Divide and Count: Generic Object Counting by Image Divisions

Tobias Stahl, Silvia L. Pintea and Jan C. van Gemert

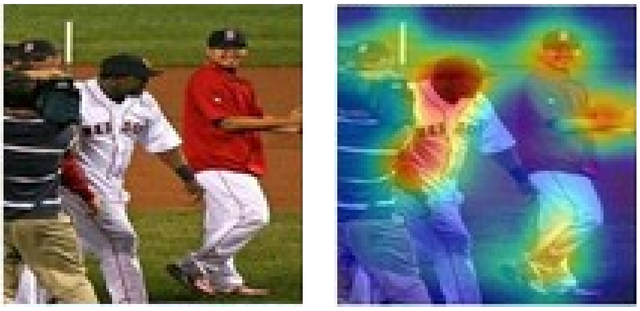
***Abstract*—We propose a general object counting method that does not use any prior category information. We learn from local image divisions to predict global image-level counts without using any form of local annotations. Our method separates the input image into a sets of image divisions — each fully covering the image. Each image division is composed of a set of region proposals or uniform grid cells. Our approach learns in an end-to-end deep learning architecture to predict global image-level counts from local image divisions. The method incorporates a counting layer which predicts object counts in the complete image, by enforcing consistency in counts when dealing with overlapping image regions. Our counting layer is based on the inclusion-exclusion principle from set theory. We analyze the individual building blocks of our proposed approach on Pascal-VOC2007 and evaluate our method on the MS-COCO large scale generic object dataset as well as on three class-specific counting datasets: UCSD pedestrian dataset, and CARPK and PUCPR+ car datasets.**

***Index Terms*—Generic-class object counting, inclusion-exclusion principle, regression, fully convolutional networks, counting with region proposals.**

I. Introduction

Counting objects in images is essential for environmental fauna monitoring, traffic control, and crowd management. Such application have inspired successful methods tied to specific objects such as cells [20], [47], [48], animals [1], [46], cars [30], [41], and people [24], [31], [32], [49], [50]. Instead of developing an object specific method, we propose an end-to-end deep learning method for generic object counting.

Generic object counting is a difficult problem. A recent in-depth analysis of visual question answering systems [19] concluded that "*CNN features contain little information relevant to counting*". The hardness of the problem perhaps motivates why state-of-the-art counting methods include extra annotations on top of the total object count. There are, for example, object counting methods that intelligently make use of ground-truth region annotations [6], [44]. Yet, manually annotating regions on a large dataset is quite time-consuming [11], [26] and other works aim to reduce the supervision effort to point annotations only [1], [14], [23], [30]. In this paper we reduce the supervision effort even further. We keep the supervision level in accordance with the task: our method only requires the total number of objects in an image to be annotated.



Count three persons. CNN activations [51].

Fig. 1. Grouping is key for object counting. Faces are unique and should not be grouped together. On the other hand, the face, hands, and feet of the person on the right belong to the same person, and should be grouped together. In this paper we group locally and integrate the local group counts to a global image-level count.

We argue that *grouping* is key for object counting. Consider Figure 1, where we show an image with three persons and the strongest CNN feature responses [51]. We make the following observations: a) In Figure 1, a face is unique for a person and multiple faces should be kept apart. Thus, multiple objects in the same image can have similar looking parts. To prevent under-counting they should not be grouped together. b) Consider the face, the hands, and the feet of the person on the right; these parts should be grouped to a single person. Thus, in the extent of a single object there can be multiple distinctive part. To prevent over-counting these parts should be grouped together. c) If the persons move, the number of persons will stay the same. Thus, object counting does not care for object location and local object counts can be integrated over the full image. We draw inspiration from the local groups from object detection to a) keep single objects disjoint from other objects; b) group single objects together; c) to ignore the relative object location, we take note of the global image grouping from image classification.

In CNNs grouping is done by *pooling*. We borrow local pooling from object detection, where object proposals are commonly used [22], [35], [42], [52]. Object proposals strive towards finding bounding boxes that contain objects. Thus, they allow us to distinguish between different object instances, while grouping parts of a single object. From image classification we borrow global pooling, where integrating over all counts of each proposal bounding box sums to the global image-level count if the counts for overlapping regions are carefully incorporated. In this paper we learn to predict object counts by a new deep network layer that correctly pools overlapping regions, as illustrated in Figure 2.

To summarize the contributions of this work: (i) we propose a general object counting approach, where we make no assumption about the object class; (ii) we learn from local

Silvia L. Pintea and Jan C. van Gemert are with the Computer Vision Lab, Delft University of Technology, Delft, The Netherlands, e-mails: (S.L.Pintea@tudelft.nl, J.C.vanGemert@tudelft.nl).

Tobias Stahl was with the University of Amsterdam, Amsterdam, The Netherlands.

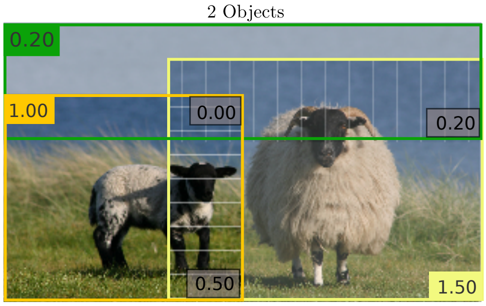


Fig. 2. Our method counts locally, and pools overlapping regions globally. This image shows a possible local grouping of the image using object proposals. Each bounding box has an associated predicted object count. The final object count for this image division is obtained by pooling all the counts of the bounding boxes, and subtracting or adding the counts of the overlapping bounding boxes. This pooled image-level count is optimized to match the global count annotation.

object proposals without the need of local image grouping annotations such as bounding boxes or point clicks; (iii) to this end, we incorporate a new global counting layer to enforce consistency between overlapping bounding box counts; (iv) and we empirically validate our model choices, as well as evaluate our proposed method on two generic object datasets: Pascal-VOC2007 and MS-COCO generic, and three class-specific datasets: UCSD, CARPK and PUCPR+.

II. Related Work

A. Counting Specific Objects

People counting. A popular application is counting people. Person-specific clues are helpful and avidly used for example with a person detector [7], [28], human head and shoulders detection [24], camera information [28], [37] or foreground/ background segmentation [21], [24]. Other approaches use the video motion information to aid with person segmentation [2], [4], [5], [32]. In [18] a dataset for people in extremely crowded scenarios is proposed. Yet other works integrate the most recent convolutional neural network architectures towards counting people [38], [45], [50]. Unlike these methods, we aim for a category-independent counting approach and we, therefore, do not make use of person-specific clues and evaluate our proposed approach on generic object datasets.

