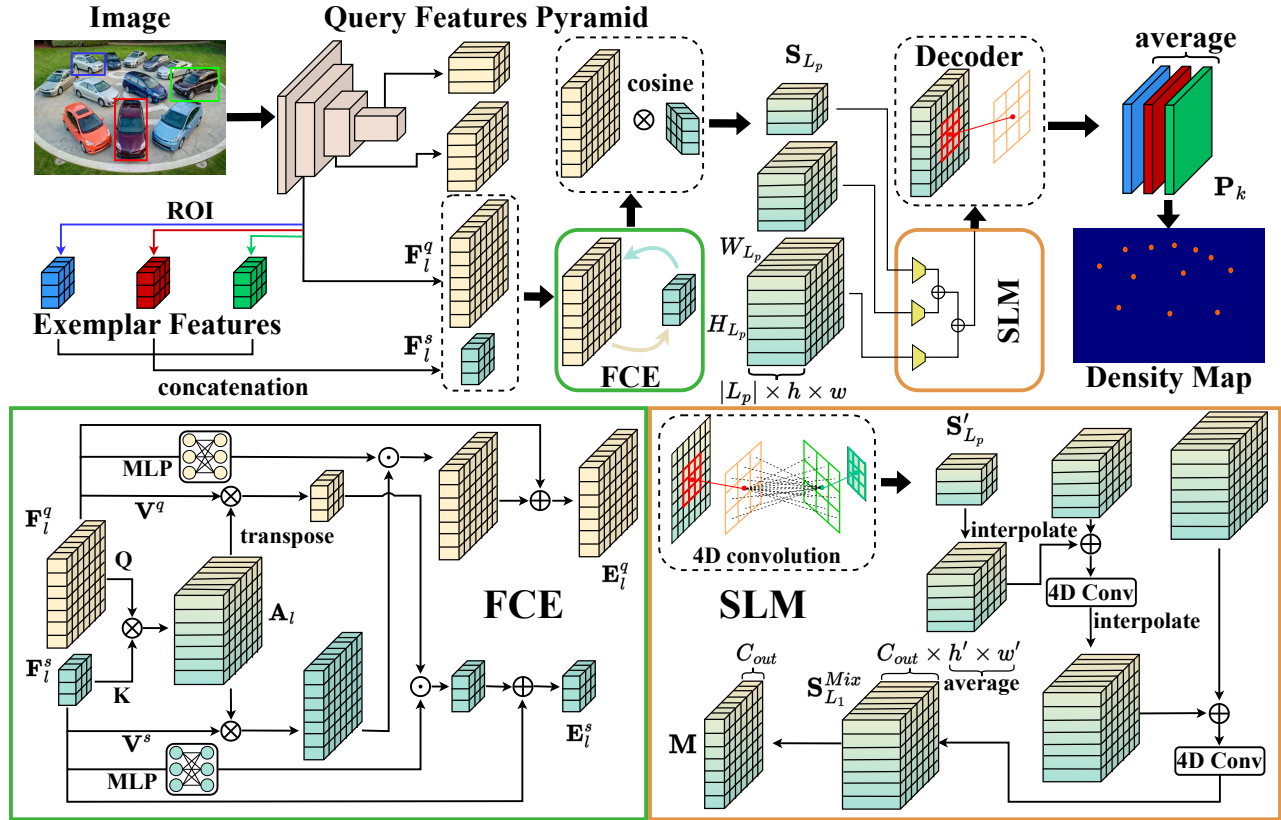


Figure 3: The whole architecture of the proposed SSD framework.

unseen categories during the testing phase is a key advantage of FSC.

Several noteworthy methods have been proposed for FSC. GMN [Lu *et al.*, 2019] concatenates support features and query features together, and regresses a predicted density map based on this concatenation. In contrast, FamNet [Ranjan *et al.*, 2021] convolves the query image with exemplars used as convolutional kernels, generating multiple similarity maps that provide insights into the comparison results between the query and exemplars. A predicted density map is then regressed from these similarity maps. Another approach, BMNet [Shi *et al.*, 2022], employs global pooling to transform exemplars into prototypes, and replaces fixed inner product operations with a learnable bilinear similarity metric for comparing exemplar prototypes with query image features. CounTR [Liu *et al.*, 2022] introduced a transformer-based architecture for extracting image features and utilized cross-attention modules for effective feature matching. Recently, LOCA [Đukić *et al.*, 2023] is proposed and considers the exemplar shape and appearance properties separately and iteratively adapts these into object prototypes by a new object prototype extraction (OPE) module considering the image-wide features.

