Towards Partial Supervision for Generic Object Counting in Natural Scenes.

— Source link ↗

Hisham Cholakkal, Guolei Sun, Salman H. Khan, Fahad Shahbaz Khan ...+2 more authors

Institutions: ETH Zurich

Related papers:

- Towards Partial Supervision for Generic Object Counting in Natural Scenes

- A Few-Shot Sequential Approach for Object Counting.

- DiscoBox: Weakly Supervised Instance Segmentation and Semantic Correspondence from Box Supervision

- Leveraging Instance-, Image- and Dataset-Level Information for Weakly Supervised Instance Segmentation.

- Scale-Recursive Network with point supervision for crowd scene analysis

Share this paper: 📘 🐦 💼 ✉

Towards Partial Supervision for Generic Object Counting in Natural Scenes

Hisham Cholakkal*, Guolei Sun*, Salman Khan, Fahad Shahbaz Khan, Ling Shao and Luc Van Gool

Abstract—Generic object counting in natural scenes is a challenging computer vision problem. Existing approaches either rely on instance-level supervision or absolute count information to train a generic object counter. We introduce a partially supervised setting that significantly reduces the supervision level required for generic object counting. We propose two novel frameworks, named lower-count (LC) and reduced lower-count (RLC), to enable object counting under this setting. Our frameworks are built on a novel dual-branch architecture that has an image classification and a density branch. Our LC framework reduces the annotation cost due to multiple instances in an image by using only lower-count supervision for all object categories. Our RLC framework further reduces the annotation cost arising from large numbers of object categories in a dataset by only using lower-count supervision for a subset of categories and class-labels for the remaining ones. The RLC framework extends our dual-branch LC framework with a novel weight modulation layer and a category-independent density map prediction. Experiments are performed on COCO, Visual Genome and PASCAL 2007 datasets. Our frameworks perform on par with state-of-the-art approaches using higher levels of supervision. Additionally, we demonstrate the applicability of our LC supervised density map for image-level supervised instance segmentation.

Index Terms—Generic object counting, Reduced supervision, Object localization, Weakly supervised instance segmentation

1 INTRODUCTION

Common object counting, also referred to as *generic object counting*, seeks to enumerate instances of different object categories in natural scenes. The problem is challenging as common object categories in natural scenes can vary from fruits to animals and counting must be performed in both indoor and outdoor scenes (*e.g.* COCO or Visual Genome datasets [22], [28]). It is a highly valuable task for scene understanding as it allows AI agents to summarize an image by quantifying instances from each object category. Moreover, object count can be used as an additional cue to improve other tasks, such as instance segmentation and object detection. Existing works generally employ a localization-based strategy [24], [42] which locate the object first and then enumerate the located object instances, or utilize regression-based models [7] for directly predicting the object count.

Most localization or regression-based generic object counting approaches [7], [24] require manual annotation of each instance, either in the form of a bounding box or a point (*e.g.* the center of the object). These instance-level annotations are time consuming since they need to be sequentially marked at each instance. The cost of user-intensive, instance-level annotation can be reduced by utilizing image-level supervision, which only requires the instance count of each object category in an image [7]. However, for large object counts, even image-level annotation is laborious since it requires the annotator to sequentially focus on each object

instance. Further, the annotation time increases with the number of object categories to be enumerated. Therefore, existing generic object counting approaches, utilizing either instance or image-level annotations are not scalable for natural scenes that have multiple instances of a large number of object categories (*e.g.* the Visual Genome dataset [22]).

In this work, we propose a partially supervised setting (see Fig. 1) that reduces the annotation effort further than image-level count supervision. Under this partially supervised setting, we reduce the annotation cost due to (i) multiple instances and (ii) large numbers of categories. To reduce the annotation cost due to multiple instances, we propose a novel LC framework based on a *lower-count (LC)* supervision strategy. The LC supervision strategy is motivated by the psychological studies [5], [11], [21], [34] suggesting that humans are capable of counting objects non-sequentially using holistic cues for fewer object counts, termed as the subitizing range or lower-count (generally 1-4). We utilize this property in our LC framework by only using object count supervision within the lower-count range. The proposed LC framework consists of an image classification and a density branch. The image classification branch estimates the presence or absence of objects, whereas the density branch predicts the category-specific object count. In addition, the density branch also predicts the spatial distribution of object instances for each object category, by constructing separate density map for each category. This spatial distribution is later shown to be useful for delineating spatially adjacent object instances in weakly supervised instance segmentation.

The LC supervised setting, introduced above, considers lower-count annotations (≤ 4) for "all object categories". Since there is often a large number of object categories appearing in natural scene datasets (*e.g.* several hundred categories in the Visual Genome dataset [22]), even ob-

- * *Hisham Cholakkal and Guolei Sun contribute equally to this paper.*

- *Hisham Cholakkal, Salman Khan, Fahad Shahbaz Khan and Ling Shao are with Inception Institute of Artificial Intelligence, UAE.*

- *Guolei Sun and Luc Van Gool are with the Computer Vision Laboratory, ETH Zurich, Zurich, Switzerland.*

Visual Genome

	Glancing	LCFCN	LC	RLC
Spoon				
Baseball				
Dessert				
Surfboard	(5)✗	(4)✗	(6)✓	(6)✓
Fan				
Fireplace				
Baseball bat				
⋮				
Person	(9)✗	(14)✗	(11)✓	(11)✓
Button				
Cake				
Glove				
Blade				
Mouse				
TC	?	?	?	(17)✓

	Glancing	LCFCN	LC	RLC
Spoon				
Baseball				
Dessert				
Surfboard				
Fan				
Baseball bat	(1)✓	(1)✓	(1)✓	(1)✓
Fireplace				
⋮				
Person	(8)✓	(7)✗	(8)✓	(8)✓
Button				
Cake				
Glove	(1)✗	(1)✗	(2)✓	(2)✓
Blade				
Mouse				
TC	?	?	?	(11)✓

Microsoft COCO

	Glancing	LCFCN	LC	RLC
Person	(2)✓	(2)✓	(2)✓	(2)✓
Banana				
Remote				
Chair	(0)✗	(1)✗	(2)✓	(2)✓
Umbrella				
Apple				
Sandwich				
Dinner table	(1)✓	(1)✓	(1)✓	(1)✓
⋮				
Orange				
Wine glass	(5)✗	(6)✓	(6)✓	(6)✓
Microwave				
Frisbee				
laptop				
TC	?	?	?	(11)✓

	Glancing	LCFCN	LC	RLC
Person				
Banana	(2)✓	(2)✓	(2)✓	(2)✓
Remote				
Chair				
Umbrella				
Apple	(1)✗	(2)✓	(2)✓	(2)✓
Sandwich				
Dinner table				
⋮				
Orange	(1)✗	(1)✗	(2)✓	(2)✓
Wine glass				
Microwave				
Frisbee				
laptop				
TC	?	?	?	(6)✓

Fig. 1. Generic object counting with partial supervision. Example results are shown on the Visual Genome (*top*) and the COCO (*bottom*) datasets. Due to large number of object categories in the dataset (609 for Visual Genome and 80 for COCO), acquiring accurate count annotations for natural scenes is laborious and costly. We propose two settings to address this, where the first (LC) reduces the annotation cost due to multiple instances and the second (RLC) further reduces the annotation cost due to large numbers of object categories. While both frameworks reduce supervision by only requiring image-level lower-count annotations, the LC framework requires these annotations for all categories, whereas RLC only needs them for a subset (blue means the corresponding categories are not count-annotated during training of RLC framework). Our two approaches (LC and RLC) significantly reduce the annotation cost in comparison to the state-of-the-art instance-level (LCFCN [24]) and image-level (Glancing [7]) supervised methods. Here, the example counting results from two datasets show the generalizability of our frameworks to both object counts beyond the lower-count range (> 4) and to count-unannotated categories. Due to the absence of category-specific counts for some classes, we introduce an additional category-agnostic total count measure in our RLC framework to facilitate generalization across categories and provide accurate category-agnostic total count (TC) predictions. (correct/incorrect/unavailable predictions are marked with ✓/✗/? respectively).

taining LC annotations for all of these object classes is cumbersome. This necessitates the development of an object counting approach that can further reduce the supervision level and generalize accurately to novel object categories. With this objective, we propose a more challenging partially supervised setting for object counting, where image-level classification labels are available for all categories, but lower-count annotations are known for "only a subset" of object categories. We call this setting *reduced lower-count (RLC) supervision*. Since the counting operation looks for repetitive patterns in an image, we hypothesize that an object counter will generalize to categories for which no count annotations are given, enabling object counting under this RLC setting. To address this challenging setting, we propose an RLC framework that extends our dual-branch (image classification and density) LC framework with the following changes. First, we introduce a novel weight modulation layer that effectively transfers knowledge from count-annotated categories to those categories without any count annotation. To further support the generalization of count predictions to these count-unannotated categories, an additional sub-branch is introduced to the density branch for predicting the total count of objects in natural scenes, irrespective of their category.

The LC framework leads to better-quality density maps that can be used for other applications, such as instance-segmentation, whereas the RLC framework scales up object counting to novel categories whose count annotations are not available during training. Fig. 1 shows that our partially supervised approaches (LC and RLC) are able to accurately predict object counts beyond the lower-count range. For example, although the count annotations are not available

for categories like 'surfboard', and 'wine glass', their instance counts beyond the lower-count range are predicted accurately by our RLC framework (see images on the left in Fig. 1). Despite the presence of multiple categories without any count annotations, our RLC framework accurately predicts the total count (TC) of objects in these images.

Contributions: In this work, we introduce a partially supervised setting that can substantially reduce the required level of supervision for generic object counting in natural scenes. Our contributions are as follows: **(a)** We propose two novel frameworks (LC and RLC) for generic object counting. Our LC framework aims at reducing the annotation cost due to multiple instances while the RLC framework targets reducing the annotation cost due to large numbers of object categories and instances in natural scene datasets. The LC framework comprises an image classification and density branch. The RLC framework extends our dual-branch LC framework with a novel weight modulation layer that contributes to the generation of category-specific density maps and is generalizable to categories without count annotations. Moreover, a category-agnostic sub-branch is introduced in the density branch of the proposed RLC framework to generate a category-independent density map and therefore estimates accurate *total object count* for an image. **(b)** We propose LC supervised training of density maps for generic object counting, and demonstrate its applicability for image-level supervised instance segmentation. **(c)** To the best of our knowledge, we are the first to introduce a generic object counting approach that targets the transferability of count prediction to the RLC supervised scenario (where only image-level category labels are available, without count-annotations). **(d)** We extensively evaluate our

approach on multiple datasets, such as Visual Genome [22] and MS-COCO [28]. Our approach obtains state-of-the-art performance for category-specific object counting on these datasets. Moreover, our RLC framework demonstrates impressive results for total and category-specific counts, even in the presence of several categories without any count annotations.

2 RELATED WORK

Generic object counting in natural scenes: Object counting methods in the literature can be categorized as the ones that provide class-wise object counts (*category-specific* counters) and those that provide a total count for all objects (*category-independent* counters). Category-specific object counting in natural scenes has been recently investigated [7], [24]. Chattopadhyay *et al.* [7] proposed object counting strategies that use instance-level and image-level supervisions. The instance-level (bounding box) supervised strategy, denoted as subitizing, estimates a large number of objects by dividing an image into non-overlapping regions, assuming the object count in each region falls within the subitizing, or lower-count, range. The image-level (per-category count) supervised strategy, denoted as glancing, uses a regression loss to train a convolutional neural network that can predict per-category object counts. It uses count annotations from both within and beyond the subitizing or lower-count range to predict the per-category count of each object, without providing information about their location. In constrast, our approach requires neither box annotations nor count information beyond the lower-count range. The work of [24] proposed a point-level supervised approach that requires post-processing steps such as [48] to estimate object counts. Our method does not require such post-processing steps and instead directly predicts the object counts by simple summation of the density maps. Aside from category-specific counting, class-agnostic or category-independent object counting was previously investigated for salient object subitizing in [2], [18], [51], where the category-agnostic counting of salient objects was performed only within the subitizing or lower-count range. In contrast, our approach estimates the object counts both within and beyond the lower-count range.