Cell counting. The task of counting cells has also received due attention because of its utility in biology. Given the large cardinality of cells, most approaches rely on dot annotations in solving this task [14], [45], [47], [48]. Dissimilar to these works, we do not focus only on object categories with large cardinality, such as cells, but rather on a range of commonly encountered objects.

Other categories. Other object categories have also been considered in the context of counting, due to their social or economical impact, such as vehicle counting [15], [27], [30], [41] or animal counting [1], [43], [46]. In this work, we do not focus on any individual object category, but rather, aim at proposing a category independent counting approach.

B. Counting Generic Category Objects

Few classes. Methods focusing on generic object counting, are less common than the ones focusing on a specific interesting category. Successful recent counting methods such [23], [30], [44] argue for the generality of the proposed models, however the evaluation is performed on two to three category-specific datasets such as pedestrians datasets, cells or vehicles datasets. Unlike these methods, we also aim to count generic objects, and we evaluate our proposed model on significantly more categories: 20 classes for the Pascal-VOC2007 dataset and 80 for the MS-COCO dataset.

Many classes. A few prior works have considered the task of generic object counting and evaluated it across a variety of object categories. In [34] a deep neural network approach with recurrent attention is proposed, using segmentation annotations and evaluates the method on a leaf datasets, a car dataset, as well as two categories in MS-COCO: person and zebra. Unlike this method, we use no local supervision and evaluate on a considerably larger variety of object categories. Similar to us, in [6], [39] generic object datasets are used towards evaluating the counting performance. Both methods, however, rely on on bounding box annotation. Dissimilar to them, in this work we propose to learn generic object counts only from image supervision and without any additional local supervision.

C. Annotation Effort

Bounding box annotations. In terms of annotations needed, a rather involved level of supervision is based on bounding-box annotations. Methods such as [2], [4]–[7], [39] have considered the use of object bounding boxes for improving the counting performance. In [44] the authors propose using annotated image region counts, rather than global image-level counts for a better performance. This validates that indeed, local information is useful. The downside of such approaches is that obtaining bounding box annotations for highly overlapping objects is difficult. Dissimilar to these works, here we choose to not employ any local supervision.

Dot annotations. Dot annotations involve marking a click location for each object present in an image. This annotation level is preferred in state-of-the-art counting methods as it allows for naturally emerged models based on density estimation [1], [30], [45]. Such an annotation level is suitable for object categories involving numerous instances that may occlude each other, as is the case for cells [14], [23], [45], [47], [48] or vehicles [15], [27], [30] or people [8], [14], [23], [30], [31], [49]. Unlike these methods, we wish to avoid the extra annotation effort involved in obtaining dot annotations, therefore, we propose a method based only on global image-level counts.

Image count annotations. The least expensive annotation level is global image-level counts. Methods replying on global image-level counts tend to be focused on categories involving a large number of objects, such as people counting [21], [28], [32], [37], [38], cell counting [20], [44] or animal counting [43], [46] and rely either on prior object detections [28], [43], [46] or on full image features [20], [38]. Unlike these methods, in our proposed approach we focus on counting generic objects. Moreover, rather than using global image features, we use local image regions to predict global image-level counts. Thus, we use only global image-level counts and still rely on local information.

III. Generic Counting by Image Division

In Figure 3 we show our proposed approach. Given an input image, we separate the image into a set of possible image divisions. We group these divisions in a hierarchy of image divisions, $\mathcal{D}$, with depth L where the granularity of the image regions increases with the hierarchy depth. Each such division, $\mathcal{D}_l$, fully covers the image. Each region, r_i, in each image division is input to a fully convolutional architecture where the feature maps are pooled over the current region to obtain region features, $\mathbf{x}_i$. The region features are fed into a per-image division fully connected layer, called IEP which combines the region features per image division. The output of the IEP is averaged over all mage divisions. An L_1 loss is computed with respect to full image-level counts for all object classes, C.

A. Underlying Architecture

We use as underlying architecture the region-based fully convolutional network described in [25]. We cut the network after the region pooling layer and here we add our IEP layer followed by the L_1 loss. We additionally change the architecture to be able to input a hierarchy of image divisions rather than a flat structure.

B. Locality from Image Divisions

We consider two possible approaches for adding locality information: (a) using a uniform grid over the image, and (b) using a hierarchy of image divisions from unsupervised object proposal regions.

(a) Uniform grid over the image. We define the image regions by separating the image into a grid of $k \times k$ equally-sized non-overlapping regions. These regions are pooled independently in the region pooling layer, following the approach described in [25]. For this case no hierarchy of image divisions is input, but rather the image is pooled separately per grid cell. In this case we only have 1 image division, therefore $L = 1$. We additionally also consider the case in which we have a hierarchical grid, where the number of cells in the grid increases with the hierarchy depth.

(b) Hierarchy of image divisions. The second approach towards adding locality, builds a hierarchy of image divisions, $\mathcal{D}$, as depicted in Figure 3. In this case our image regions represent object proposals. These proposals have different granularity levels therefore we utilize this by building a hierarchy of image divisions with increasing granularity levels, L. We start from the unsupervised object proposal method of [42] as it has an innate hierarchical structure, however, other object proposal methods can be used.

Our hierarchy of image divisions is built in a straightforward fashion based on 2 constraints: (i) each image division should fully cover the image, and (ii) the regions in one image division, $\mathcal{D}_l$, can overlap without being fully included in other regions. The first constraint allows us to use only global count annotations, while the second constraint allows us to separate the image divisions with respect to the level of granularity of the regions. We start the hierarchy construction by adding the full image as our depth-1 image division. We subsequently order all the object proposal regions in descending order of their size. We keep adding regions to the current image division, $\mathcal{D}_l$, until there is no region that can be added such that it is not fully included in another region of the current image division. We stop building the hierarchy when the available regions do not fully cover the image anymore. Each hierarchical division of an image can have variable depth, L, and variable number of regions per image division, $\mathcal{D}_l$.

Added value of locality. Using image divisions allows us to add local image information into the counting optimization while still relying on global image-level counts as annotations. The set of regions in each image division is disjoint from the other image divisions. Each image division can be optimized independently, as the object counts in an image division match the global object counts.