2.3 Generalized Loss

Generalized loss function [Wan *et al.*, 2021] is proposed for learning density maps for crowd counting and localization,

which is based on unbalanced optimal transport. And [Wan *et al.*, 2021] prove that both L2 loss and Bayesian loss [Ma *et al.*, 2019] are special cases of the generalized loss. The approach proposed in [Lin *et al.*, 2022] also utilizes this loss function and introduces a scale-sensitive generalized loss that applies different loss computation methods to object categories of different scales.

3 Methodology

3.1 Problem Setting

In few-shot object counting, the dataset is split into base classes C_{base} and novel classes C_{novel} , where C_{base} and C_{novel} do not overlap. The remarkable generalization capability of Few-shot Object Counting (FSC) lies in its ability to achieve high performance on the val set and test set, even for categories C_{novel} that have not been encountered during training on C_{base} . FSC addresses the task of counting the number of objects of interest present in a query image $\mathbf{X} \in \mathbb{R}^{3 \times H \times W}$, with the assistance of K exemplars \mathbf{Z} .

3.2 Overall Architecture

As shown in Fig.3, our entire framework follows the following steps: (1) Feature extraction, (2) Feature Cross Enhancement (FCE), (3) Similarity pyramid calculation, (4) Similarity learning and (5) Regression decoder. Initially, the ResNet-50 [He *et al.*, 2016] feature extractor is used to ex-

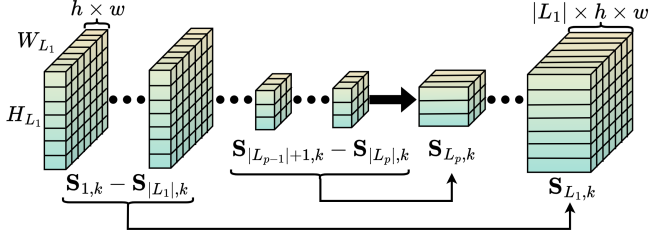


Figure 4: Concatenation of multi-level similarity matrices.

tract features from the image $\mathbf{X} \in \mathbb{R}^{3 \times H \times W}$, with the option of weight freezing (no fine-tuning is performed), and generated pyramid features $\{\mathbf{F}_l^q\}_{l=1}^L$, where each level feature $\mathbf{F}_l^q \in \mathbb{R}^{C_{L_p} \times H_{L_p} \times W_{L_p}}$ ($l \in L_p$). Among all levels, several adjacent levels have features with the same spatial dimensions. All these levels together form a large layer L_p ($p = 1, 2, \dots, P$). For each level in the feature pyramid, K exemplar features $\mathbf{F}_l^s \in \mathbb{R}^{K \times C_{L_p} \times h \times w}$ are extracted using the RoIAlign method [He *et al.*, 2017]. C_{L_p} represents the feature channel dimension of the respective layer, while $H_{L_p} \times W_{L_p}$ and $h \times w$ denote the spatial dimensions of the query feature and exemplar features at that layer. Here we maintain all exemplar features at a uniform size $h \times w$.

The K exemplar features are input into the FCE module, along with the query feature at the same level, resulting in enhanced features \mathbf{E}_l^q and \mathbf{E}_l^s with the same dimensions as \mathbf{F}_l^q and \mathbf{F}_l^s . We partition the K exemplars, with each exemplar corresponding to a set of feature pyramid combinations $\{\mathbf{E}_l^q, \mathbf{E}_{l,k}^s\}_{l=1}^L$, where $\mathbf{E}_{l,k}^s \in \mathbb{R}^{C_{L_p} \times h \times w}$. For pyramid feature set of each exemplar, we perform cosine multiplication on each feature pair to generate the similarity matrix:

$$S_{l,k}(x^q, x^s) = \text{ReLU} \left(\frac{\mathbf{E}_l^q(x^q) \cdot \mathbf{E}_{l,k}^s(x^s)}{\|\mathbf{E}_l^q(x^q)\| \|\mathbf{E}_{l,k}^s(x^s)\|} \right). \quad (1)$$

Here, x^q and x^s denote 2-dimensional spatial positions of query feature map \mathbf{E}_l^q and exemplar feature map $\mathbf{E}_{l,k}^s$, respectively. ‘ \cdot ’ denotes vector dot product. For each similarity tensor, $\mathbf{S}_{l,k} \in \mathbb{R}^{H_{L_p} \times W_{L_p} \times h \times w}$.