Another strategy for common-object counting is to count the number of instances detected by an object detector [42]. Although these object detectors are trained for precise localization of object boundaries with bounding-box supervision, the number of instances detected may not match with the ground-truth count, especially at large ground-truth counts [7]. This is mainly due to the fact that optimizing the network for precise localization of each object boundary may result in false negative detections on challenging images with heavy occlusion, and hence under-counting [7]. Adjusting the detector parameters (such as the non-maximum suppression threshold) to address this issue may lead to false positives and hence over-counting. Our method has dedicated count loss terms which enable accurate count predictions under these challenging scenarios.

Crowd counting in surveillance scenes: Crowd counting approaches in surveillance scenes [4], [8], [12], [27], [29], [30], [43], [47], [49], generally aim at counting large numbers of instances from one or few object categories. Most crowd counting approaches [6], [8], [27], [30], [31], [47], require instance-level supervision and use a density map to preserve the spatial distribution of people in an image, obtaining the person count for the image by summing over the predicted density map. Scale variations in surveillance scenes are addressed in [6], [30]. Additional unlabelled data and synthetic data are used in [31], [47], respectively to improve crowd counting. A fully convolutional framework with multiple dilated convolution kernels was proposed in [27]. These dilated convolution kernels increase the receptive fields of deep feature maps, without losing spatial resolution through pooling operations, enabling counting under highly congested scenes. Lu *et al.* [33] proposed a generic matching network that counts the instances of a given object category, using an exemplar patch containing the object of interest. This method uses an image-matching strategy to obtain the instance count of an object category represented by a training exemplar. However, the exemplar need to be similar to the object instances in a test image, hence less accurate for generic object counting in natural scenes having large intra-class variations.

In this paper, we address the problem of large-scale common or generic object counting, where the objective is to enumerate instances from a large number of object categories. Although the number of instances in an individual category is smaller compared to crowd counting, accurately predicting the object count is highly challenging due to the presence of a large number of diverse object categories in the same image. Moreover, dense occlusions, cluttered regions and large intra-class variations in natural scenes increase the complexity of the problem (see Fig. 1). We propose partially supervised (LC or RLC) density map estimation approaches for generic object counting in natural scenes, which predict the category-specific object count while preserving the spatial distribution of objects. Fig. 3 shows examples of the category-specific density maps of natural scenes generated by our partially supervised LC and RLC frameworks.

Generic object counting with limited supervision: In this work, our aim is to address the problem of generic object counting with limited supervision. Our RLC framework is motivated by the transfer function used for instance-mask prediction in [20] and object detection in [19]. Different from these, however, our weight modulation layer modifies the convolution weights of an image classifier branch to estimate category-specific object counts. As a result, the proposed layer can be learned from an entire image, without using instance-level annotations or pooling as in [19], [20]. Although several weak supervision strategies, such as zero-shot learning, semi-supervised learning and weakly supervised learning [14], have been employed for applications such as object detection [41] and semantic segmentation [50], they have not been employed for generic object counting in natural scenes. Zero-shot learning approaches [41] generally require a natural language description of unseen categories, while few-shot learning and semi-supervised learning approaches [44], [52] require a minimum amount of training data for every category. Different to these approaches, our RLC framework adopts the convolution weights learned for the image classification task to obtain the object count of categories without count annotations.

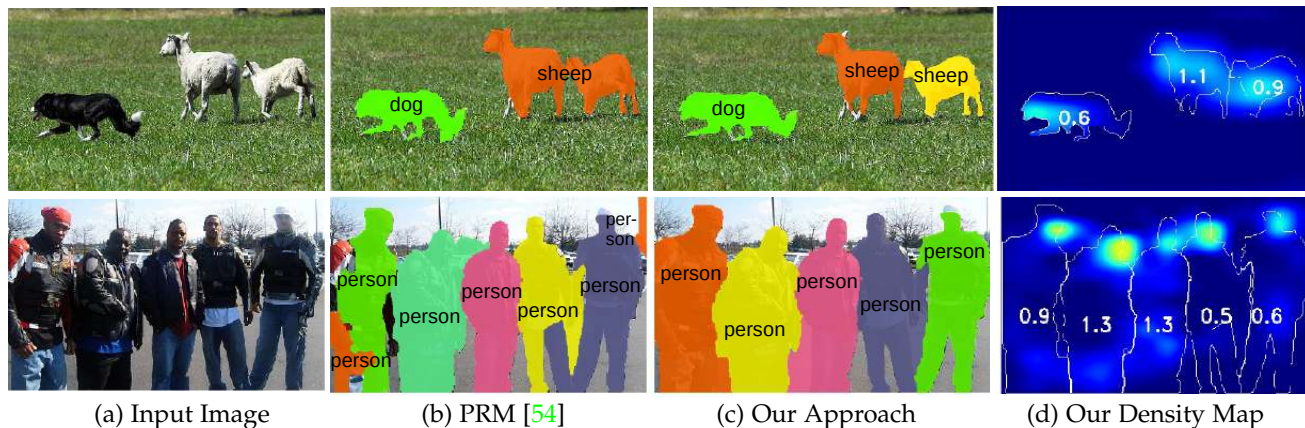|			
(a) Input Image	(b) PRM [54]	(c) Our Approach	(d) Our Density Map

Fig. 2. Examples showing the category-specific density maps, generated by our LC framework, and their usability for improving image-level supervised instance segmentation. We show instance segmentation results using the PRM method [54] (b) and our approach (c), on PASCAL VOC 2012 images. Top row: The PRM approach [54] fails to delineate two spatially adjacent sheep category instances. Bottom row: Single person parts are predicted as multiple persons along with inaccurate mask separation results in over-prediction (7 instead of 5) by the PRM method [54]. The per-category density maps (d) obtained in the lower-count supervised (LC) setting provides the spatial distribution of object count; hence the accumulation of the density map over a local spatial region generally indicates the object count in that region. This property is used to penalize the instance mask predictions containing multiple object instances (object count ≥ 2) or part of objects (object count ≈ 0), and hence to improve weakly supervised instance segmentation (c). The density map accumulation for each predicted mask is shown inside the contour, drawn for clarity. In the top row, the density maps for the sheep and dog categories are overlaid.

Differences from the preliminary version [10]: This paper is an extension of our preliminary work published at the CVPR conference in 2019 [10]. The main contributions of this longer version compared to its preliminary work are the following. The LC supervised setting introduced in the preliminary version [10] requires counts within the lower-count range for '*all*' object categories. Here, we extend this to an even more challenging setting where counts within the lower-count range are known for only a subset of categories during training (RLC supervision). For count prediction in the RLC setting, we introduce a novel weight modulation layer that transfers knowledge from objects with annotated counts to the cases where count information is unknown. Further, the RLC framework introduced in this paper can also predict category-agnostic total object counts, whereas the approach proposed in the preliminary version [10] can only predict category-specific object counts. Moreover, our partially supervised RLC problem setting enables object counting on large-scale datasets that have hundreds of object categories (e.g., Visual Genome), where count annotation for all object categories is difficult to obtain. Compared to [10], additional ablation experiments for the LC framework [10] are performed on the COCO dataset and performance is evaluated on an additional large-scale dataset with a high number of object categories (Visual Genome). Finally, we also perform a thorough evaluation of the proposed RLC framework on the COCO and Visual Genome datasets.

3 GENERIC OBJECT COUNTING WITH PARTIAL SUPERVISION

As mentioned earlier, obtaining object count annotations for natural scenes is expensive due to the presence of: (i) multiple instances, and (ii) a large number of object categories that need to be enumerated. To address these two challenges, we introduce two partially supervised settings with the aim of significantly reducing the annotation cost beyond image-level count annotation. In the first setting, named lower-count (LC) supervision, exact counts are annotated *only until four*, for all object categories, and hence the annotation cost is reduced on images that have a large number of instances. In the second setting, named reduced lower-count (RLC) supervision, we further reduce the supervision such that the LC supervision is only provided for a subset of object categories, and the remaining ones are simply annotated with category labels (*i.e.*, binary labels indicating the presence or absence of object categories). This helps in reducing the annotation costs due to multiple instances and large numbers of object categories. To facilitate object counting under these partially supervised settings (LC and RLC), we introduce a novel dual-branch architecture, where the first branch, named the 'image classification branch', predicts the presence or absence of object categories, while the second branch, named the 'density branch', produces a density map for accurately predicting the object count. Fig. 3 shows examples of the density maps produced by our LC and RLC frameworks. Specifically, Fig. 3 demonstrates that the category-specific density maps produced by our LC framework better preserve spatial information and can be used for object localization applications, such as weakly supervised instance segmentation (see Fig. 2).

As discussed earlier, it is easier to obtain LC supervision when there is a lower number of object categories (*e.g.* 20 categories in the PASCAL VOC dataset). Therefore, our LC framework, that utilizes the LC supervision, is more suited for such datasets. The annotation effort increases with the number of object categories and it is thus difficult to obtain LC supervision when the number becomes too large (*e.g.* several hundred categories in Visual Genome dataset). In such cases, our RLC framework is more suitable since it only requires LC supervision for a subset of categories and category-level annotations for the remaining ones. Further, our RLC framework has the added advantage of having a dedicated sub-branch to predict the category-agnostic total
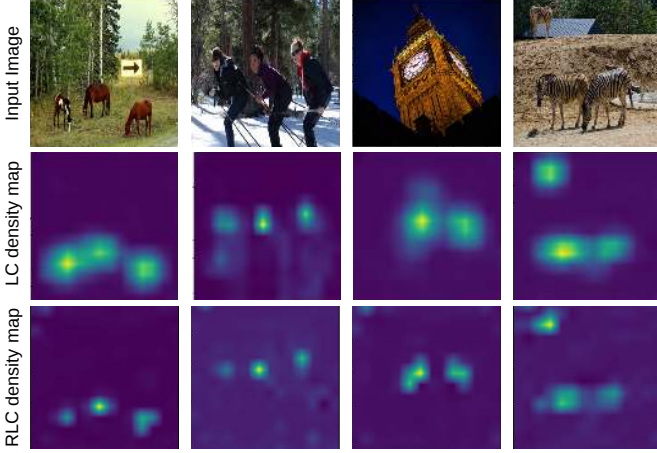
Fig. 3. Example category-specific density maps produced by our LC and RLC frameworks for horse, person, clock and zebra categories, respectively, on images from COCO dataset. Despite being trained using image-level lower-count supervision, the spatial distributions of objects are preserved in both the LC and RLC density maps. Note that, in the case of our RLC framework, only category-level annotations (no lower-count annotations) were available for the clock and zebra categories.

I	Training image
S_0	Set of categories which are absent in an image I
S	Set of categories in the image I that have counts within the lower-count range (*i.e.*, $1 \leq t_c < \tilde{t}$)
$\tilde{S}$	Set of categories in the image I that have counts beyond the lower-count range (*i.e.*, $t_c \geq \tilde{t}$)
C	Total number of object categories in a given dataset
$\mathcal{A}$	Set of categories in the dataset that have both count and category annotations
$\mathcal{B}$	Set of categories in the dataset that have only category annotations
$\mathbf{t}$	A vector indicating the ground-truth count of all C object categories
t_c	Ground-truth count of object category c
$\hat{t}_c$	Predicted count of object category c
$\tilde{t}$	Smallest count for $\tilde{S}$, *i.e.* 5
t_{tot}	Ground-truth category-agnostic total count in an image I
$\hat{t}_{tot}$	Predicted category-agnostic total count in an image I

TABLE 1
Notations used in our partially supervised (LC and RLC) frameworks.

count (TC) of objects in an image. The TC predictions can be further generalized to images containing unsupervised object categories (*i.e.*, without category and count annotations).

Important notations related to our partially supervised experimental settings are shown in Table 1. In the next section, we introduce our novel framework for LC supervision, followed by its extension to the RLC framework in Sec. 5.