C. Inclusion-Exclusion Layer

Inclusion-Exclusion Principle. We propose a generic counting method that is not dependent on the region proposal method. In our architecture, we add a per-image division fully connected layer which combines the region features in one image division, $\mathcal{D}_l$. We consider multiple such image divisions. This layer is based on the inclusion-exclusion principle [40] — used when estimating counts over possibly overlapping sets. In our case we may have overlapping image regions in a given image division, as described in Section III-B.(b), and depicted in Figure 3. The underlying idea is that we can estimate the overall object counts in a given image division by adding the counts of all regions in that division, and then subtracting or adding the counts of the overlapping subregions.

Our IEP layer is based on the inclusion-exclusion principle from set theory [40]. Figure 4 depicts an example of a fixed image division, $\mathcal{D} = \{r_1, r_2, r_3\}$ composed of three overlapping image regions: r_1, r_2 and r_3. For this example we can formalize the inclusion-exclusion principle for counting objects as:

$$\begin{aligned}&\text{IEP}(\text{Count}(\cdot), \mathcal{D} = \{r_1, r_2, r_3\}) = \\ &\quad (\text{Count}(r_1) + \text{Count}(r_2) + \text{Count}(r_3)) - \\ &\quad (\text{Count}(r_1 \cap r_2) + \text{Count}(r_1 \cap r_3) + \text{Count}(r_2 \cap r_3)) + \\ &\quad \text{Count}(r_1 \cap r_2 \cap r_3). \end{aligned} \tag{1}$$

The function $\text{Count}(\cdot)$ denotes the object count over an image region demarcated by either a single bounding box or an intersection of bounding boxes. For an image division $\mathcal{D}_l = \{r_i\}_{i \in \{1,..N_l\}}$ composed of N_l region proposals r_i, the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2875353, IEEE Transactions on Image Processing

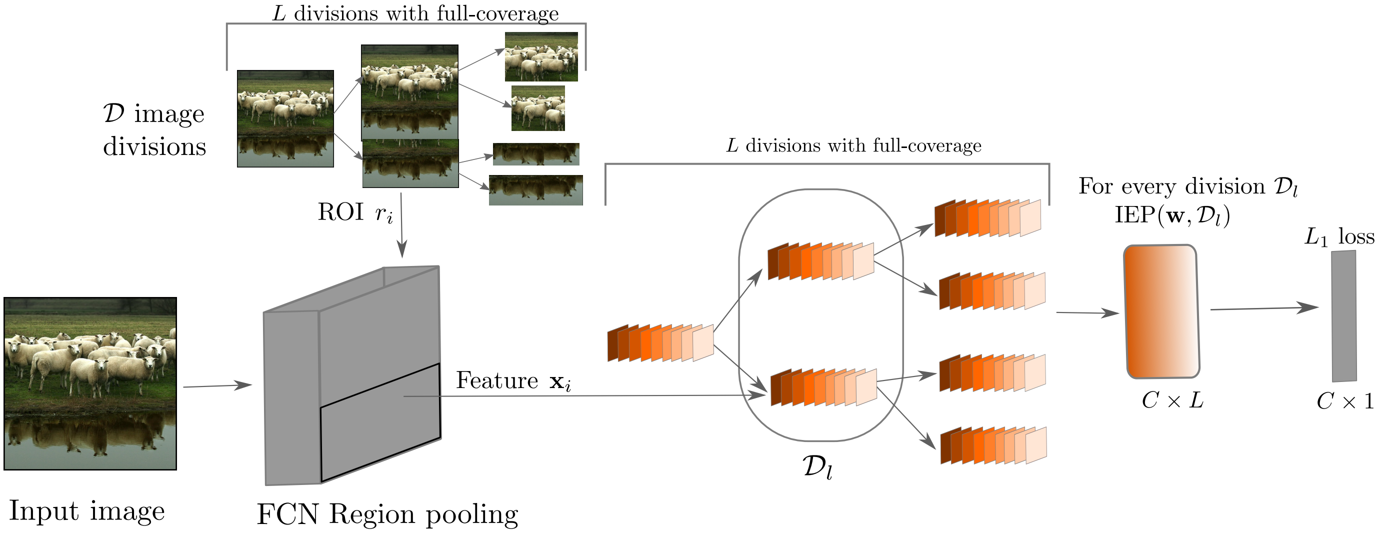

L divisions with full-coverage
$\mathcal{D}$ image divisions
ROI r_i
L divisions with full-coverage
For every division $\mathcal{D}_l$
IEP$(\mathbf{w}, \mathcal{D}_l)$
L_1 loss
Feature $\mathbf{x}_i$
$C \times L$
$C \times 1$
$\mathcal{D}_l$
Input image
FCN Region pooling

Fig. 3. Our approach: we start with an fully convolutional network (FCN) architecture to which we input the image together with the hierarchy of image division, $\mathcal{D}$. Region features, $\mathbf{x}$, are extracted in the region pooling layer from each region in each image division $\mathcal{D}_l$. The IEP layer combines the region features per image division per object category, outputing $C \times L$ scores — where L is the total numer of image divisions and C is the total number of object categories. The average over image divisions is optimized in an L_1 loss with respect to the image-level object count per object category.

inclusion-exclusion principle applied to object counts is given by equation (2).

$$\text{IEP}(\text{Count}(\cdot), \mathcal{D}_l = \{r_i\}_{i \in \{1,..N_l\}}) = \sum_{k=1}^{N_l} \left((-1)^{k+1} \sum_{S_k \in \binom{\mathcal{D}_l}{k}} \text{Count}\Big(\bigcap_{r_i \in S_k} r_i \Big) \right), \quad (2)$$

where $\binom{\mathcal{D}_l}{k}$ are all possible subsets S_k containing k image regions, r_i, from the image division $\mathcal{D}_l$.

Function Count$(\cdot)$ returns real numbers, representing complete objects, parts of objects, or multiple objects. As an illustration, consider the hypothetical case where we would have ground truth object pixel masks, denoted by Mask$(\mathbf{o})$ for object $\mathbf{o}$ in an image. Then we could formally define the count for an image region, r_i, as:

$$\text{Count}(r_i) = \sum_{\mathbf{o} \in \text{GT}} \frac{\text{Mask}(\mathbf{o}) \cap r_i}{\text{Mask}(\mathbf{o})}, \quad (3)$$

where GT represents the set of ground truth objects. Thus, function Count$(\cdot)$ is precisely what we aim to learn, however we do so using only the total object counts for the whole image as supervision.