As shown in Fig. 4, we concatenate the similarity matrices of the same large layer L_p and partition $\{\mathbf{S}_{l,k}\}_{l=1}^L$ accordingly, transforming it into $\{\{\mathbf{S}_{l,k}\}_{l \in L_p}\}_{p=1}^P$ and set it as $\{\mathbf{S}_{L_p,k}\}_{p=1}^P$. And each tensor in it is $\mathbf{S}_{L_p,k} \in \mathbb{R}^{|L_p| \times H_{L_p} \times W_{L_p} \times h \times w}$, where $|L_p|$ is the number of pyramid levels. They are then fed into the Similarity Learning Module (SLM) to produce a learned and fused feature map \mathbf{M}_k . Finally, \mathbf{M}_k is input into the regression decoder module to obtain the density map $\mathbf{P}_k \in \mathbb{R}^{1 \times H \times W}$. The K sets of feature pyramids correspond to the generation of K density maps. The final predicted density map, denoted as \mathbf{P} , is obtained by taking the mean of these K maps:

$$\mathbf{P} = \frac{\sum_{k=1}^K \mathbf{P}_k}{K}. \quad (2)$$

3.3 Feature Cross Enhancement

The distribution of object features within the query features of the same category is often uneven. Directly matching and counting using the original features can result in varying density values for each object. To address this issue, we propose a Feature Cross Enhancement (FCE) module that aims to bring the object features within the query closer to the exemplar features while also facilitate the exemplar features to be closer to the center position of all object features. By enhancing the proximity of the object features specific to a certain category, the model is able to generate more accurate density values.

In the FCE module, the input features \mathbf{F}_l^q and \mathbf{F}_l^s are jointly transformed into $\mathbf{V}^q \in \mathbb{R}^{C_p^e \times H_{L_p} \times W_{L_p}}$ and $\mathbf{V}^s \in \mathbb{R}^{C_p^e \times K \times h \times w}$ through a convolutional layer. They are then individually passed through other two convolutional layers, with \mathbf{F}_l^q being transformed into \mathbf{Q} and \mathbf{F}_l^s into \mathbf{K} , which are the same dimensions as \mathbf{V}^q and \mathbf{V}^s . Multiplying the transpose of \mathbf{Q} and \mathbf{K} matrices results in the attention matrix \mathbf{A}_l :

$$\mathbf{A}_l = \text{SoftMax}(\mathbf{Q}^T \mathbf{K}). \quad (3)$$

Then we utilize \mathbf{A}_l to separately enhance \mathbf{F}_l^q and \mathbf{F}_l^s :

$$\begin{aligned} \mathbf{E}_l^q &= \mathbf{F}_l^q + \text{MLP}(\mathbf{F}_l^q) \odot \text{Trans}(\mathbf{V}^s \mathbf{A}_l^T) \\ \mathbf{E}_l^s &= \mathbf{F}_l^s + \text{MLP}(\mathbf{F}_l^s) \odot \text{Trans}(\mathbf{V}^q \mathbf{A}_l). \end{aligned} \quad (4)$$

Here, $\text{MLP}(\cdot)$ is a multi-layer perceptron consisting of fully connected layers and activation functions, and used to map the channel vector into a channel-wise feature space of similarity relation. $\text{Trans}(\cdot)$ represents the convolutional layer that transforms channel C_p^e into the original channel C_{L_p} , and \odot denotes element-wise multiplication.

3.4 Similarity Learning Module

4D convolution. Several existing works [Rocco *et al.*, 2018; Yang and Ramanan, 2019; Min *et al.*, 2021] have proposed various implementations of 4D convolutions. In our framework, we employ the center-pivot 4D convolution from [Min *et al.*, 2021] which sparsifies a significant portion of unimportant weights and computations. This method focuses solely on the information associated with the convolution center, reducing computational overhead while maintaining effectiveness. With 4D convolutions, tensors are fused for each 4D position based on convolution kernel weights, integrating information from the vicinity in 4D space and transforming the vector at that position into the corresponding output dimension.

For the input set of similarity tensors $\{\mathbf{S}_{L_p}\}_{p=1}^P$ (here we omit the exemplar subscript k), each tensor is fed into its corresponding 4D convolutional module:

$$\mathbf{S}'_{L_p} = f_{L_p}^e(\mathbf{S}_{L_p}) \in \mathbb{R}^{C_{out} \times H_{L_p} \times W_{L_p} \times h' \times w'}, \quad (5)$$

where $f_{L_p}^e(\cdot)$ is an encoding module composed of multiple 4D convolutional layers, group normalization [Wu and He, 2018], and ReLU activation function. The large strides of the 4D convolution compresses the spatial dimensions $h \times w$ of the exemplar spatial structure to $h' \times w'$, while embedding the dimensions of all similarity tensors from $|L_p|$ into C_{out} .