4 LOWER-COUNT SUPERVISED FRAMEWORK

Let $\mathbf{I}$ be a training image and $\mathbf{t} = \{t_1, t_2, ..., t_c, ..., t_C\}$ be the corresponding vector for the ground-truth count of C object categories. Instead of using an absolute object count, we employ a lower-count strategy to reduce the amount of image-level supervision. Given an image $\mathbf{I}$, object categories are divided into three non-overlapping sets based on their respective instance counts. The first set, S_0, indicates object categories which are absent in $\mathbf{I}$ (i.e., $t_c = 0$). The second set, S, represents categories within the lower-count range (i.e,

$0 < t_c \leq 4$). The final set, $\tilde{S}$, indicates categories beyond the lower-count range (i.e., $t_c \geq \tilde{t}$, where $\tilde{t} = 5$). Next, we explain the proposed network architecture (LC architecture) for the LC supervised setting.

The Proposed LC Architecture: Our approach is built upon an ImageNet pre-trained network backbone (ResNet50) [17]. The proposed network architecture has two output branches: the image classification and density branches (see Fig. 4). The image classification branch estimates the presence or absence of objects. The density branch predicts the category-specific object count and the spatial distribution of object instances for each object category, by constructing a separate density map for each category. We remove the global average pooling layer from the backbone and preserve the spatial information from the backbone features. The resulting backbone features with 2048 channels are used as a common input to both the image classification and density branches. We then add two 1×1 convolutions in each branch, where the first convolution has P output channels and the second convolution has C output channels, resulting in a fully convolutional network [32]. Here, C is the number of object categories and P is empirically set to be proportional to C. These convolutions are separated by a batch normalization and a ReLU layer. The input features for the last convolution layers of the image classification branch and the density branch are indicated as $\mathbf{F_{cls}}$ and $\mathbf{F_{cnt}}$, respectively (see Fig. 4). The last convolution layers in the image classification and density branches output C spatial maps (corresponding to C object categories), named as object category maps and density maps, respectively. The object category maps help in object localization, while the density maps are useful for estimating the object count in a given spatial region.

The Proposed Loss: Let $\mathbf{M} = \{\mathbf{M}^c \in \mathbb{R}^{H \times W} : c \in [1, C]\}$ denotes the object category maps in the image classification branch and $\mathbf{D} = \{\mathbf{D}^c \in \mathbb{R}^{H \times W} : c \in [1, C]\}$ represents category-specific density maps produced by the density branch. Here, $H \times W$ is the spatial size of both the object category and category-specific density maps. The image classification and density branches are jointly trained, in an end-to-end fashion, given only LC supervision with the following loss function:

$$\mathcal{L} = \mathcal{L}_{class} + \underbrace{\mathcal{L}_{spatial} + \mathcal{L}_{count}}_{\text{Category-specific density branch}}. \qquad (1)$$

Here, the first term refers to the multi-label image classification loss [23] (see Sec. 4.1). The last two terms, $\mathcal{L}_{spatial}$ and $\mathcal{L}_{count}$, are used to train the density branch (Sec. 4.2).

4.1 Image Classification Branch

Generally, training a density map requires instance-level supervision, such as point-level annotations [26]. Such information is unavailable in our LC supervised setting. To address this issue, we propose generating a spatial mask by exploiting the coarse-level localization capabilities of an image classifier [36], [53] via object category maps. These object category maps are generated from a fully convolutional architecture, shown in Fig. 4.

While specifying classification confidence at each image location, class activation maps (CAMs) struggle to delineate
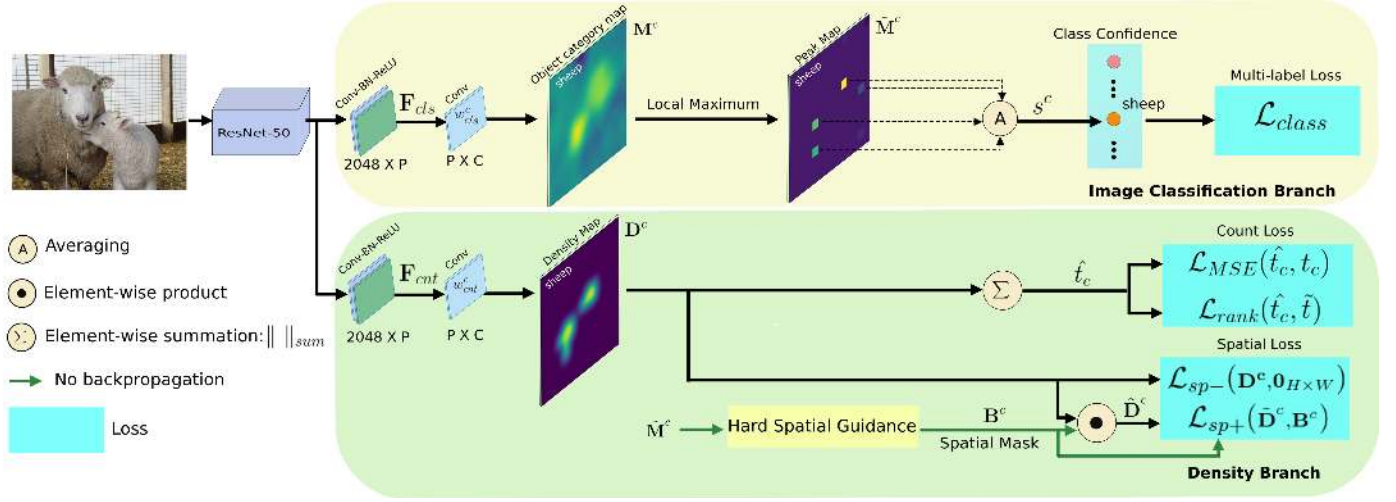
Fig. 4. Overview of our LC architecture with image classification and density branches which are trained jointly using lower-count (LC) supervision. The image classification branch predicts the presence or absence of objects. This branch is used to generate a spatial mask for training the category-specific density branch. The density branch has two terms (spatial and count) in the loss function and produces a category-specific density map to predict the category-specific object count and preserves the spatial distribution of objects.

multiple instances from the same object category [36], [53]. Recently, the local maxima of CAMs were further boosted to produce object category maps, during an image-classifier training for image-level supervised instance segmentation [54]. The boosted local maxima aim at falling on distinct object instances. For details on boosting local maxima, we refer to [54]. Here, we use local maxima locations to generate a spatial mask, which is used as a pseudo ground-truth for training the density branch (Sec. 4.2).

As described earlier, object categories in $\mathbf{I}$ are divided into three non-overlapping sets: S_0, S and $\tilde{S}$. To train a one-versus-rest image classifier, we derive binary labels from t_c that indicate the presence ($c \in \{S, \tilde{S}\}$) or absence ($c \in S_0$) of object categories. Let the peak map $\tilde{\mathbf{M}}^c \in R^{H \times W}$ be derived from the cth object category map ($\mathbf{M}^c$) of $\mathbf{M}$, by using local maxima locations such that:

$$\tilde{\mathbf{M}}^c(i,j) = \begin{cases} \mathbf{M}^c(i,j), & \text{if } \mathbf{M}^c(i,j) > \mathbf{M}^c(i-r_i, j-r_j), \\ 0, & \text{otherwise.} \end{cases}$$

Here, $-r \leq r_i \leq r$ and $-r \leq r_j \leq r$, where r is the radius for the local maxima computation. We set $r = 1$, as in [54]. The local maxima are searched at all spatial locations with a unit stride. To train an image classifier, a class confidence score s^c of the cth object category is computed as the average of non-zero elements of $\tilde{\mathbf{M}}^c$. In this work, we use the multi-label soft-margin loss [23] for binary classification.

4.2 Density Branch

The image classification branch described above predicts the presence or absence of objects using the class confidence scores derived from the peak map $\tilde{\mathbf{M}}^c$. The object category map obtained from the image classifier branch provides a coarse localization of objects, which can provide spatial guidance during the training of our density branch. However, it struggles to differentiate between multiple objects and single-object parts due to the lack of prior information about the number of object instances (see Fig. 2(b)). This causes a large number of false positives in the peak map $\tilde{\mathbf{M}}^c$. To address this issue, we introduce a hard spatial-guidance

module that utilizes the count information and generates a spatial mask from the peak map $\tilde{\mathbf{M}}^c$.

4.2.1 Hard Spatial-guidance

Here, the coarse-localization ability of the object category map is used to generate a spatial mask. For all object categories $c \in S$, the t_c-th highest peak value of peak map $\tilde{M}^c$ is computed using the heap-max algorithm [9]. The t_c-th highest peak value h_c is then used to generate a spatial mask $\mathbf{B}^c$ as,

$$\mathbf{B}^c = u(\tilde{\mathbf{M}}^c - h_c). \tag{2}$$

Here, $u(n)$ is the unit step function which is 1 only if $n \geq 0$. We use the spatial mask as a pseudo-ground truth mask to compute the spatial loss term used for training the density branch. This supports the preservation of the spatial distribution of object counts in a density map. Later, we show that this property helps to improve instance segmentation (Sec. 7.3).

4.2.2 Category-specific Density Map

The density branch produces a category-specific density map $\mathbf{D}^c$, where each pixel indicates the number of objects belonging to category c in the corresponding image region. Additionally, the accumulation of $\mathbf{D}^c$ over any image region estimates the instance count of category c in that region [26]. On the other hand, the pixels in the object category map $\mathbf{M}^c$ indicate the confidence that the corresponding image pixels belong to object category c.

When constructing a density map, it is desired to estimate accurate object counts in every image sub-region. Our spatial loss term $\mathcal{L}_{spatial}$ in Eq. 1 ensures that individual object instances are localized while the count term $\mathcal{L}_{count}$ constrains the category-specific object count to that of the ground-truth count. These terms are explained next.

Spatial Loss: The spatial loss $\mathcal{L}_{spatial}$ is divided into the loss $\mathcal{L}_{sp+}$ which enhances the positive peaks corresponding to instances of object categories within S, and the loss $\mathcal{L}_{sp-}$

which suppresses false positives of categories within S_0. Due to the unavailability of absolute object counts, the set $\tilde{S}$ is not used in the spatial loss and treated separately later. To enable LC supervised density map training using $\mathcal{L}_{spatial}$, we employ the spatial mask $\mathbf{B}^c$ as a pseudo ground-truth mask.

Although the non-zero elements of the spatial mask $\mathbf{B}^c$ indicate object locations, its zero elements do not necessarily point towards the background. Therefore, we construct a masked density map $\tilde{\mathbf{D}}^c$ to exclude density map $\mathbf{D}^c$ values at locations where the corresponding $\mathbf{B}^c$ values are zero. Those density map $\mathbf{D}^c$ values should also be excluded during the loss computation in Eq. 4 and backpropagation, due to their risk of introducing false negatives. This is achieved by computing the Hadamard product between the density map $\mathbf{D}^c$ and $\mathbf{B}^c$ as,

$$\tilde{\mathbf{D}}^c = \mathbf{D}^c \odot \mathbf{B}^c. \tag{3}$$

The spatial loss $\mathcal{L}_{sp+}$ for object categories within the lower-count range S is computed between $\mathbf{B}^c$ and $\tilde{\mathbf{D}}^c$ using a logistic binary cross entropy (logistic BCE) [37] loss for positive ground-truth labels. The logistic BCE loss transfers the network prediction ($\tilde{\mathbf{D}}^c$) through a sigmoid activation layer σ and computes the standard BCE loss as,

$$\mathcal{L}_{sp+}(\tilde{\mathbf{D}}^c, \mathbf{B}^c) = -\sum_{\forall c \in S} \frac{\|\mathbf{B}^c \odot \log(\sigma(\tilde{\mathbf{D}}^c))\|_{sum}}{|S| \cdot \|\mathbf{B}^c\|_{sum}}. \tag{4}$$

Here, $|S|$ is the cardinality of the set S and the norm $\| \|_{sum}$ is computed by taking the summation over all elements in a matrix. For example, $\|\mathbf{B}^c\|_{sum} = \mathbf{1}^h \mathbf{B}^c \mathbf{1}^w$, where $\mathbf{1}^h$ and $\mathbf{1}^w$ are all-ones vectors of size $1 \times H$ and $W \times 1$, respectively. Here, the highest t_c peaks in $\tilde{\mathbf{M}}^c$ are assumed to fall on t_c instances of object category $c \in S$. Due to the unavailability of ground-truth object locations, we use this assumption and observe that it holds in most scenarios.