Count learning with IEP layer. During count regression we enforce that the sum of all object counts over regions in an image division, $\mathbf{D}_l$, has to be equal to the total image-level object counts. To avoid over-counting, especially when dealing with highly overlapping image regions, we explicitly incorporate the object counts of overlapping regions located at the intersection of bounding boxes by employing the inclusion-exclusion principle [40].

After having pooled the feature maps over the input regions, r_i, to obtain region features, $\mathbf{x}_i$, we estimate the counts in each image division, $\mathcal{D}_l$ by learning a weight $\mathbf{w}$ shared among image divisions, but learned per object category. Our Count$(\cdot)$ function learns jointly to predict counts for all object categories, due to the features $\mathbf{x}_i$ being shared among object categories. The IEP$(\text{Count}_{\mathbf{w}}(r_i)_{r_i \in \mathcal{D}_l}, \mathbf{D}_l)$ layer aggregates image regions r_i in image division $\mathcal{D}_l$ using equation (2), where the count of a region r_i with features $\mathbf{x}_i = \Phi(r_i)$ is defined as:

$$\text{Count}_{\mathbf{w}}(r_i)_{r_i \in \mathcal{D}_l} = \mathbf{w}^T \Phi(r_i) = \mathbf{w}^T \mathbf{x}_i, \quad (4)$$

where the region feature $\mathbf{x}_i$ is shared among object categories, while the weights $\mathbf{w}$ are learned per object category. The object counts predicted by the IEP layer per image division are input to the L_1 loss (eq. (6)) and optimized such that each predicted image-division count is close to the global object count. At prediction time the overall image-level object counts are obtained as the average over the predictions of the IEP

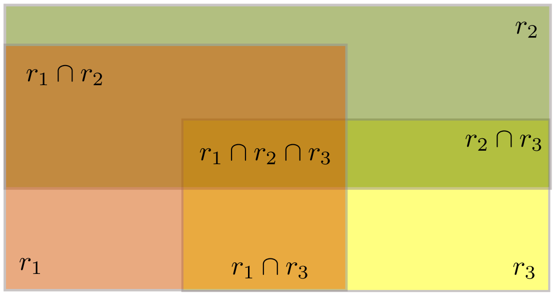


Fig. 4. Venn diagram depicting the inclusion-exclusion principle. The object count of division $\mathcal{D} = \{r_1, r_2, r_3\}$ depends on the counts of the intersection areas. The overall object count is estimated by adding the individual region counts of r_1, r_2, r_3, and subsequently subtracting the double counted regions ($\{r_1 \cap r_2\}, \{r_2 \cap r_3\}, \{r_1 \cap r_3\}$) and adding back the double subtracted $\{r_1 \cap r_2 \cap r_3\}$ region.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2875353, IEEE Transactions on Image Processing

layer per image division:

$$y = \frac{1}{L}\sum_{\mathcal{D}_l} \max(0, \lfloor \text{IEP}(\text{Count}_{\mathbf{w}}(r_i)_{r_i \in \mathcal{D}_l}, \mathbf{D}_l)\rfloor) \quad (5)$$

$$\begin{aligned}\mathcal{L}(y^*, y) = \frac{1}{L}\sum_{\mathcal{D}_l} \mid \max(0, \lfloor \text{IEP}(\text{Count}_{\mathbf{w}}(r_i)_{r_i \in \mathcal{D}_l}, \mathbf{D}_l)\rfloor) \\ - y^* \mid + \alpha\|\mathbf{w}\|_2, \quad (6)\end{aligned}$$

where the $\lfloor \cdot \rfloor$ converts the predicted counts to integers, as the total object counts in any image are integers. The term α controls the importance of the regularization, where$\|\mathbf{w}\|_2$ is the norm of $\mathbf{w}$. By not punishing the negative predictions in the loss function we allow more flexibility by focusing the model on correctly predicting the positive counts.

In our experiments we also test a version of our approach in which the final prediction is evaluated on the full image, while the training is still performed over all the levels in the hierarchy of region proposals, as described above. This version tests the added value of optimizing our model by backpropagating a loss per hierarchy level as in eq. (6) during training.

IV. Experimental Evaluation

A. *Experimental Setup*

We evaluate our specific model choices on the Pascal-VOC2007 dataset [12]. We additionally, evaluate our method on MS-COCO [26] large scale generic object dataset, and three class-specific counting datasets: pedestrian counting in the UCSD dataset [3], and two vehicle counting datasets: CARPK and PUCPR+ [10]. We use the standard SGD (Stochastic Gradient Descent) with the learning rate decay $\gamma = 0.5$, a momentum $\beta = 0.1$ and a starting learning rate of 0.005. For generating image divisions we use the selective search method [42] where for efficiency we extract object region proposals only over the HSV color space, giving rise to ≈ 500 regions per image. Our IEP layer outputs $C \times L$ predictions, where C is the number of object categories in the dataset and L is the depth of the hierarchy of image divisions for the current image. Unless indicated otherwise, we initialize our model with Resnet-50 features [16] pretrained on ImageNet [36]. The final prediction is $C \times 1$, as we have one model that jointly learns counts for all object categories. For all experiments we report MAE (Mean Absolute Error) or MSE (Mean Squared Error).

B. *Experiment 1: Model Choices*

We analyse on Pascal-VOC2007 the contributions of the individual building blocks in our method: (1) the effect of the hierarchy depth; (2) the importance of the preciseness of the localization given by the image regions; (3) the added value of the IEP-based counting; (4) the IEP layer generalization on a different backbone architecture. We evaluate our model choices on the Pascal-VOC2007 [12] dataset by training on the *training* set and evaluating on the *test* set.

TABLE I
Experiment 1.(1): MAE on Pascal-VOC2007 when using a hierarchy of image divisions rather than a flat structure. Using a hierarchy improves the performance as the optimization is performed independently over each depth — image division. We underline the approaches exceeding the baselines, and highlight in bold the best result.