Next, starting from the apex of pyramid $\left\{ \mathbf{S}'_{L_p} \right\}_{p=1}^P$, we proceed to fuse each subsequent layer downwards. For instance, the tensor $\mathbf{S}'_{L_p} \in \mathbb{R}^{C_{out} \times H_{L_p} \times W_{L_p} \times h' \times w'}$ is upsampled on its dimensions $H_{L_p} \times W_{L_p}$ to match the corresponding dimensions $H_{L_{p-1}} \times W_{L_{p-1}}$ of the layer below. It is then added to the respective tensor $\mathbf{S}'_{L_{p-1}}$ of the layer below and passed through a fusion module based on 4D convolution:

$$\mathbf{S}_{L_{p-1}}^{Mix} = f_{L_{p-1}}^{Mix} \left(\text{upsample} \left(\mathbf{S}'_{L_p} \right) + \mathbf{S}'_{L_{p-1}} \right). \quad (6)$$

The structure of function $f_{L_{p-1}}^{Mix}$ is identical to that of function $f_{L_p}^e$, with the difference being that the stride of $f_{L_{p-1}}^{Mix}$ is set to 1, which does not alter the spatial dimensions of the tensor. And the input and output dimensions in $f_{L_{p-1}}^{Mix}$ are all set to C_{out} .

$\mathbf{S}_{L_{p-1}}^{Mix}$ is fused with the tensor $\mathbf{S}'_{L_{p-2}}$ in a similar manner, iteratively continuing the fusion process with each subsequent layer until reaching the bottom layer of the pyramid \mathbf{S}'_{L_1} . Consequently, we obtain the final fused tensor $\mathbf{S}_{L_1}^{Mix} \in \mathbb{R}^{C_{out} \times H_{L_1} \times W_{L_1} \times h' \times w'}$. By calculating the mean along the last two dimensions, we derive the fused feature $\mathbf{M} \in \mathbb{R}^{C_{out} \times H_{L_1} \times W_{L_1}}$.

3.5 Regression Decoder

The decoder module used for regression consists of multiple component modules composed of convolutional layers, ReLU activation layers, and upsampling layers. With each component module, the size of feature \mathbf{M} is increased to twice until reaching the size of the input image $H \times W$. Subsequently, it passes through a 1×1 convolutional layer and a ReLU activation layer. The output is the predicted density map.

3.6 Generalized Loss

In previous object counting tasks, ground truth density maps are generated by convolving dot labels with fixed Gaussian kernels. The MSE loss function is then employed for supervised training of the predicted density map. In a recent study [Wan *et al.*, 2021], a generalized loss function was proposed that directly measures the distance between the predicted density map and the dot labels. This loss function is based on entropic-regularized unbalanced optimal transport cost.

We represent the predicted results as $\mathbf{A} = \{(a_i, \mathbf{x}_i)\}_{i=1}^n$, where a_i denotes the predicted density value at pixel $\mathbf{x}_i \in \mathbb{R}^2$. Here, n represents the total number of pixels. Then we denote the predicted density map as $\mathbf{a} = [a_i]_i$. On the other hand, the ground truth dot label is denoted as $\mathbf{B} = \{(b_j, \mathbf{y}_j)\}_{j=1}^m$, where \mathbf{y}_j indicates the location of the j -th annotation and b_j represents the number of objects represented by that annotation. In general, it is assumed that $\mathbf{b} = [b_j]_j = \mathbf{1}_m$. The whole loss function can be defined as:

$$L_C(\mathbf{A}, \mathbf{B}) = \min_{\mathbf{D}} \langle \mathbf{C}, \mathbf{D} \rangle - \varepsilon H(\mathbf{D}) + \tau \|\mathbf{D}\mathbf{1}_m - \mathbf{a}\|_2^2 + \tau \|\mathbf{D}^T \mathbf{1}_n - \mathbf{b}\|_1, \quad (7)$$

where \mathbf{C} is the transport cost matrix, with C_{ij} representing the cost of moving the predicted density at \mathbf{x}_i to the

ground truth dot annotation \mathbf{y}_j . \mathbf{D} is the transport matrix that assigns fractional weights to associate each location \mathbf{x}_i in \mathbf{A} with its corresponding \mathbf{y}_j in \mathbf{B} for cost calculation. The optimal transport cost is obtained by minimizing the loss over \mathbf{D} . $H(\mathbf{D}) = -\sum_{ij} D_{ij} \log D_{ij}$ is the entropic regularization. The intermediate density map representation $\hat{\mathbf{a}} = \mathbf{D}\mathbf{1}_m$ is constructed from the ground truth annotations, while $\hat{\mathbf{b}} = \mathbf{D}^T \mathbf{1}_n$ is the reconstruction of the ground truth dot annotations.