The spatial loss $\mathcal{L}_{sp+}$ for the positive ground-truth labels enhances positive peaks corresponding to instances of object categories within S. However, the false positives of the density map for $c \in S$ are not penalized in this loss. We therefore introduce another term, $\mathcal{L}_{sp-}$, into the loss function to address the false positives of $c \in S_0$.

For $c \in S_0$, positive activations of $\mathbf{D}^c$ indicate false detections. A zero-valued mask $\mathbf{0}_{H \times W}$ is used as ground-truth to reduce such false detections using the logistic BCE loss,

$$\mathcal{L}_{sp-}(\mathbf{D}^c, \mathbf{0}_{H \times W}) = -\sum_{c \in S_0} \frac{\|\log(1 - \sigma(\mathbf{D}^c)\|_{sum}}{|S_0| \cdot H \cdot W}. \tag{5}$$

Though the spatial loss ensures the preservation of the spatial distribution of objects, only relying on local information may result in deviations in object count.

Count Loss: The count loss penalizes deviations of the predicted count $\hat{t}_c$ from the ground-truth. It has two components: the ranking loss $\mathcal{L}_{rank}$ for object categories beyond the lower-count range (i.e., $\forall c \in \tilde{S}$) and the mean-squared error (MSE) loss $\mathcal{L}_{MSE}$ for the rest of the categories. $\mathcal{L}_{MSE}$ penalizes the predicted density map if the category-specific count prediction does not match with the ground-truth count, i.e.,

$$\mathcal{L}_{MSE}(\hat{t}_c, t_c) = \sum_{c \in \{S_0, S\}} \frac{(\hat{t}_c - t_c)^2}{|S_0| + |S|}. \tag{6}$$

Here, the predicted count $\hat{t}_c$ is the accumulation of the density map for a category c over its entire spatial region, i.e., $\hat{t}_c = \|\mathbf{D}^c\|_{sum}$. Note that object categories in $\tilde{S}$ were not previously considered when computing the spatial loss $\mathcal{L}_{spatial}$ or mean-squared error loss $\mathcal{L}_{MSE}$. Here, we introduce a ranking loss [46] with a zero margin that penalizes under-counting for object categories within $\tilde{S}$,

$$\mathcal{L}_{rank}(\hat{t}_c, \tilde{t}) = \sum_{c \in \tilde{S}} \frac{max(0, \tilde{t} - \hat{t}_c)}{|\tilde{S}|}. \tag{7}$$

The ranking loss penalizes the density branch if the predicted object count $\hat{t}_c$ is less than $\tilde{t}$ for $c \in \tilde{S}$. Recall, the beyond lower-count range $\tilde{S}$ starts from $\tilde{t} = 5$.

Within the lower-count range S, the spatial loss term $\mathcal{L}_{spatial}$ is optimized to locate object instances, while the MSE loss ($\mathcal{L}_{MSE}$) is optimized to accurately predict the corresponding category-specific count. Due to the joint optimization of both these terms within the lower-count range, the network learns to correlate between the located objects and the object count. Further, the network is able to locate object instances, generalizing beyond the lower-count range $\tilde{S}$ (see Fig. 2). Additionally, the ranking loss $\mathcal{L}_{rank}$ term in the proposed loss function ensures the penalization of under-counting for cases beyond the lower-count range $\tilde{S}$.

Mini-batch Loss: The normalized loss terms $\hat{\mathcal{L}}_{sp+}$, $\hat{\mathcal{L}}_{sp-}$, $\hat{\mathcal{L}}_{MSE}$ and $\hat{\mathcal{L}}_{rank}$ are computed by averaging their respective loss terms over all images in the mini-batch. Then, the average spatial loss over a mini-batch, $\mathcal{L}_{spatial}$, is computed by $\hat{\mathcal{L}}_{sp+}[\![c \in S]\!] + \hat{\mathcal{L}}_{sp-}[\![c \in S_0]\!]$, where $[\![\cdot]\!]$ denotes Iverson brackets. For categories beyond the lower-count range, $\hat{\mathcal{L}}_{rank}$ can lead to over-estimation of the count. Hence, the overall count loss $\mathcal{L}_{count}$ is computed by assigning a relatively low weight ($\lambda = 0.1$) to $\hat{\mathcal{L}}_{rank}$ (see Table 3), i.e., $\mathcal{L}_{count} = \hat{\mathcal{L}}_{MSE}[\![c \in S_0, S]\!] + \lambda * \hat{\mathcal{L}}_{rank}[\![c \in \tilde{S}]\!]$.

Backpropagation: We use the spatial mask $\mathbf{B}^c$ derived from the image classification branch as a pseudo ground-truth to train the density branch. Therefore, the backpropagation of gradients through $\mathbf{B}^c$ to the classifier branch is not required (shown with green arrows in Fig. 4). The image classification branch is backpropagated as in [54]. In the density branch, we use the Hadamard product of the density map with $\mathbf{B}^c$ in Eq. 3 to compute $\mathcal{L}_{sp+}$ for $c \in S$. Hence, the gradients (δ^c) for the cth channel of the last convolution layer of the density branch, due to $\mathcal{L}_{sp+}$, are computed as,

$$\delta^c_{sp+} = \frac{\partial \hat{\mathcal{L}}_{sp+}}{\partial \tilde{\mathbf{D}}^c} \odot \mathbf{B}^c. \tag{8}$$

Since $\mathcal{L}_{MSE}$, $\mathcal{L}_{rank}$ and $\mathcal{L}_{sp-}$ are computed using MSE, ranking and logistic BCE losses on convolution outputs, their respective gradients are computed using off-the-shelf Pytorch implementation [37]. Next, we describe the extension of our LC framework to enable object counting in a reduced lower-count (RLC) supervised setting.
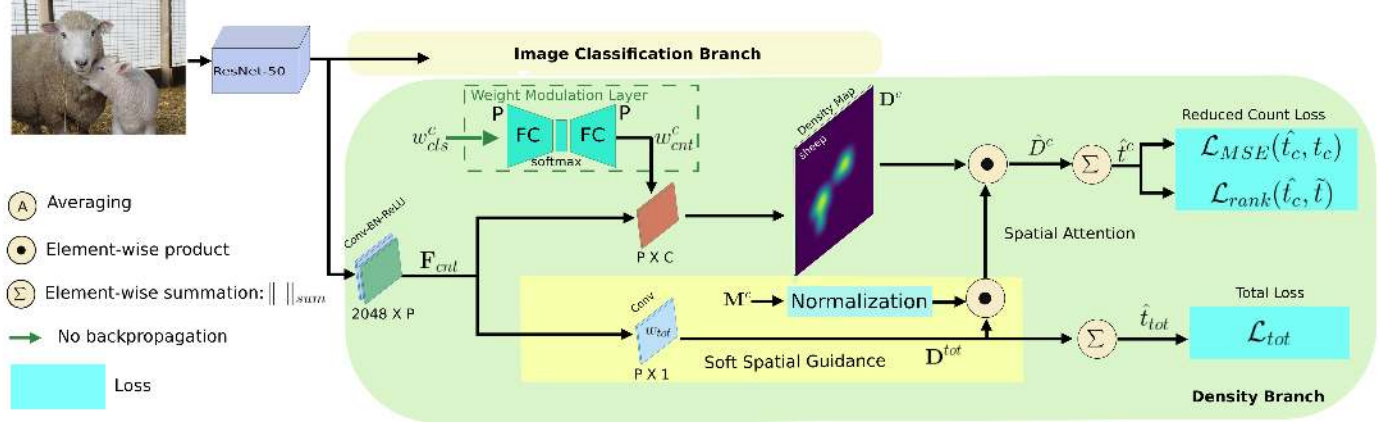
Fig. 5. Overview of our RLC architecture, which comprises an image classification branch and a density branch. The image classification branch has an identical structure as the LC architecture, and is trained on all categories ($\mathcal{A} \cup \mathcal{B}$) using the class labels indicating the presence or absence of the objects. The density branch has two sub-branches: a category-specific and a category-independent density sub-branch. The category-specific sub-branch adapts the convolution weights from the image classifier branch using a weight modulation layer and then generates a category-specific density map, which is used to obtain the category-specific counts. Training this branch updates the weight modulation layer, and only categories with known counts (set $\mathcal{A}$) are used for the training. The convolution operator using adapted convolution weights is shown in red color. The category-independent density sub-branch predicts the total counts of all objects.

5 REDUCED LOWER-COUNT SUPERVISED FRAMEWORK

The LC supervision requires counts within the lower-count range for 'all' object categories. These are laborious to obtain for natural scene datasets that have a large number of object categories (*e.g.*, Visual Genome). Here, we introduce an even more challenging setting, named reduced lower-count (RLC) supervision, which further reduces the level of supervision such that counts within the lower-count range are known for only a subset of categories during training. For this setting, we propose an RLC framework that extends our dual-branch LC framework and predicts counts for *all categories*, irrespective of the count annotations.

In our RLC supervised setting, the set $\mathcal{A}$ indicates the object categories in the *dataset* for which both the category labels and count annotations are known, whereas the set $\mathcal{B}$ indicates the categories whose category labels are known, but count annotations are not available. We employ a LC annotation to the object categories in $\mathcal{A}$. This further splits $\mathcal{A}$ into three disjoint subsets: S_0, S and $\tilde{S}$, based on the *object count in image I* (see Table 1). As mentioned in Sec. 4, the set S_0 indicates the object categories which are absent in image I (*i.e.*, t_c=0). Similarly, the set S represents the categories within the lower-count range ($t_c < \tilde{t}$), where $\tilde{t} = 5$. Finally, the set $\tilde{S}$ indicates the categories beyond the lower-count range ($t_c \geq \tilde{t}$) whose *exact* count annotations are not available. This RLC supervised setting is highly challenging compared to an LC supervised setting (Sec. 4) that has $\mathcal{B} = \emptyset$.

The Proposed RLC Architecture: Similar to the LC architecture, our RLC architecture has an image classification branch and a density branch built on a ResNet50 backbone (See Fig. 5). Here, we explain the changes introduced to our LC architecture (Fig. 4) for count prediction in the challenging RLC setting. First, we introduce a novel weight modulation layer (Sec. 5.1.1) that transfers knowledge from object categories with annotated counts to the categories where count information is unknown. We then introduce an auxiliary category-independent sub-branch to the density branch to produce a category-independent density map $\mathbf{D}^{tot}$ (Sec. 5.1.2). This additionally introduced category-independent density sub-branch provides category-independent total count predictions for an image, including the object categories without count annotations. In summary, the density branch in our RLC framework has two sub-branches; namely, the category-specific sub-branch and category-independent sub-branch, both of which share a common input $\mathbf{F_{cnt}}$, as shown in Fig. 5.

The object category maps (class activation maps with boosted local maxima) used in the image classification branch can provide a coarse localization of object instances from their respective categories. With LC supervision, this coarse-localization ability of the object category map can be used along with the category-specific LC count information, to generate a spatial mask (see Sec. 4.2.1) that preserves the spatial information for the category-specific density map training. However, in RLC supervision, category-specific count annotations are not available for a set of object categories (the set $\mathcal{B}$). Therefore, employing such a hard spatial-guidance strategy is infeasible for obtaining a spatial mask, to train the category-specific density map. To alleviate this limitation, we introduce a soft spatial-guidance strategy that uses the category-independent density map $\mathbf{D}^{tot}$ along with the object category map $\mathbf{M}^c$ of the image classifier to produce a spatial attention map $\mathbf{G}^c$ for the category c. This spatial attention map is used to provide spatial information to the category-specific density map. We train our RLC framework with a novel loss function described below.