	MAE
Grid 3 × 3	0.139
Grid 5 × 5	0.146
Grid 7 × 7	0.168
Hierarchical Grid	0.136
Full Image	0.143
IEP Counting	0.134
IEP Counting*	**0.129**

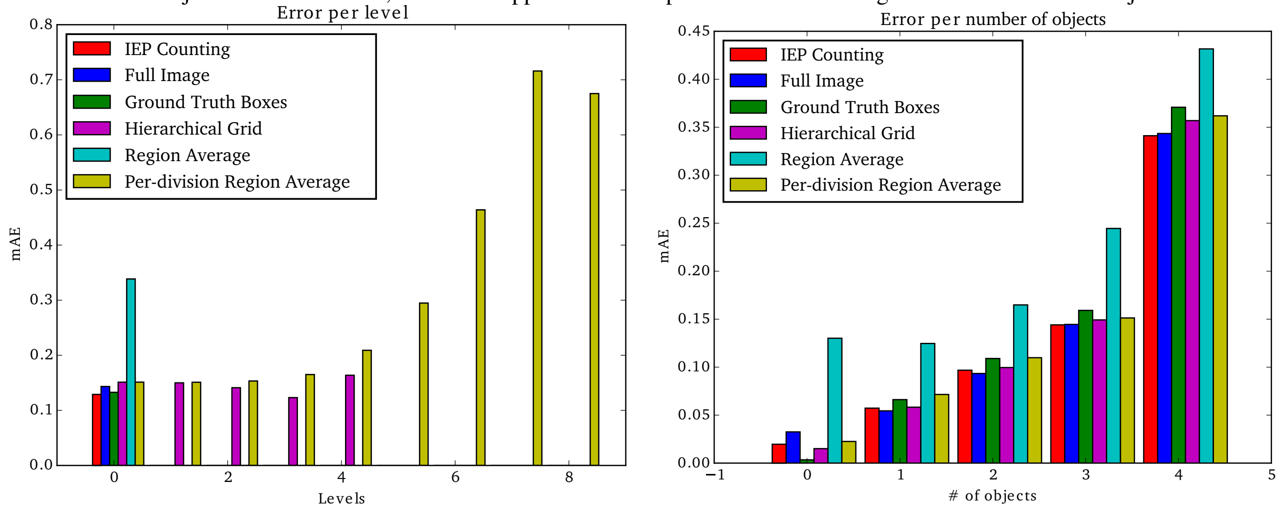
Experiment 1.(1): What is the effect of hierarchy depth? We evaluate the predictions of our model at each depth in our hierarchy of proposals. The hierarchy is trained jointly, yet we evaluate individual hierarchy depths to see if there is a discrepancy in the errors that the network makes at different hierarchy depths — region granularity levels. Each depth is optimized to predict the same global object counts and therefore the errors remain stable with the increase in depth. From depth 7 onwards the error increases slightly which indicates that the image region counts become more noisy. The results are plotted in Figure 5.(a).

Table I analyzes the gain of using a hierarchy of proposal rather than a flat structure. For the *Hierarchical Grid* we use a grid with depth 5 composed of: full image, 1×2, 2×2, 3×3 and 4×4 cells. The *IEP Counting* is based on the hierarchy obtained over selective search region proposals as described in Section III-B.(b). The more fine-grained the region proposal are, the more difficult the learning problem becomes. However, learning over a hierarchy rather than a flat structure helps, as each hierarchy level is independently optimized in the L_1 loss. Our *IEP Counting** approach uses at test-time the same model as *IEP Counting* for predicting. However, rather than predicting on all boxes in the hierarchy, it predicts only on the full image. Therefore, in this case the hierarchy of region proposals is used only at train-time. This validates the added value of performing the optimization per hierarchy level.

Experiment 1.(2): What is the importance of the preciseness of the localization? We test the importance of the preciseness of the localization for the counting problem. Table II shows the MAE with respect to a decreasing level of preciseness in the region locality. The *Ground Truth Boxes* contains the true object regions in each image and represents our upper bound. In our method we use only global image-level counts as supervision, therefore no bounding box or point annotations. The *IEP Counting* uses a hierarchy of region proposals that are self-contained within certain criteria, while the *Hierarchical Grid* uniformly segments the image which may result in object parts being separated in different grid cells. This explains why *IEP Counting* approach outperforms the *Hierarchical Grid.*

Experiment 1.(3): What is the added value of the IEP-based counting? The motivation of the IEP layer is twofold:

Fig. 5. **Experiment 1:** (a) MAE with respect to the considered depth of the hierarchy of image divisions evaluated on Pascal-VOC2007. The error tends to remain stable with the increase in the depth of the hierarchy, while slightly increasing from depth 7 onwards, which indicates the image region counts become more noisy. (b) MAE with respect to the average number of object per image. The error increases slightly with the number of objects for all methods, while our approach is less prone to over-counting for lower number of objects in the image.



(a) MAE wrt hierarchy depth.

(b) MAE wrt average number of objects.

TABLE II
EXPERIMENT 1.(2): MAE ON PASCAL-VOC2007 WITH DECREASING DEGREE OF PRECISENESS OF LOCALIZATION. *The Ground Truth Boxes* IS OUR UPPER BOUND AS WE USE GLOBAL IMAGE-LEVEL COUNTS AND NO LOCAL SUPERVISION IN OUR TRAINING. THE *IEP Counting* USES A HIERARCHY OF REGION PROPOSAL BASED ON IMAGE CUES AND, THUS PERFORMS BETTER THAN THE *Hierarchical Grid*. WE HIGHLIGHT IN BOLD THE BEST RESULT.

	MAE
Ground Truth Boxes	**0.133**
Hierarchical Grid	0.146
IEP Counting	0.134

TABLE III
EXPERIMENT 1.(3): TESTING THE IMPORTANCE OF THE IEP-BASED COUNTING ON PASCAL-VOC2007. THE *Region Average* OPTIMIZES THE AVERAGE COUNTS OVER ALL IMAGE REGIONS WITHOUT THE HIERARCHY. THE *Per-division Region Average* OPTIMIZES FOR EACH IMAGE DIVISION THE AVERAGE OVER THE PREDICTED OBJECT COUNTS. THE *IEP Counting* OUTPERFORMS THE OTHER TWO METHODS AS IT USES THE IEP OPTIMIZATION TO AVOID OVER-COUNTING. WE UNDERLINE THE APPROACHES EXCEEDING THE BASELINES, AND HIGHLIGHT IN BOLD THE BEST RESULT.

	MAE
Region Average	0.339
Per-division Region Average	0.158
IEP Counting	0.134
IEP Counting*	**0.129**

TABLE IV
EXPERIMENT 1.(4): MAE ON PASCAL-VOC2007 WHEN USING AS A BACKBONE ARCHITECTURE THE RESNET-50 AND RESNET-101 [16]. WHEN THE UNDERLYING ARCHITECTURE IS MORE DESCRIPTIVE AND FLEXIBLE, IT AIDS THE COUNTING TASK AS THERE IS A VISIBLE GAIN IN PERFORMANCE. WE HIGHLIGHT IN BOLD THE BEST RESULT.