3.7 Dynamic Image Scale

Certain sample images may contain objects that are small sizes or densely distributed, leading to challenges in effectively distinguishing between individual objects. This results in overlapping density within the predicted density map, thereby impacting model performance. To address this issue, we dynamically resize the input images based on the size of exemplar boxes before entering the backbone. This resizing is performed proportionally to the dimensions of the exemplar boxes, allowing the model to better recognize samples containing smaller objects. For an input image \mathbf{X} , we compute the average length and width of K exemplar boxes $\mathbf{B} \in \mathbb{R}^{K \times 2}$:

$$\bar{\mathbf{B}} = \frac{\sum_{k=1}^K \mathbf{B}_k}{K}. \quad (8)$$

If $\min(\bar{\mathbf{B}})$ is below a threshold γ , we calculate the scale of image expansion:

$$scale = \frac{\gamma - \min(\bar{\mathbf{B}})}{\eta} + 1, \quad (9)$$

where both γ and η are hyperparameters to be tuned. Finally, the image size and exemplar boxes \mathbf{B} are simultaneously expanded by the determined scale value before being input into the model.

4 Experiments

4.1 Datasets and Metrics

Datasets. **FSC-147** is a comprehensive multi-class few-shot object counting dataset. It comprises a total of 6,135 images covering 89 distinct object categories. The images in the dataset exhibit significant variations in terms of object counts, ranging from as low as 7 to as high as 3,731 objects, with an average count of 56 per image. Notably, each image in the dataset is accompanied by three or four exemplar images that are annotated with bounding boxes. To facilitate experimentation, the dataset is further divided into training, validation, and testing subsets, with each subset containing 29 non-overlapping object categories.

CARPK is a class-specific car counting dataset, which consists of 1448 images of parking lots from a bird's view. These images are captured from four different parking lots, encompassing various scenes. The training set comprises three scenes, while a separate scene is designated for test.

Metrics. We employ Mean Average Error (MAE) and Root Mean Squared Error (RMSE) as performance metrics

Methods	Backbone	3shot				1shot			
		Val		Test		Val		Test	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
GMN [Lu <i>et al.</i> , 2019]	ResNet-50	29.66	89.81	26.52	124.57	—	—	—	—
MAML [Finn <i>et al.</i> , 2017]	ConvNet	25.54	79.44	24.90	112.68	—	—	—	—
FamNet [Ranjan <i>et al.</i> , 2021]	ResNet-50	23.75	69.07	22.08	99.54	26.55	77.01	26.76	110.95
CFOCNet [Yang <i>et al.</i> , 2021]	ResNet-50	21.19	61.41	22.10	112.71	27.82	71.99	28.60	123.96
LaoNet [Lin <i>et al.</i> , 2021]	VGG-19	—	—	—	—	17.11	56.81	15.78	97.15
BMNet+ [Shi <i>et al.</i> , 2022]	ResNet-50	15.74	58.53	14.62	91.83	17.89	61.12	16.89	96.65
SAFECOUNT [You <i>et al.</i> , 2023]	ResNet-18	15.28	47.20	14.32	85.54	—	—	—	—
SPDCN [Lin <i>et al.</i> , 2022]	VGG-19	14.59	49.97	13.51	96.80	—	—	—	—
CounTR [Liu <i>et al.</i> , 2022]	ViT/ConvNet	13.13	49.83	11.95	91.23	13.15	49.72	12.06	90.01
LOCA [Đukić <i>et al.</i> , 2023]	ResNet-50	10.24	32.56	10.79	56.97	11.36	38.04	12.53	75.32
SSD(ours)	ResNet-50	9.73	29.72	9.58	64.13	11.03	34.83	11.61	71.55

Table 1: Comparison with state-of-the-art approaches on the FSC-147 dataset. ‘—’ means the result is not reported.