The Proposed Loss Function: The complete network, comprising of an image classification and a density branch, is jointly trained in an end-to-end fashion using the proposed objective, given below:

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{rcount} + \mathcal{L}_{tot}, \qquad (9)$$

where $\mathcal{L}_{cls}$ refers to the multi-label classification loss used to train the classification branch. The second and third terms are used to train the density branch, where $\mathcal{L}_{rcount}$ is used to train the category-specific sub-branch, and $\mathcal{L}_{tot}$ is used to train the category-independent one.

Since category labels are available for all object categories, the *image classifier branch* is trained on all object categories using $\mathcal{L}_{cls}$, without introducing any changes compared to Sec. 4.1. Considering that count annotations are only available for set $\mathcal{A}$, we adapt the density branch for RLC supervision, as explained in the next section.

5.1 Density Branch

To address the unavailability of count annotations for set $\mathcal{B}$ in our RLC setting, we introduce a weight modulation layer that adapts the convolution weights from the image classifier branch to generate category-specific density maps that estimate category-specific object counts for *all* categories. In this section, we first introduce this weight modulation layer, followed by the generation of category-specific density maps with RLC supervision.

5.1.1 Weight Modulation Layer

The weight modulation layer Ψ is class-agnostic by nature and is learned using the object categories that have both class and count annotations (set $\mathcal{A}$). It is then used to compute the convolution weights of the category-specific sub-branch for all categories, including those categories without count annotations (set $\mathcal{B}$). Since object counting looks for repetitive patterns in a category-specific projected embedding space, the function Ψ learns a generalizable mapping that works alike for objects with known and unknown count annotations (RLC supervised setting).

Let $\boldsymbol{w}_{cls}^c$ be the convolution weights in the image classification branch (trained using the set $\mathcal{A} \cup \mathcal{B}$) for the object category c, and $\mathbf{D} = \{\mathbf{D}^c \in \mathbb{R}^{H \times W} : c \in [1, C]\}$ represent the category-specific density maps for all C object categories. The weights $\boldsymbol{w}_{cls}^c$ are passed through the weight modulation layer Ψ, resulting in the convolution weights $\boldsymbol{w}_{cnt}^c$ that produce the category-specific density map $\mathbf{D}^c$ for category c, *i.e.*,

$$\boldsymbol{w}_{cnt}^c = \Psi(\boldsymbol{w}_{cls}^c), \text{ and } \mathbf{D}^c = \boldsymbol{w}_{cnt}^c * \mathbf{F_{cnt}}, \qquad (10)$$

where '$*$' denotes convolution. Note that this convolution operator (shown in red in Fig. 5) has no learnable parameters. Instead, it uses the $\boldsymbol{w}_{cnt}^c$ obtained from the $\boldsymbol{w}_{cls}^c$ through the weight modulation layer.

The proposed weight modulation layer is shown in Fig. 5. It passes $\boldsymbol{w}_{cls}^c$ through a $P \times \frac{P}{2}$ fully connected layer, followed by a softmax non-linear layer for normalization, and finally through another $\frac{P}{2} \times P$ fully connected layer. The low-dimensional embedding (to $\frac{P}{2}$) by the first fully connected layer acts as a bottleneck architecture that leads to superior performance. We conjecture that the bottleneck architecture of the modulation layer Ψ projects the convolution weights from similar object categories to similar representations, which enables generalization of the weight modulation layer to the categories whose count annotations are not available. Backpropagation is not performed from Ψ

to the image classifier branch (shown with green colored line in Fig. 5) to avoid creating a discrepancy in $\boldsymbol{w}_{cls}^c$ between $\mathcal{A}$ and $\mathcal{B}$, since only $\boldsymbol{w}_{cls}^c$ for $c \in \mathcal{A}$ will receive gradient signals from Ψ while the image classifier ($\boldsymbol{w}_{cls}^c$) is trained on both $\mathcal{A}$ and $\mathcal{B}$.

5.1.2 Soft Spatial-guidance

The object category maps (class activation maps with boosted local maxima) used in the classification branch can provide a coarse localization of object instances from their respective categories. In our LC setting (Sec. 4.2.1), this coarse-localization ability of object category maps is used along with the category-specific count information to generate a spatial mask that preserves the spatial information for the density map training. Category-specific count information is not available for set $\mathcal{B}$, so such a spatial mask cannot be used in the RLC framework. To alleviate this limitation, the category-independent density map $\mathbf{D}^{tot}$ is used along with the object category map $\mathbf{M}^c$ of the image classifier to produce a spatial attention map $\mathbf{G}^c$ for category c. Considering the large variations in $\mathbf{M}^c$, we pass it through a sigmoid activation and obtain a normalized map $\hat{\mathbf{M}}^c$. The spatial attention for category c, $\mathbf{G}^c$, is computed as,

$$\mathbf{G}^c = \hat{\mathbf{M}}^c \circ \mathbf{D}^{tot}. \qquad (11)$$

The spatial attention map $\mathbf{G}^c$ thus obtained is used with the corresponding density map $\mathbf{D}^c$. Next, we introduce the procedure to generate a category-independent density map that provides total object counts for $\mathbf{I}$.

Total-count Loss and Category-independent Density Map: The category-independent density map $\mathbf{D}^{tot}$ is generated using a category-independent sub-branch (operating on input feature $\mathbf{F_{cnt}}$) that consist of an additional 1×1 convolution with a single output channel. The category-independent total count of objects in an image can be estimated by accumulating $\mathbf{D}^{tot}$ over its entire spatial region, *i.e.*, $\hat{t}_{tot} = \sum_{ij} \mathbf{D}_{ij}^{tot}$. We use only the count-annotated object categories S and $\tilde{S}$ to train the category-independent density branch, and the ground-truth total count t_{tot} for the image I is computed as

$$t_{tot} = \sum_{c \in S} t_c + (\tilde{t} \times |\tilde{S}|). \qquad (12)$$

The category-independent density branch is trained with the following loss function,

$$\mathcal{L}_{tot} = \mathcal{L}_{MSE}(\hat{t}_{tot}, t_{tot})[\![Z = 0]\!] + \mathcal{L}_{rank}(\hat{t}_{tot}, t_{tot})[\![Z > 0]\!]. \qquad (13)$$

Z is the total number of object categories whose exact count is unknown, *i.e.*, $Z = |\tilde{S}| + |\mathcal{B}'|$, where $\mathcal{B}'$ is the set of positive categories of $\mathcal{B}$ in image I. Here, $\mathcal{L}_{MSE}(\hat{t}_{tot}, t_{tot}) = (\hat{t}_{tot} - t_{tot})^2$ is the mean-squared error loss and $\mathcal{L}_{rank}(\hat{t}_{tot}, t_{tot})$ is the ranking loss, such that:

$$\mathcal{L}_{rank}(\hat{t}_{tot}, t_{tot}) = \max(0, t_{tot} - \hat{t}_{tot}). \qquad (14)$$

Next, we describe the generation of category-specific density map.

5.1.3 Category-specific Density Map

As mentioned in Sec. 5.1.1, the category-specific density map $\mathbf{D}^c$ for class c is generated using the modulated convolution weights $\boldsymbol{w}_{cnt}^c$ (obtained using Ψ). We use a category-specific spatial attention $\mathbf{G}^c$ to preserve the spatial information in the density map, resulting in a refined density map $\hat{\mathbf{D}}^c$, as

$$\hat{\mathbf{D}}^c = \mathbf{D}^c \circ \mathbf{G}^c, \tag{15}$$

where '$\circ$' denotes the Hadamard product. The refined density map $\hat{\mathbf{D}}^c$ is used to estimate the category-specific object count $\hat{t}_c$ by accumulating $\hat{\mathbf{D}}^c$ over its entire spatial region, i.e., $\hat{t}_c = \sum_{ij} \hat{\mathbf{D}}_{ij}^c$, where i, j are indices of matrix elements.

The category-specific sub-branch is trained with the *reduced-count loss function* $\mathcal{L}_{rcount}$,

$$\mathcal{L}_{rcount} = \mathcal{L}_{MSE}(\hat{t}_c, \ t_c) [\![c \in S]\!] + \mathcal{L}_{rank}(\hat{t}_c, \ \tilde{t}) [\![c \in \tilde{S}]\!]. \tag{16}$$

Here, $\mathcal{L}_{MSE}$ and $\mathcal{L}_{rank}$ are computed using Eq. 6 and Eq. 7, respectively. It is worth noting that the predicted count $\hat{t}_c$ is obtained from a spatial attention ($\mathbf{G}^c$) weighted density map $\hat{\mathbf{D}}^c$. Hence, the spatial distribution of objects is preserved to a certain degree while minimizing the count error in Eq. 16 (see Fig. 3). Considering that the modulated convolution weight $\boldsymbol{w}_{cnt}^c$ used to generate $\mathbf{D}^c$ is obtained through $\Psi(\boldsymbol{w}_{cls}^c)$, minimizing $\mathcal{L}_{rcount}$ will result in training the class-agnostic weight modulation layer Ψ.

6 TRAINING AND INFERENCE

Throughout our experiments, we use a fixed set of training hyper-parameters. A ResNet-50 backbone architecture pre-trained on the ImageNet dataset is used in our evaluations. The backbone network is trained with an initial learning rate of 10^{-4}, while the image classification and density branches are trained with an initial learning rate of 0.01. The number of input channels P of the final $1{\times}1$ convolutions in each branch is empirically set to $P = 1.5{\times}C$ across datasets, based on experiments performed on the validation sets. A mini-batch size of 16 is used for the SGD optimizer. The momentum and weight decay are set to 0.9 and 10^{-4}, respectively. Our algorithm is implemented using Pytorch [38] on a Tesla V100 GPU. Our code will be made publicly available for reproducible research.

Inference: For both LC and RLC supervision, the image classification branch outputs a class confidence score s^c for each class, indicating the presence ($\hat{t}_c > 0$, if $s^c > 0$) or absence ($\hat{t}_c = 0$, if $s^c \leq 0$) of object category c. With LC, the predicted count $\hat{t}_c$ is obtained by summing the density map $\mathbf{D}^c$ for category c over its entire spatial region, whereas with RLC supervision, $\hat{t}_c$ is obtained from the refined density map $\hat{\mathbf{D}}^c$. The proposed approach only utilizes count annotations within the lower-count range ($t_c \leq 4$) and accurately predicts object counts for *both* within and beyond the lower-count range (see Fig. 7 and Fig. 8). Additionally, the accumulation of the category-independent density map in our RLC architecture leads to a total count for all objects, irrespective of their category (Sec. 5.1.2).

7 EXPERIMENTS

Datasets: We evaluate our method on three challenging datasets: PASCAL VOC 2007 [13], COCO [28] and Visual Genome [22]. For fair comparison with existing approaches on PASCAL VOC 2007 [13] and COCO, we employ the same splits, named as count-train, count-val and count-test, as used in the state-of-the-art methods [7], [24].

Specifically, for the PASCAL VOC dataset, the training, validation and test set are used as count-train, count-val and count-test, respectively. For the COCO dataset, the training set is used as count-train, the first half of the validation set as the count-val, and the second half as count-test. The best models on the count-val set are used to report the results on the count-test set.

Our RLC framework aims at reducing the annotation cost for applications targeting large numbers of object categories. Hence, we only use COCO and Visual Genome (VG) for evaluation since these datasets contain a large number of object categories. The COCO dataset has 80 object categories and Visual Genome (VG) dataset has 80,000 classes. Following [3], we remove non-visual classes from the VG dataset and consider the remaining 609 classes.

To evaluate image-level supervised instance segmentation [1], [25], [54], [55], we train and report the results on the PASCAL VOC 2012 dataset, similar to [54]. Specifically, our model is trained on an augmented set of 10,582 training images, provided by [16], using only image-level lower-count supervision. The performance is evaluated on 1,449 object segmentation images from the validation set of PASCAL VOC 2012.