	Backbone architecture	
	Resnet-50	Resnet-101
IEP Counting	0.134	**0.111**
IEP Counting*	0.129	**0.101**

(i) performing the counting optimization independently per image division, where each image division has a specific level of granularity of image regions; and (ii) avoiding over-counting for highly overlapping image regions. Here we test the importance of the IEP-based counting within both of these aspects. Table III shows the MAE when using the average over all proposal regions — *Region Average*, and when optimizing the average over the counts of all proposal regions in each image division — *Per-division Region Average*, and when optimizing the IEP counts in each image division — *IEP Counting*. Note that our chosen backbone architecture is the fully convolutional R-FCN architecture [25]. The region features input to the IEP layer are obtained from the region-pooling layer of the R-FCN architecture [25]. Our *IEP* layer can be seen as a variant of a fully connected layer, while the *Region Average* and *Per-division Region Average* retain the fully convolutional nature of the network as they are two different approached of pooling the region features. The *IEP Counting* considerably outperforms the two approaches as it avoid over-counting highly overlapping image regions. This is also conveyed in Figure 5.(b) where both the *Region Average* and *Per-division Region Average* tend to over-count the objects especially for images where fewer object are present.

Experiment 1.(4): How does the IEP layer generalize on another architecture? This experiment tests the behavior of our proposed IEP layer when used with a different backbone architecture. We compare the results obtained with Resnet-50 [16] as a starting point, pretrained on ImageNet [36], with

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2875353, IEEE Transactions on Image Processing

TABLE V
EXPERIMENT 1.(5): VOC2007 OBJECT COUNTING PERFORMANCE. WE COMPARE OUR PROPOSED METHOD *IEP Counting**, WITH THE RESULTS OF [6] USING A SET OF THEIR MODELS. THE GLANCE MODELS, *glance-noft-2L* AND *glance-sos-2L* USE FULL-IMAGE SUPERVISION AND PERFORM ON PAR WITH OUR PROPOSED METHOD, WHILE *ens* RELIES ON AN LARGE ENSAMBLE OF DEEP NETWORKS, SOME USING FULL IMAGE SUPERVISION AND OTHER USING PRECISE BOUNDING-BOX ANNOTATIONS, WHICH MAKES THE METHOD MORE ACCURATE BUT ALSO MORE EXPENSIVE BOTH IN TERMS OF RESOURCES AS WELL AS SUPERVISION. WE PROPOSE A MORE SIMPLE, THEORETICALLY MORE PRINCIPLED METHOD FOR OBJECT COUNTING.

Method	Supervision	mRMSE
ens [6]	Full	0.42 (± 0.170)
glance-noft-2L [6]	Weak	0.50 (± 0.020)
glance-sos-2L [6]	Weak	0.51 (± 0.020)
IEP Counting* (ours)	Weak	0.51 (± 0.007)

the results obtained when using Resnet-101 [16] pretrained on ImageNet [36]. In both case we still train from scratch the additional convolutional layer that makes the transition from the backbone architecture to the IEP layer and the weights in the IEP layer. Table IV shows that our proposed IEP layer is not architecture dependent and it can be used with any other backbone architecture, with the only restriction that the network should contain a ROI pooling layer. Moreover, if the underlying architecture is more descriptive, this aids the counting performance. For our subsequent experiments we use the Resnet-50 architecture for time efficiency.

Experiment 1.(5): Comparison with existing generic object counting. In Table V we compare with the very recent generic object counting work in [6]. In [6] a set of deep learning based methods are proposed for solving counting, among which: *glance-noft-2L* and *glance-sos-2L* use full image supervision, and their counting performance is on par with us. The best model of [6], *ens*, performs better than us by using an aggregation of deep learning counting architectures, where the supervision level ranges from full image to precise bounding boxes, and percentages of overlap with ground truth bounding boxes. We proposed a more simple, method for counting, which achieves competitive results.

C. Experiment 2: MS-COCO Experimental Evaluation

We evaluate our approach on MS-COCO [26] large-scale generic object dataset. We train on the *training* set of MS-COCO and predict on *val*. Counting objects on a generic dataset is more challenging than counting objects for a specific class, because the method must be able to count equally well objects that typically appear in large numbers such as *sheep* and *bottles*, as well as objects that are photographed alone, such as *cats* and *dogs*.

In Table VI we show the performance on the MS-COCO large-scale generic object dataset for our method compared with a naive baseline as well as the full image baseline. We report the MAE of our proposed approach, *IEP Counting**, when compared with *Always-1* — always predicting count 1, and the *Full Image* baseline — the network prediction using the full image. For our *IEP Counting** approach we use the hierarchy of boxes only at training-time, to learn stronger models, while for predicting, we only predict on the full image, as we have noticed that the lower levels in the hierarchy corresponding to fine-grained boxes tend to add noise to the count prediction. The *IEP Counting** outperforms the *Full Image* baseline, despite the highly challenging setting of the experiment.

TABLE VI
EXPERIMENT 2: MS-COCO PERFORMANCE. WE COMPARE OUR PROPOSED METHOD *IEP Counting**, WITH *Always-1* — ALWAYS PREDICTING COUNT 1, AND THE *Full Image* BASELINE — THE NETWORK COUNT PREDICTION OVER THE FULL IMAGE. WE HIGHLIGHT IN BOLD THE BEST RESULT.

	MAE (MSE)
Always-1	1.018 (1.359)
Full Image	0.111 (0.517)
IEP Counting* (ours)	**0.092** (**0.499**)

TABLE VII
EXPERIMENT 3.(1) COMPARATIVE MSE SCORES ON THE UCSD PEDESTRIAN COUNTING DATASET. OUR METHOD ACHIEVE COMPARABLE RESULTS, WHILE NOT RELYING ON PERSON-SPECIFIC MODELS AS [9], [13], OR DETAILED MOTION SEGMENTATION [5].

	MSE
N. Dalal and B. Triggs [9]	39.75
P. Felzenszwalb et. al [13]	24.72
A Chan and N Vasconcelos, [5]	9.95
Full Image	26.89
IEP Counting* (ours)	24.73

D. Experiment 3: Class-specific Counting

Experiment 3.(1): Pedestrian counting. This experiment evaluates our performance on a class-specific problem and compares with existing prior work [5], [9], [13]. For this we use the UCSD pedestrian counting dataset [3] and compare with three other existing methods.