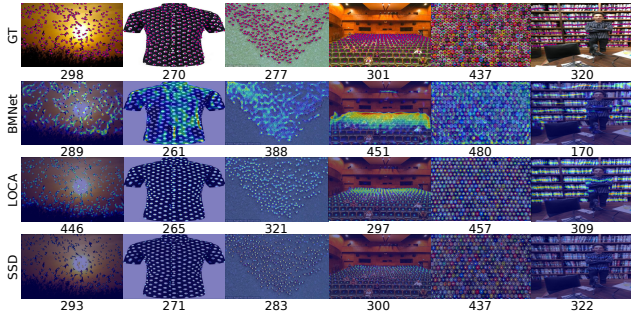


Figure 5: Qualitative results on the FSC-147 dataset.

for evaluating the SSD method, as these metrics are widely utilized in counting tasks.

$$\begin{aligned}
 MAE &= \frac{1}{N} \sum_{i=1}^N |C_{pred}^i - C^i|, \\
 RMSE &= \sqrt{\frac{1}{N} \sum_{i=1}^N (C_{pred}^i - C^i)^2},
 \end{aligned} \tag{10}$$

where N is the number of all the query images, C^i and C_{pred}^i are the ground truth and the predicted number of objects for i -th image respectively.

4.2 Implementation Details

Architecture Details. Our approach involves resizing the input image initially to 384×576 . Then the image is dynamically resized to the suitable scale, followed by the application of the pre-trained ResNet-50 backbone, utilizing features extracted from the final three layers. The number of features at each layer, denoted as $|L_p|$, is 4, 6, and 3, with corresponding feature channel dimensions of 512, 1024, and 2048, respectively. The size of exemplar features extracted from each layer is uniformly resized to 16×16 . In the FCE module, the embedded channel dimension is set to half of dimension of the input feature. In the Similarity Learning module, the

4D convolution module consists of three component modules with output dimensions of 32, 128, and 256, respectively. To fuse the three layers of similarity tensors, two fusion modules are required, each containing three component modules, and all the output dimensions are 256. γ and η in DIS method are set to 32 and 12.

Training Details. We apply AdamW [Loshchilov and Hutter, 2017] as the optimizer with a learning rate of 1×10^{-4} and the learning rate decays with a rate of 0.95 after each epoch. The batch size is 4 and the model is trained for 100 epochs.

4.3 Comparison with State of the Art

We evaluate the proposed SSD on the FSC-147 dataset with several state-of-the-art methods. The results are summarized in Tab.1. We conduct both 3-shot and 1-shot experiments on the dataset. SSD consistently outperforms existing methods in terms of performance.

In the 3-shot scenario, even compared to the recent state-of-the-art method LOCA [Đukić *et al.*, 2023], SSD demonstrates superior performance on the val set with a 5.0% improvement in MAE and an 8.7% improvement in RMSE. Notably, SSD also exhibits better performance on the test set with a 11.2% improvement in MAE.

In the 1-shot scenario, SSD surpasses all previous state-of-the-art methods. This underscores the minimal dependence of SSD on reference samples, showcasing its robust adaptability to scenarios with limited available data.

Qualitative Results. In Fig.5, we visualize and compare the predicted density maps of BMNet, LOCA, and SSD. The results demonstrate that SSD has higher accuracy compared to the other two methods.

4.4 Cross-dataset Generalization

Following [Ranjan *et al.*, 2021], we validate the generalizability of SSD on the CARPK dataset. The model is trained on the FSC-147 dataset and then tested on the CARPK dataset, with the car category samples excluded during training. During testing, we randomly select twelve annotations from the CARPK dataset as exemplars to count cars in images. The experimental results is presented in Tab.2. SSD

Methods	BackBone	Test	
		MAE	RMSE
FamNet [Ranjan <i>et al.</i> , 2021]	ResNet-50	28.84	44.47
BMNet [Shi <i>et al.</i> , 2022]	ResNet-50	10.44	13.77
LOCA [Đukić <i>et al.</i> , 2023]	ResNet-50	9.97	12.51
SSD(ours)	ResNet-50	9.58	12.15

Table 2: Comparison with the state-of-the-art approaches on the CARPK dataset.

FCE	G-Loss	DIS	Val		Test	
			MAE	RMSE	MAE	RMSE
✗	✗	✗	18.96	64.44	16.75	108.12
✓	✗	✗	18.56	61.69	16.39	108.64
✗	✓	✗	15.14	54.75	14.99	107.40
✗	✗	✓	13.68	45.53	14.25	91.17
✓	✓	✗	13.92	51.08	14.43	106.84
✗	✓	✓	10.50	31.86	11.38	74.45
✓	✗	✓	13.37	39.26	12.90	82.16
✓	✓	✓	9.73	29.72	9.58	64.13

Table 3: Ablation studies on the FSC-147 dataset. ‘G-Loss’ means Generalized Loss. ‘DIS’ denotes Dynamic Image Scale.