Evaluation Criteria: Following previous works [7], [24], we evaluate generic object counting using the root-mean-square error (RMSE) metric and its three variants; namely, RMSE non-zero (RMSE-nz), relative RMSE (relRMSE) and relative RMSE non-zero (relRMSE-nz). The $RMSE_c$ and $relRMSE_c$ errors for a category c are computed as $\sqrt{\frac{1}{T}\sum_{i=1}^{T}(t_{ic} - \hat{t_{ic}})^2}$ and $\sqrt{\frac{1}{T}\sum_{i=1}^{T}\frac{(t_{ic} - \hat{t_{ic}})^2}{t_{ic}+1}}$, respectively. Here, T is the total number of images in the test set and $\hat{t}_{ic}$, t_{ic} are the predicted count (rounded to the nearest integer) and the ground-truth count for category c in an image i. The errors are then averaged across all categories to obtain the mRMSE and m-relRMSE on a dataset. The above metrics are also evaluated for ground-truth instances with non-zero counts as mRMSE-nz and m-relRMSE-nz. To evaluate the category-independent total count $\hat{t}_{tot}$, we only use RMSE and relRMSE since each image has at least one instance and the RMSE-nz metric is equal to the RMSE metric. For all error metrics mentioned above, *smaller* numbers indicate *better* performance. We refer to [7] for more details. For instance segmentation, the performance is evaluated using Average Best Overlap (ABO) [40] and mAP^r, as in [54]. The mAP^r is computed with intersection over union (IoU) thresholds of 0.25, 0.5 and 0.75. Throughout our experiments (Table 2 - 12) , we report the top-two results in red and blue font, respectively.

Supervision Levels: The level of supervision is indicated as SV in Tables 2, 6, 7 and 8. BB indicates bounding box supervision and PL indicates point-level supervision for each object instance. Image-level supervised methods using only within subitizing or lower-count range counts are denoted

Approach	SV	mRMSE	mRMSE-nz	m-rel RMSE	m-rel RMSE-nz
CAM+MSE	IC	**0.45**	**1.52**	**0.29**	**0.64**
Peak+MSE	IC	0.64	2.51	0.30	1.06
Proposed (LC)	LC	**0.29**	**1.14**	**0.17**	**0.61**

TABLE 2

Counting performance on the Pascal VOC 2007 count-test set using our approach and two baselines. Both baselines are obtained by training the network using the MSE loss function.

	$\mathcal{L}_{class}+\mathcal{L}_{MSE}$	$\mathcal{L}_{class}+\mathcal{L}_{spatial}+\mathcal{L}_{MSE}$	$\mathcal{L}$ $\lambda=0.1$	$\mathcal{L}$ $\lambda=0.01$	$\mathcal{L}$ $\lambda=0.05$	$\mathcal{L}$ $\lambda=0.5$	$\mathcal{L}$ $\lambda=1$
mRMSE	0.36	0.33	**0.29**	0.31	**0.30**	0.32	0.36
mRMSE-nz	1.52	1.32	**1.14**	1.27	**1.16**	1.23	1.40

TABLE 3

Left: Progressive integration of different terms in the loss function and their impact on the final counting performance of our LC framework on the PASCAL VOC count-test set. Right: Influence of the weight (λ) of ranking loss.

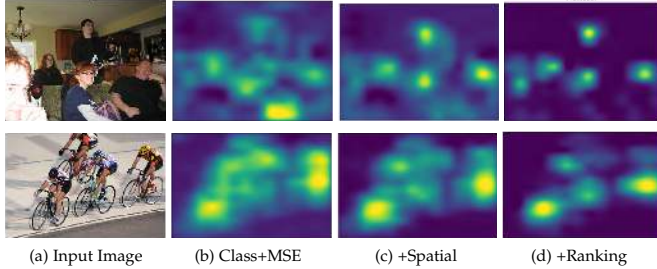(a) Input Image   (b) Class+MSE   (c) +Spatial   (d) +Ranking

Fig. 6. Progressive improvement in density map quality with the incremental introduction of spatial and ranking loss terms in our LC framework. In both cases (top row: person and bottom row: bicycle), our overall loss function integrating all three terms provides the best density maps. The category-specific object count is accurately predicted (top row: 5 persons and bottom row: 4 bicycles) by accumulating the respective density map.

as LC, while the methods using both within and beyond subitizing range counts are indicated as IC. The method using reduced lower-count supervison is indicated as RLC.

7.1 Ablation Study

7.1.1 Counting with Lower-count Supervision (LC)

Importance of Dual-branch Architecture: We perform an ablation study on the PASCAL VOC 2007 count-test. First, the impact of our dual-branch architecture is analyzed by comparing it with two baselines: class-activation [53] based regression (CAM+MSE) and peak-based regression (Peak+MSE), using the local-maximum boosting approach of [54]. Both baselines are obtained by end-to-end training of the network, employing the same backbone, and using the MSE loss function to directly predict the category-specific object count. Table 2 shows the comparison. Our approach largely outperforms both baselines, highlighting the importance of having a dual-branch architecture. Next, we evaluate the contribution of each term in our loss function towards the final counting performance.

Importance of Different Loss terms: Fig. 6 shows the systematic improvement in the quality of density maps (top row: person and bottom row: bicycle) with the incremental addition of (c) spatial $\mathcal{L}_{spatial}$ and (d) ranking ($\mathcal{L}_{rank}$) loss terms to the (b) MSE ($\mathcal{L}_{MSE}$) loss term. Incorporating the spatial loss term improves the spatial distribution of objects in both density maps. The density maps are further improved by the incorporation of the ranking term, which penalizes the under-estimation of counts beyond the lower-count range (top row) in the loss function. Moreover, it also helps to reduce the false positives within the lower-count range (bottom row). Table 3 shows the systematic improvement, in terms of mRMSE and mRMSE-nz, when integrating different terms in our loss function. The best results are obtained when integrating all three terms (classification, spatial and count) in our loss function. We also evaluate
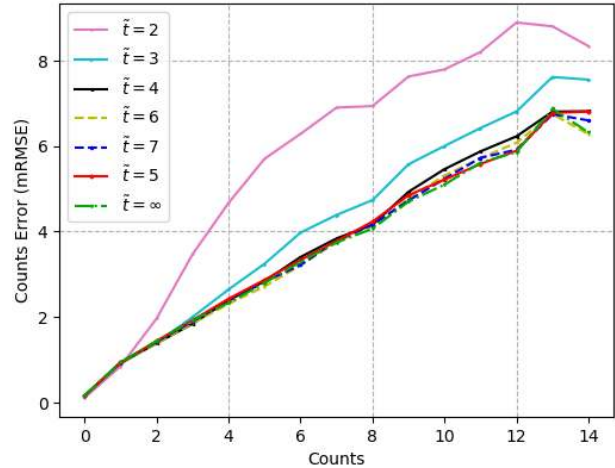

Fig. 7. Impact of lower-count range on the counting performance, evaluated on the COCO count-test set. The lower-count range is defined from the count 1 till $\tilde{t} - 1$. We vary the value of $\tilde{t}$ and plot mRMSE w.r.t. the ground-truth counts. As can be observed, very small values such as $\tilde{t} = 2$ (pink) and $\tilde{t} = 3$ (cyan) lead to considerable reduction in accuracy, compared to $\tilde{t} = 5$ (red). At larger counts between 8 to 12 (x-axis), $\tilde{t} = 4$ (black) leads to a slight reduction in performance compared to $\tilde{t} = 5$. Nearly the same performance is obtained for $\tilde{t} = 5$, $\tilde{t} = 6$ and $\tilde{t} = 7$, indicating that the optimum balance between the annotation cost and counting performance is achieved at $\tilde{t} = 5$. We also trained the proposed model using image-level count supervision (IC), by removing the ranking loss term from the proposed loss function, shown as $\tilde{t} = \infty$ (green). Although $\tilde{t} = \infty$ gives the best performance, it requires higher supervision (more costly) to annotate the counts.

the influence of λ, which controls the relative weight of the ranking loss. We observe $\lambda = 0.1$ provides the best results and fix it for all datasets.

Effect of Lower-count Range: In our LC and RLC frameworks, object count information beyond the lower-count range is not used for all object categories, i.e, for object categories with ground truth counts ($t_c \geq \tilde{t}$). Psychological studies show that humans require less time to count a smaller number of instances. This property can be used to largely reduce the count annotation cost by instructing the annotators not to count beyond the lower-count range, i.e. not to count more than $\tilde{t}$ instances of the same object category. In Fig. 7, we analyze the influence of $\tilde{t}$ on the counting performance of our LC framework, on COCO count-test set, by varying $\tilde{t}$ from 2 to 7 and plotting the count error (RMSE) at various ground-truth counts. The figure shows that $\tilde{t} = 5$, $\tilde{t} = 6$, and $\tilde{t} = 7$ give nearly the same counting performance, and hence an optimum balance between the annotation cost and performance can be obtained at $\tilde{t} = 5$.

Evaluation of Density Map: We employ the standard grid average mean absolute error (GAME) evaluation metric [15] used in crowd counting to evaluate spatial distribution

Method	Split	mRMSE
Proposed (RLC)	20/60	0.41
	40/40	0.37
	60/20	**0.36**
LCFCN [24]	80/0	0.39
Proposed (LC)	80/0	**0.34**

TABLE 4
Evaluation of RLC framework with different known/unknown count splits, on the COCO count-test set.

Design	Size of hidden layer	Activation	mRMSE	mRMSE-nz
2 layers	$P/2$	Relu	0.36	1.97
2 layers	$P/2$	Leaky Relu	0.36	**1.96**
2 layers	$P/2$	Softmax	0.36	**1.94**
2 layers	P	Softmax	0.37	1.98
3 layers	P	Softmax	0.38	2.07

TABLE 5
Study on the structure of the weight modulation layer on the COCO count-test set by varying the number of fully connected layers, size of hidden layers, and activation functions. We observe that the best performance (mRMSE-nz) is obtained by a 2-layered network with a bottleneck low-dimensional embedding (to $\frac{P}{2}$) and a softmax non-linear activation, and fix it for all datasets.

Approach	SV	mRMSE	mRMSE-nz	m-rel RMSE	m-rel RMSE-nz
Aso-sub-ft-3×3 [7]	BB	0.38	2.08	0.24	0.87
Seq-sub-ft-3×3 [7]	BB	**0.35**	**1.96**	**0.18**	**0.82**
ens [7]	BB	0.36	1.98	**0.18**	**0.81**
Fast-RCNN [7]	BB	0.49	2.78	0.20	1.13
LCFCN [24]	PL	0.38	2.20	0.19	0.99
glancing [7]	IC	0.42	2.25	0.23	0.91
Proposed (LC)	LC	**0.34**	**1.89**	**0.18**	0.84
Proposed (RLC)	RLC	0.36±0.01	**1.96±0.01**	0.19±0.00	0.84±0.01

TABLE 6
State-of-the-art counting performance comparison on the COCO count-test set. Despite using reduced supervision, our LC approach provides superior results compared to existing methods on three metrics. Compared to the image-level count (IC) supervised approach [7], our LC method achieves an absolute gain of 8% in terms of mRMSE. For the RLC framework, we follow a 60/20 known/unknown count annotation split, and repeat experiments three times, randomly interchanging object categories among these sets, and report the mean and standard deviation. Note that despite using only the category labels for 20 categories, our RLC framework performs favorably compared to its LC counterpart.

consistency in the density map. In GAME (n), an image is divided into 4^n non-overlapping grid cells. The mean absolute error (MAE) between the predicted and the ground-truth local counts are reported for $n = 0, 1, 2$ and 3, as in [15]. We compare our approach with the state-of-the-art PL supervised counting approach (LCFCN) [24] on the 20 categories of the PASCAL VOC 2007 count-test set. Furthermore, we also compare with a popular crowd counting approach (CSRnet) [27] on the person category of the PASCAL VOC 2007 by retraining it on this dataset. For the person category, the PL supervised LCFCN and CSRnet approaches achieve scores of 2.80 and 2.44 in GAME (3). The proposed method outperforms LCFCN and CSRnet in GAME (3) with a score of 1.83, demonstrating the capabilities of our approach in estimating the precise spatial distribution of object counts. Moreover, our method outperforms LCFCN for all 20 categories. Additional ablation results are available in the supplementary material.