Table VII depicts our results. For comparison with existing work, we report only MSE scores here. Our method manages to achieve comparable performance without employing person-specific models as in the case of [9], [13], or detailed motion segmentation masks as in [5]. This experiment validates the generality of our counting approach.

Experiment 3.(2): Car counting. We evaluate the task of vehicle counting using the datasets CARPK and PUCPR+ [10]. We compare our performance on the class-specific counting problem with the very recent counting approaches in [17], [29] as well as competitive object detection approaches [33], [35] fine-tuned on the CARPK and PUCPR+ datasets. Following [17], we use the same data setup, training on the *training* set and evaluating on the *test* set.

All methods are fine-tuned on the CARPK and PUCPR+ dataset, respectively. [33], [35] perform counting by object detection, while [29] is the most similar to us as it performs one-look counting over more loose localization given by

TABLE VIII

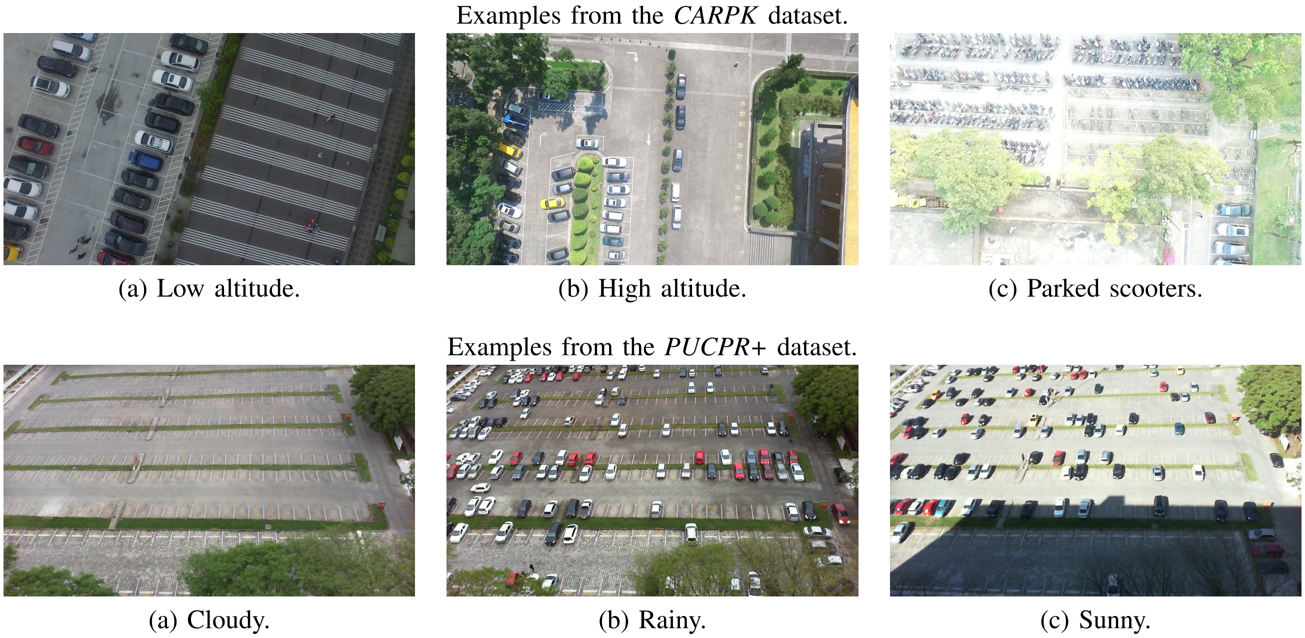
Experiment 3.(2): Comparative MAE scores on the CARPK and PUCPR+ vehicle counting datasets. We compare our method with the recent counting methods of [17], [29] relying on patch count annotations or precise bounding-box count annotations, as well as the object detection methods [33], [35] fine-tuned on CARPK and PUCPR+ dataset, respectively. Our method does not rely on ground truth bounding boxes and yet achieves similar performance to the recent counting methods. We underline the approaches exceeding the baselines, and highlight in bold the best result.

	Annotation level	MAE
YOLO [33]	box	48.89
Faster R-CNN [35]	box	47.45
LPN counting [17]	box count	23.80
One look regression [29]	patch count	59.46
IEP Counting (ours)	image count	52.69 (± 0.884)
IEP Counting* (ours)	image count	51.83 (± 0.883)

(a) MAE results on the CARPK dataset.

	Annotation level	MAE
YOLO [33]	box	156.00
Faster R-CNN [35]	box	111.40
LPN counting [17]	box count	22.76
One look regression [29]	patch count	21.88
IEP Counting (ours)	image count	15.78 (± 5.18)
IEP Counting* (ours)	image count	**15.12** (± 4.79)

(b) MAE results on the PUCPR+ dataset.



Examples from the *CARPK* dataset.

(a) Low altitude. (b) High altitude. (c) Parked scooters.

Examples from the *PUCPR+* dataset.

(a) Cloudy. (b) Rainy. (c) Sunny.

Fig. 6. **Experiment 3.(2):** Examples from the *CARPK* and *PUCPR+* datasets. In the *CARPK* dataset the resolution of the cars and the viewpoint and location vary. Moreover, in the *CARPK* dataset there are similar concepts present such as: buses and scooters. Dissimilar, the *PUCPR+* records a parking lot, under different weather conditions, with a fixed camera, and there are no other dynamic objects present apart from cars.

patches. However, the method in [29] uses patch annotations where patches in the image have associated annotations with the number of cars present. In our case we use patches, but we do not have associated patch counts, we only have the total global counts of all cars present in the image. In [17] a Layout Proposal Network is used to first localize the cars and subsequently count. Dissimilar to us, the work in [17] employs precise ground truth bounding boxes to jointly localize and count the cars present in the images. For this dataset the location information is important, as detection methods relying on tight bounding box annotations outperform counting methods such as [29] and our approach.