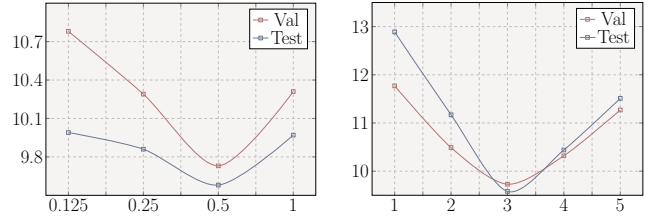
outperforms three other methods, achieving an improvement of 3.9% in MAE and 2.9% in RMSE compared to the most recent state-of-the-art method LOCA.

4.5 Ablation Study

We design a series of experiments to validate the individual contributions of the FCE module, generalized loss, and dynamic image scale on the performance improvement. In the absence of the FCE module, the model directly computes the similarity between \mathbf{F}_i^q and \mathbf{F}_i^s . When excluding the generalized loss, we replace it with the more commonly used MSE loss. Each component undergoes four sets of comparative experiments with and without that component.

FCE module. Analysis of the four sets of experiments involving the FCE module reveals a consistent improvement in model performance. The addition of the FCE module results in a performance boost ranging from 2% to 19% in MAE and up to 14% in RMSE. This indicates that FCE module significantly enhances the ability of model to recognize objects within a given category by minimizing the distance between individual object features and exemplar features, leading to improved accuracy and uniformity in similarity and final density predictions across object positions.

Generalization loss. The contribution of generalization loss is pronounced to performance improvement. The four sets of comparative experiments show performance gains ranging from 10% to 27% in MAE and 1% to 30% in RMSE. The substantial improvement attributed to the generalization loss is due to the more precise recognition capabilities compared to MSELoss. By measuring point-to-point distance loss between predicted values and ground true labels, the generalized loss effectively guides the model to accurately locate object center positions.



(a) Channel ratio in FCE module

(b) Number of component modules in fusion module

Figure 6: Ablation study of channel ratio in FCE module and number of component modules in fusion module. The vertical coordinates are the values of MAE on the val set.

Dynamic image scale. The utilization of dynamic image scale also significantly improves model performance, particularly for dense samples. Expanding image scales proves effective in distinguishing between individual objects and counting them separately. Application of this method results in performance improvements ranging from 15% to 35% in MAE and 14% to 43% in RMSE.

Channel ratio in the FCE module. The query features and example features are embedded into another channel before they enhance each other. We conduct a series of experiments to determine the optimal ratio of the embedded channel length to the original channel length, setting various ratios at $\frac{1}{8}$, $\frac{1}{4}$, $\frac{1}{2}$, and 1. We then train the model to achieve the best performance and present the experimental results in Fig.6 (a). It is observed that as the ratio increases, the model performance peaks at a ratio of $\frac{1}{2}$ and then begins to deteriorate.

Number of component modules in fusion module. The fusion module used to integrate tensors from different levels of the similarity pyramid is composed of several component modules. We set the number of component modules to range from 1 to 5 and conduct experiments on the FSC-147 dataset, with the results displayed in Fig.6 (b). The line graph in the figure indicates that as the number of component modules increases from 1 to 3, the performance of the model gradually improves, peaking at 3, and then begins to decline. This decline could be attributed to an increase in the number of model parameters due to more component modules, leading to overfitting and negatively affecting model performance.

5 Conclusion

We propose a novel few-shot object counting method, SSD, which leverages a point-to-point 4D space to learn the spatial similarity distribution between query and exemplar features. In contrast to existing methods, we exploit the distribution information of similarity, enabling accurate identification of the position and precise prediction of the count for objects of arbitrary classes. Additionally, we introduce a Feature Cross Enhancement (FCE) module that enhances the interaction between query and exemplar features, reducing the feature distance within the same class for improved matching. Experimental results on datasets such as FSC-147 and CARPK demonstrate that SSD outperforms state-of-the-art methods.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China under the Grants 62371235 and 62072246, and in part by Key Research and Development Plan of Jiangsu Province (Industry Foresight and Key Core Technology Project) under the Grant BE2023008-2.