7.1.2 Counting with Reduced Lower-count Supervision (RLC)

To evaluate the significance of the additional components introduced in our RLC framework, we report an ablation study on the COCO [28] dataset. First, we show how the number of count-annotated categories ($|\mathcal{A}|$) affects the overall category-specific accuracy. We start with only using count annotations of 20 categories, and then gradually increase the number of count-annotated categories to 40 and 60. Note that the compared methods from our LC and LCFCN [24] use counts for all 80 categories. The comparison is shown in Table 4. With only half the number of count-annotated categories (40/40 split), our RLC approach performs better than LCFCN [24], which requires point-level supervision. We can see that our method obtains consistent performance gain with the increase of supervision. Furthermore, when using count annotations for 60 categories, we obtain an mRMSE of 0.36, which is comparable with the 0.34 of our LC framework, which uses lower-count supervision for all categories.

In Table 5, we perform an experiment to identify the optimum structure for the weight modulation layer, by varying the number of fully connected layers, size of the hidden layer and activation functions. We find that a 2-layer network with a bottleneck low-dimensional embedding (to $\frac{P}{2}$), and using softmax activation works best and therefore follow this structure in all our experiments.

7.2 State-of-the-art Comparison

Here, we compare the counting performance of our LC and RLC frameworks with state-of-the-art approaches. At first, we evaluate category-specific object counting on COCO, Visual Genome and PASCAL VOC 2007 datasets and the results are shown in Tables 6, 7 and 8, respectively. Secondly, we evaluate category-independent total object counting performance on COCO, Visual Genome datasets and the results are compared with existing methods in Tables 9, 10, respectively. Finally, we evaluate the generalization ability of category-independent counts predicted by our RLC framework, to unsupervised object categories, and the results are compared with methods using count-supervision in Table 11. Additional results and failure cases are included in the supplementary material.

7.2.1 Category-specific counting

Table 6 compares the category-specific object counting performances on the COCO dataset, which has 80 object categories. To train our RLC framework, we use a 60/20 split, where the 80 categories are randomly split into 60 and 20 categories. Here, 60 categories have both class labels and count annotations, and 20 categories only have class labels. To avoid any bias due to a particular random split, we repeat experiments three times with different random splits and report the mean and standard deviation in the table. For all other methods, both count (lower-count for LC) information and class labels are used for all 80 object categories. Among all existing methods, the two BB supervised approaches (Seq-sub-ft-3x3 and ens) yield mRMSE scores of 0.35 and 0.36, respectively. The PL supervised LCFCN approach [24] achieves an mRMSE score of 0.38. The IC supervised glancing approach obtains an mRMSE score of 0.42. Our LC framework outperforms the glancing approach with an absolute gain of 8% in mRMSE. Furthermore, our LC framework also provides consistent improvements over the glancing approach, in the other three error metrics and

Metrics	Glancing [7]	LCFCN [24]	Proposed (LC)	Proposed (RLC)	
				split 304/305	split 479/130
SV	IC	PL	LC	RLC	RLC
mRMSE	**0.15**	0.16	**0.15**	0.15	**0.15**
mRMSE-nz	1.57	1.62	**1.51**	1.54	**1.53**
m-relRMSE	**0.09**	**0.09**	**0.09**	0.10	**0.09**

TABLE 7

Results on Visual Genome dataset for all classes. Our approaches perform favorably against methods using higher levels of supervision. Our RLC framework is evaluated on two different known/unknown count splits and, for both splits, performs on par with its LC counterpart which uses lower-count annotations for all object categories.

Approach	SV	mRMSE	mRMSE -nz	m-rel RMSE	m-rel RMSE-nz
Aso-sub-ft-3×3 [7]	BB	0.43	1.65	0.22	0.68
Seq-sub-ft-3×3 [7]	BB	0.42	1.65	0.21	0.68
ens [7]	BB	0.42	1.68	0.20	0.65
Fast-RCNN [7]	BB	0.50	1.92	0.26	0.85
LCFCN [24]	PL	**0.31**	**1.20**	**0.17**	**0.61**
LC-PSPNet [24]	PL	0.35	1.32	0.20	0.70
glancing [7]	IC	0.50	1.83	0.27	0.73
Proposed (LC)	LC	**0.29**	**1.14**	**0.17**	**0.61**

TABLE 8

State-of-the-art counting performance comparison on the Pascal VOC 2007 count-test. Our LC supervised approach outperforms existing methods.

	glancing [7]	LCFCN [24]	Proposed (LC)	Proposed (RLC)
Categ.-indep.	✗	✗	✗	✓
mRMSE	7.01± 0.00	6.13± 0.00	**4.4±0.00**	**4.28±0.03**
m-relRMSE	2.05± 0.00	1.65± 0.00	**1.14±0.00**	**1.06± 0.01**

TABLE 9

Results of total count estimation on COCO dataset. Note that our RLC framework does not use the count annotations of 20 classes during training and performs favorably compared to methods that use count annotations for all classes, demonstrating its the generalizability .

Fig. 8. Object counting examples on the COCO and Visual Genome datasets. The ground-truth is shown in green, while the predictions by glancing [7], LCFCN [24], our LC framework, and our RLC framework, are shown sequentially inside the parentheses. The examples show that our LC and RLC frameworks accurately predict counts for diverse categories (animals to food items), and even beyond the lower-count range. Although the count-annotation of object categories indicated with blue are not used for training the RLC framework, their counts are predicted accurately. Finally, the category-independent total count (TC) predicted by our RLC framework is shown separately, inside parentheses. Best viewed in zoom.

is only below the two BB supervised methods (Seq-sub-ft3x3 and ens) in m-relRMSE-nz. Further, our RLC approach, despite using only category-level binary labels for 20 categories, performs favorably compared to its LC counterpart.

For the large-scale Visual Genome dataset, our proposed RLC method is evaluated under two different splits. The first split follows [3]: all 609 classes are split into 479 and 130 classes, where 479 classes have both image labels and counts annotations, and 130 classes only have image labels. For the second split, we experiment on a more challenging case: the 609 classes are randomly split into half and half (304 classes and 305 classes), where the first half has both image labels and count annotations, and the second half only has image labels. The comparisons are shown in Table 7. It shows that both our proposed LC and RLC methods perform on par with existing methods (glancing and LCFCN). Note that in the case of our RLC approach, count information for some classes is not used during training. Fig. 8 shows object counting examples on the COCO and Visual Genome datasets, using glancing [7], LCFCN [24] and the proposed LC, RLC frameworks. Both our approaches perform accurate counting on various categories (animals to food items) under challenging situations.

On the PASCAL VOC 2007 dataset, we only evaluate our LC method since RLC is designed for large-scale datasets and PASCAL VOC only has 20 classes. Table 8 shows that the glancing approach of [7] using image-level supervision both within and beyond the lower-count range (IC) achieves an mRMSE score of 0.50. Our LC supervised approach outperforms the glancing method with an absolute gain of 21% in mRMSE. Furthermore, our approach achieves favorable performance on all error metrics, compared to the state-of-the-art point-level and bounding box based supervised methods.

7.2.2 Category-independent Total Object Counting

For the COCO dataset, we report the performance of category-independent total count prediction in Table 9. Our RLC method has a dedicated category-independent density

estimation sub-branch that can directly provide the total count predictions. For methods that do not have such a capability, we compute the total count by summing up the category-specific counts. We note that, in the presence of highly complex scenes with a diverse range of object categories, difficult object instances are missed during category-specific counting. When such errors occur for several classes, they get accumulated in the total object count. As a result, the total counts have a larger variation from the ground-turth. In such cases, a holistic density map helps in estimating the overall object count. This is evident from Table 9, where the proposed category-independent density estimation sub-branch of the RLC framework predicts the most accurate total counts on the COCO dataset.

For the Visual Genome (VG) dataset, the results for total counting are shown in Table 10. The VG dataset has high diversity in terms of object classes, which makes it a very challenging dataset. Furthermore, each image has a large number of objects, i.e., 35 object categories/image on average, which makes the counting task even more difficult. Remarkably, for both splits, our proposed category-independent density sub-branch achieves the best performance despite the fact that the counts for the 305 or 130 classes (depending on split) are not used during training. The favorable performance of our RLC framework compared to its LC counterpart demonstrates the generalization ability of the RLC framework on large-scale datasets.

Method	glancing [7]	LCFCN [24]	Proposed (LC)	Proposed (RLC)	
				split 304/305	split 479/130
Categ.-indep.	✗	✗	✗	✓	✓
mRMSE	9.88	13.20	9.49	**7.35**	**6.93**
m-relRMSE	2.40	3.12	2.34	**1.97**	**1.94**

TABLE 10

Results of total count estimation on the Visual Genome dataset. On this large-scale dataset, our LC and RLC frameworks demonstrate superior performance compared to existing approaches. Our RLC framework is evaluated on two challenging known/unknown count splits, and for both splits, the total counts predicted by our RLC framework, which has a dedicated category-independent density sub-branch, are more accurate, compared to the total count estimated using the LC framework.

	glancing [7]	LCFCN [24]	Proposed (LC)	Proposed (RLC)
Categ.-indep.	✗	✗	✗	✓
mRMSE	2.66	1.59	**1.48**	**1.42**
m-relRMSE	1.60	0.87	**0.76**	**0.85**

TABLE 11

Total count estimation performance of RLC framework, on 10 unsupervised classes of the COCO split 60-10-10. The images containing these 10 classes are not included in our RLC framework training. Both [7], [24] and our LC setting are trained using the complete count-train set with count annotations for all 80 categories. Our RLC approach with less supervision performs comparably to these methods.

7.2.3 Generalization to Unsupervised Object Counting

Here, we consider a learning scenario that is motivated by the zero-shot setting [3], where 80 COCO classes are split into subsets of 60-10-10. The first 60 classes have both image-level labels and object count annotations. The second 10 classes only have image-level labels but no count annotations. The last 10 classes do not have any annotations and images containing objects from any of those 10 categories are not used during the end-to-end training of our RLC framework. We train our RLC method using the first two sets (60+10 classes) with labels, and we test the category-independent density branch on images that only have objects from the last 10 classes. The total count performance on the last 10 classes is shown in Table 11. Note that all other methods, including our LC framework, are trained using count/lower-count annotations for all 80 object categories. We can see that, for the unannotated categories, our RLC framework performs favorably compared to LCFCN [24], which requires point-level annotations for training. Our RLC method achieves 1.42 and 0.85, in terms of mRMSE and m-relRMSE, while LCFCN obtains 1.59 and 0.87. Our RLC setting even performs comparably to our LC setting in mRMSE. Since the last 10 categories are not used during our training, the favorable performance demonstrates that the proposed RLC method can generalize to predict the total count of un annotated categories.

7.3 Application to Weakly Supervised Instance Segmentation

The category-specific density maps generated by our LC framework can also be utilized for instance segmentation. Note that the local summation of an ideal density map over a ground-truth segmentation mask is nearly one. We use this property to improve a recent image-level supervised instance segmentation method (PRM [54]). PRM employs a scoring metric that combines instance-level cues from peak response maps R, class-aware information from object category maps and spatial continuity priors from off-the-shelf object proposals [35], [39]. The peak response maps

Method	$mAP_{0.25}^r$	$mAP_{0.5}^r$	$mAP_{0.75}^r$	ABO
MELM+MCG [45]	36.9	22.9	8.4	32.9
CAM+MCG [53]	20.4	7.8	2.5	23.0
SPN+MCG [56]	26.4	12.7	4.4	27.1
PRM [54]	**44.3**	**26.8**	**9.0**	**37.6**
Proposed	**48.5**	**30.2**	**14.4**	**44.3**

TABLE 12

Image-level supervised instance segmentation results on the PASCAL VOC 2012 val. set in terms of mean average precision (mAP%) and Average Best Overlap(ABO). Our approach improves PRM [54] with a relative gain of 17.8% in terms of ABO.

are generated from local maxima (peaks of $\tilde{\mathbf{M}}^c$) through a peak backpropagation process [54]. The scoring metric is then used to rank object proposals corresponding to each peak for instance mask prediction. We improve the scoring metric by introducing an additional term d_p. The term d_p penalizes an object proposal P_r if the predicted count in those regions of the density map $\mathbf{D}^c$ is deviated from one, as $d_p = |1 - \|\mathbf{D}^c \cdot P_r\|_{sum}|$. Here, $|\ |$ is the absolute value operator. For each peak, the new scoring metric $Score$ selects the highest-scoring object proposal P_r:

$$Score = \alpha \cdot R * P_r + R * \hat{P}_r - \beta \cdot Q * P_r - \gamma \cdot d_p. \quad (17)$$

Here, the background mask Q is derived from the object category map and $\hat{P}_r$ is the contour mask of the proposal P_r, derived using a morphological gradient [54]. Parameters α, β and γ are empirically set as in [54].

Following PRM, we evaluate the instance segmentation performance on the PASCAL VOC 2012 dataset and the results are shown in Table 12. For fair comparison, we utilize the same proposals (MCG) as used in [54]. Specifically, the combinatorial grouping framework of [39] is used in conjunction with the region hierarchies of [35], which is referred to as MCG in [54]. Note that our approach is generic and can be used with any object proposal method. In addition to PRM, the image-level supervised object detection methods MELM [45], CAM [53] and SPN [56] used with MCG and reported by [54] are also included in Table 12.

The proposed method largely outperforms all the baseline approaches and [54], in all four evaluation metrics. Even though our approach slightly increases the level of supervision (lower-count information), it improves PRM with a relative gain of 17.8% in terms of average best overlap (ABO). Compared to PRM, the gain obtained at a lower IoU threshold (0.25) highlights the improved location prediction capabilities of the proposed method. Furthermore, the gain obtained at a higher IoU threshold (0.75) indicates the effectiveness of the proposed scoring function in assigning a higher score to the object proposal that has the highest overlap with the ground-truth object, as indicated by the improved ABO performance. Fig. 9 shows a qualitative instance segmentation comparison between our approach and PRM.

8 CONCLUSION

We introduced a partially supervised setting for generic object counting in natural scenes, and proposed two novel frameworks to enable object counting under this challenging setting. Our frameworks were built on a novel dual-branch architecture having an image classification branch,

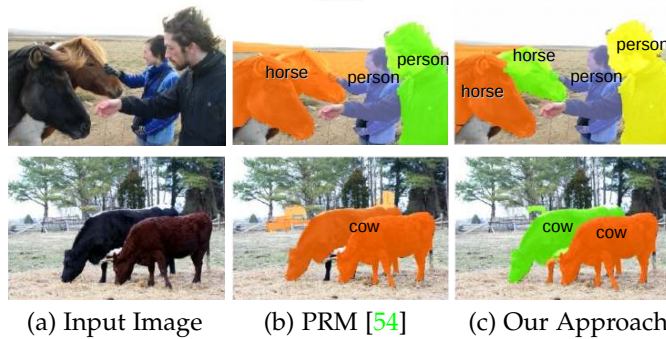(a) Input Image   (b) PRM [54]   (c) Our Approach

Fig. 9. Instance segmentation examples of PRM [54] and our approach. Our approach accurately delineates multiple spatially adjacent object instances of the horse and cow categories.

and a density branch that estimates the object count in an image. Our first framework (the LC framework) requires only lower-count supervision, and hence reduces the annotation cost due to large numbers of instances in an image. As an extension, the second framework (the RLC framework) uses lower-count supervision only for a subset of object categories, and hence further reduces the annotation cost due to large numbers of object categories appearing in natural scene datasets. To the best of our knowledge, we are the first to propose image-level lower-count supervised training for a density map, and demonstrate the applicability of density maps in image-level supervised instance segmentation. Thorough experiments were performed on three challenging datasets (COCO, Visual Genome and PASCAL VOC) to evaluate the category-specific and category-independent object counting performance of the proposed method. The evaluations demonstrated that the proposed frameworks perform on par with approaches using higher levels of supervision.

REFERENCES

[1] Jiwoon Ahn, Sunghyun Cho, and Suha Kwak. Weakly supervised learning of instance segmentation with inter-pixel relations. In *CVPR*, 2019. 10

[2] Md Amirul Islam, Mahmoud Kalash, and Neil D. B. Bruce. Revisiting salient object detection: Simultaneous detection, ranking, and subitizing of multiple salient objects. In *CVPR*, June 2018. 3

[3] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-shot object detection. In *ECCV*, 2018. 10, 13, 14

[4] O Barinova, V Lempitsky, and P Kohli. On the detection of multiple object instances using hough transforms. In *CVPR*, 2010. 3

[5] Sarah T Boysen and E John Capaldi. *The development of numerical competence: Animal and human models*. Psychology Press, 2014. 1

[6] Xinkun Cao, Zhipeng Wang, Yanyun Zhao, and Fei Su. Scale aggregation network for accurate and efficient crowd counting. In *ECCV*, 2018. 3

[7] Prithvijit Chattopadhyay, Ramakrishna Vedantam, Ramprasaath R. Selvaraju, Dhruv Batra, and Devi Parikh. Counting everyday objects in everyday scenes. In *CVPR*, 2017. 1, 2, 3, 10, 12, 13, 14

[8] Zhi-Qi Cheng, Jun-Xiu Li, Qi Dai, Xiao Wu, and Alexander Hauptmann. Learning spatial awareness to improve crowd counting. In *ICCV*, 2019. 3

[9] Chhavi. k largest(or smallest) elements in an array-added min heap method, 2018. 6

[10] Hisham Cholakkal, Guolei Sun, Fahad Shahbaz Khan, and Ling Shao. Object counting and instance segmentation with image-level supervision. In *CVPR*, 2019. 4

[11] Douglas H Clements. Subitizing: What is it? why teach it? *Teaching children mathematics*, 5, 1999. 1

[12] C Desai, D Ramanan, and C Fowlkes. Discriminative models for multi-class object layout. In *ICCV*, 2009. 3

[13] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1), 2015. 10

[14] Mingfei Gao, Ang Li, Ruichi Yu, Vlad I. Morariu, and Larry S. Davis. C-wsl: Count-guided weakly supervised localization. In *ECCV*, 2018. 3

[15] R Guerrero, B Torre, R Lopez, S Maldonado, and D Onoro. Extremely overlapping vehicle counting. In *IbPRIA*, 2015. 11, 12

[16] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. 2011. 10

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[18] Shengfeng He, Jianbo Jiao, Xiaodan Zhang, Guoqiang Han, and Rynson WH Lau. Delving into salient object subitizing and detection. In *ICCV*, pages 1059–1067, 2017. 3

[19] Judy Hoffman, Sergio Guadarrama, Eric S Tzeng, Ronghang Hu, Jeff Donahue, Ross Girshick, Trevor Darrell, and Kate Saenko. Lsda: Large scale detection through adaptation. In *NIPS*, pages 3536–3544, 2014. 3

[20] Ronghang Hu, Piotr Dollár, Kaiming He, Trevor Darrell, and Ross Girshick. Learning to segment every thing. In *CVPR*, 2018. 3

[21] Brenda RJ Jansen, Abe D Hofman, Marthe Straatemeier, Bianca MCW van Bers, Maartje EJ Raijmakers, and Han LJ van der Maas. The role of pattern recognition in children's exact enumeration of small numbers. *British Journal of Developmental Psychology*, 32(2), 2014. 1

[22] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016. 1, 3, 10

[23] Maksim Lapin, Matthias Hein, and Bernt Schiele. Analysis and optimization of loss functions for multiclass, top-k, and multilabel classification. *TPAMI*, 40(7), 2018. 5, 6

[24] Issam H. Laradji, Negar Rostamzadeh, Pedro O. Pinheiro, David Vazquez, and Mark Schmidt. Where are the blobs: Counting by localization with point supervision. In *ECCV*, 2018. 1, 2, 3, 10, 12, 13, 14

[25] Issam H Laradji, David Vazquez, and Mark Schmidt. Where are the masks: Instance segmentation with image-level supervision. In *BMVC*, 2019. 10

[26] Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. In *NIPS*. 2010. 5, 6

[27] Yuhong Li, Xiaofan Zhang, and Deming Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *CVPR*, 2018. 3, 12

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*. Springer, 2014. 1, 3, 10, 12

[29] Ning Liu, Yongchao Long, Changqing Zou, Qun Niu, Li Pan, and Hefeng Wu. Adcrowdnet: An attention-injective deformable convolutional network for crowd understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3

[30] Weizhe Liu, Mathieu Salzmann, and Pascal Fua. Context-aware crowd counting. In *CVPR*, 2019. 3

[31] Xialei Liu, Joost van de Weijer, and Andrew D Bagdanov. Leveraging unlabeled data for crowd counting by learning to rank. In *CVPR*, 2018. 3

[32] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 15. 5

[33] E. Lu, W. Xie, and A. Zisserman. Class-agnostic counting. In *ACCV*, 2018. 3

[34] George Mandler and Billie J Shebo. Subitizing: an analysis of its component processes. *Journal of Experimental Psychology: General*, 111(1), 1982. 1

[35] Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Pablo Arbeláez, and Luc Van Gool. Convolutional oriented boundaries. In *ECCV*. Springer, 2016. 14

[36] Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In *CVPR*, 2015. 5, 6

[37] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 7

[38] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS Workshop*, 2017. 10

[39] Jordi Pont-Tuset, Pablo Arbelaez, Jonathan T Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. *TPAMI*, 39(1), 2017. 14

[40] Jordi Pont-Tuset and Luc Van Gool. Boosting object proposals: From pascal to coco. In *ICCV*, 2015. 10

[41] Shafin Rahman, Salman Khan, and Fatih Porikli. Zero-shot object detection: Learning to simultaneously recognize and localize novel concepts. *ACCV*, 2018. 3

[42] Girshick Ross. Fast r-cnn. In *ICCV*, 2015. 1, 3

[43] Miaojing Shi, Zhaohui Yang, Chao Xu, and Qijun Chen. Revisiting perspective information for efficient crowd counting. In *CVPR*, 2019. 3

[44] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NIPS*, 2017. 3

[45] Fang Wan, Pengxu Wei, Jianbin Jiao, Zhenjun Han, and Qixiang Ye. Min-entropy latent model for weakly supervised object detection. In *CVPR*, 2018. 14

[46] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *CVPR*, 2014. 7

[47] Qi Wang, Junyu Gao, Wei Lin, and Yuan Yuan. Learning from synthetic data for crowd counting in the wild. In *CVPR*, 2019. 3

[48] Kesheng Wu, Ekow Otoo, and Arie Shoshani. Optimizing connected component labeling algorithms. In *Medical Imaging 2005: Image Processing*, volume 5747. International Society for Optics and Photonics, 2005. 3

[49] Haipeng Xiong, Hao Lu, Chengxin Liu, Liang Liu, Zhiguo Cao, and Chunhua Shen. From open set to closed set: Counting objects by spatial divide-and-conquer. In *ICCV*, 2019. 3

[50] Jia Xu, Alexander G Schwing, and Raquel Urtasun. Learning to segment under various forms of weak supervision. In *CVPR*, 2015. 3

[51] Jianming Zhang, Shugao Ma, Mehrnoosh Sameki, Stan Sclaroff, Margrit Betke, Zhe Lin, Xiaohui Shen, Brian Price, and Radomir Mech. Salient object subitizing. In *CVPR*, 2015. 3

[52] Fang Zhao, Jian Zhao, Shuicheng Yan, and Jiashi Feng. Dynamic conditional networks for few-shot learning. In *ECCV*, September 2018. 3

[53] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016. 5, 6, 11, 14

[54] Yanzhao Zhou, Yi Zhu, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Weakly supervised instance segmentation using class peak response. In *CVPR*, 2018. 4, 6, 7, 10, 11, 14, 15

[55] Yi Zhu, Yanzhao Zhou, Huijuan Xu, Qixiang Ye, David Doermann, and Jianbin Jiao. Learning instance activation maps for weakly supervised instance segmentation. In *CVPR*, 2019. 10

[56] Yi Zhu, Yanzhao Zhou, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Soft proposal networks for weakly supervised object localization. In *ICCV*, 2017. 14