References

- [Abousamra *et al.*, 2021] Shahira Abousamra, Minh Hoai, Dimitris Samaras, and Chao Chen. Localization in the crowd with topological constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 872–881, 2021.
- [Arteta *et al.*, 2016] Carlos Arteta, Victor Lempitsky, and Andrew Zisserman. Counting in the wild. In *Proceedings of the European Conference on Computer Vision*, pages 483–498, 2016.
- [Cheng *et al.*, 2022] Zhi-Qi Cheng, Qi Dai, Hong Li, Jingkuan Song, Xiao Wu, and Alexander G Hauptmann. Rethinking spatial invariance of convolutional networks for object counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19638–19648, 2022.
- [Du *et al.*, 2023] Zhipeng Du, Jiankang Deng, and Miaojing Shi. Domain-general crowd counting in unseen scenarios. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 561–570, 2023.
- [Finn *et al.*, 2017] 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*, pages 1126–1135, 2017.
- [He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [He *et al.*, 2017] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.
- [Hsieh *et al.*, 2017] 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/CVF International Conference on Computer Vision*, pages 4145–4153, 2017.
- [Liang *et al.*, 2023] Dingkan Liang, Jiahao Xie, Zhikang Zou, Xiaoqing Ye, Wei Xu, and Xiang Bai. Crowdclip: Unsupervised crowd counting via vision-language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2893–2903, 2023.
- [Lin and Chan, 2023] Wei Lin and Antoni B Chan. Optimal transport minimization: Crowd localization on density maps for semi-supervised counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21663–21673, 2023.
- [Lin *et al.*, 2021] Hui Lin, Xiaopeng Hong, and Yabin Wang. Object counting: You only need to look at one. *arXiv preprint arXiv:2112.05993*, 2021.
- [Lin *et al.*, 2022] Wei Lin, Kunlin Yang, Xinzhu Ma, Junyu Gao, Lingbo Liu, Shinan Liu, Jun Hou, Shuai Yi, and Antoni Chan. Scale-prior deformable convolution for exemplar-guided class-agnostic counting. In *Proceedings of the British Machine Vision Conference*, 2022.
- [Liu *et al.*, 2022] Chang Liu, Yujie Zhong, Andrew Zisserman, and Weidi Xie. Countr: Transformer-based generalised visual counting. In *Proceedings of the British Machine Vision Conference*, 2022.
- [Loshchilov and Hutter, 2017] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [Lu *et al.*, 2019] Erika Lu, Weidi Xie, and Andrew Zisserman. Class-agnostic counting. In *Proceedings of the Asian Conference on Computer Vision*, pages 669–684, 2019.
- [Ma *et al.*, 2019] Zhiheng Ma, Xing Wei, Xiaopeng Hong, and Yihong Gong. Bayesian loss for crowd count estimation with point supervision. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6142–6151, 2019.
- [Min *et al.*, 2021] Juhong Min, Dahyun Kang, and Minsu Cho. Hypercorrelation squeeze for few-shot segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6941–6952, 2021.
- [Ranjan *et al.*, 2021] Viresh Ranjan, Udbhav Sharma, Thu Nguyen, and Minh Hoai. Learning to count everything. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3394–3403, 2021.
- [Rocco *et al.*, 2018] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. *Advances in neural information processing systems*, 31, 2018.
- [Shi *et al.*, 2022] Min Shi, Hao Lu, Chen Feng, Chengxin Liu, and Zhiguo Cao. Represent, compare, and learn: A similarity-aware framework for class-agnostic counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9529–9538, 2022.
- [Shu *et al.*, 2022] Weibo Shu, Jia Wan, Kay Chen Tan, Sam Kwong, and Antoni B Chan. Crowd counting in the frequency domain. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19618–19627, 2022.
- [Stewart *et al.*, 2016] Russell Stewart, Mykhaylo Andriluka, and Andrew Y Ng. End-to-end people detection in crowded scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2325–2333, 2016.
- [Đukić *et al.*, 2023] Nikola Đukić, Alan Lukežič, Vitjan Zavrtnik, and Matej Kristan. A low-shot object counting network with iterative prototype adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18872–18881, 2023.

- [Wan *et al.*, 2021] Jia Wan, Ziquan Liu, and Antoni B Chan. A generalized loss function for crowd counting and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1974–1983, 2021.
- [Wang *et al.*, 2020] 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(6):2141–2149, 2020.
- [Wu and He, 2018] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.
- [Yang and Ramanan, 2019] Gengshan Yang and Deva Ramanan. Volumetric correspondence networks for optical flow. *Advances in neural information processing systems*, 32, 2019.
- [Yang *et al.*, 2021] Shuo-Diao Yang, Hung-Ting Su, Winston H Hsu, and Wen-Chin Chen. Class-agnostic few-shot object counting. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 870–878, 2021.
- [You *et al.*, 2023] Zhiyuan You, Kai Yang, Wenhan Luo, Xin Lu, Lei Cui, and Xinyi Le. Few-shot object counting with similarity-aware feature enhancement. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6315–6324, 2023.