Table VIII depicts our results. On the CARPK dataset, in table VIII.(a), our method achieves comparable performance with the recent work of Mundhenk et al. [29], without using region-based counting supervision. The work of Hsieh et al. [17] performs better than the work in [29] and our work, as it uses additional supervision by employing precise bounding box annotations during training. On PUCPR+ dataset, in table VIII.(b), we outperform all detection baselines [33], [35]. Our method has a large variance on this dataset, which we believe to be caused by the very limited number of training examples: in the range one hundred. However, our method performs on par with the recent counting methods [17], [29], despite not using region counts, or bounding box ground truth annotations. This experiment validates the generality of our counting approach, which can be applied either for generic object counting as in experiment IV-C, or for class-specific counting as tested here.

Cross-dataset variance analysis on the task of car counting. There is a large variance in performance between the two datasets for most of the methods reported in Table VIII. The *CARPK* dataset contains an aerial video recorded from different altitudes and in different locations. The resolution of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2875353, IEEE Transactions on Image Processing

cars varies considerably throughout the video, as seen in the first row of Figure 6. Moreover, in the *CARPK* dataset other concepts are present, that may confuse the car counting: e.g. buses, pedestrians, scooters, etc. The supervision is provided only for cars, the other concepts being ignored.

Unlike *CARPK*, the *PUCPR+* dataset contains images from a fixed surveillance camera, filming a parking lot under different weather conditions: e.g. sunny, rainy, cloudy. In this dataset there are no other dynamic objects present, apart from cars. A few examples from the *PUCPR+* dataset are displayed on the second row of Figure 6.

Table VIII shows that counting methods based on object detection such as [33] and [35] perform considerably better on the *CARPK* dataset, while having a large error on the *PUCPR+* dataset. On the contrary, counting methods such as ours and [29] perform well on the *PUCPR+* dataset while having lower performance on the *CARPK* dataset. The method in [17] has a consistent performance on both datasets as it relies on car-localization, with bounding box supervision, and it fine-tunes both the car localization and the car counting on the *CARPK* and *PUCPR+* dataset. The object detection methods [33], [35] are also fine-tuned on the two car- datasets. The *CARPK* dataset has $\approx 10 \times$ more samples than the *PUCPR+* dataset, and therefore the fine-tuning of the car detectors is more effective on the *CARPK* dataset than on the *PUCPR+* dataset. The number of training samples per dataset, is one motivation for the variance in performance across datasets. Another reason for this variance across datasets, is the confusion generated in car counts by other objects present in the *CARPK* dataset such as scooters and pedestrians, which do not have associated object counts. Object detectors are optimized to distinguish cars from scooters or other objects and, therefore, are more accurate on the *CARPK* dataset. Our method is only trained with the object count labels, which makes it more challenging for it not to over-count in the presence of unlabelled similar classes.

V. Conclusion

This work proposes an approach towards generic object counting with unsupervised local image information. Our method relies on unsupervised object proposals or uniform grid partitions and adds geometric information in the loss optimization through the inclusion-exclusion principle. Moreover, we propose to learn from local image features, and predict global image object counts. Therefore, we do not rely on any form of local supervision. In experimental section IV-B we analyse the added value of each one of the building blocks composing our proposed approach: the effect of the depth of the hierarchy of image divisions on the counting performance, the importance of the preciseness of locality offered by object regions, the added value of the IEP-based counting, as well as the generality of the IEP layer when applied on a different architecture. Finally, in the section IV-C we evaluate our proposed method on the large scale MS-COCO dataset, while IV-D tests the generality of our method on three class-specific dataset: pedestrian counting in the UCSD dataset, and two vehicle counting dataset, CARPK and PUCPR+, and compares with recent object detection and counting works.

References

[1] C. Arteta, V. Lempitsky, and A. Zisserman. Counting in the wild. In *ECCV*, pages 483–498, 2016.

[2] A. B. Chan, Z. S. J. Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *CVPR*, pages 1–7, 2008.

[3] A. B. Chan and N. Vasconcelos. Modeling, clustering, and segmenting video with mixtures of dynamic textures. *PAMI*, 30(5):909–926, 2008.

[4] A. B. Chan and N. Vasconcelos. Bayesian poisson regression for crowd counting. In *ICCV*, pages 545–551, 2009.

[5] A. B. Chan and N. Vasconcelos. Counting people with low-level features and bayesian regression. *TIP*, 21(4):2160–2177, 2012.

[6] P. Chattopadhyay, R. Vedantam, D. Batra, and D. Parikh. Counting everyday objects in everyday scenes. *CVPR*, 2017.

[7] K. Chen, C. C. Loy, S. Gong, and T. Xiang. Feature mining for localised crowd counting. In *BMVC*, 2012.

[8] S. Chen, A. Fern, and S. Todorovic. Person count localization in videos from noisy foreground and detections. In *CVPR*, pages 1364–1372, 2015.

[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893, 2005.

[10] P. R. De A, L. S. Oliveira, A. S. Britto, E. J. Silva, and A. L. Koerich. Pklot–a robust dataset for parking lot classification. *Expert Systems with Applications*, 42(11):4937–4949, 2015.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*. IEEE, 2009.

[12] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[13] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, pages 1–8, 2008.

[14] L. Fiaschi, U. Köthe, R. Nair, and F. A. Hamprecht. Learning to count with regression forest and structured labels. In *ICPR*, pages 2685–2688, 2012.

[15] R. Guerrero-Gómez-Olmedo, B. Torre-Jiménez, R. López-Sastre, S. Maldonado-Bascón, and D. Onoro-Rubio. Extremely overlapping vehicle counting. In *ICPRIA*, pages 423–431, 2015.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[17] M. Hsieh, Y. Lin, and W. Hsu. Drone-based object counting by spatially regularized regional proposal network. In *ICCV*, volume 1, 2017.

[18] H. Idrees, I. Saleemi, C. Seibert, and M. Shah. Multi-source multi-scale counting in extremely dense crowd images. In *CVPR*, pages 2547–2554, 2013.

[19] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. *arXiv preprint arXiv:1612.06890*, 2016.

[20] A. Khan, S. Gould, and M. Salzmann. Deep convolutional neural networks for human embryonic cell counting. In *ECCV*, pages 339–348, 2016.

[21] D. Kong, D. Gray, and H. Tao. A viewpoint invariant approach for crowd counting. In *ICPR*, volume 3, pages 1187–1190, 2006.

[22] P. Krähenbühl and V. Koltun. Geodesic object proposals. In *ECCV*, pages 725–739, 2014.

[23] V. Lempitsky and A. Zisserman. Learning to count objects in images. In *NIPS*, pages 1324–1332, 2010.

[24] M. Li, Z. Zhang, K. Huang, and T. Tan. Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In *CVPR*, pages 1–4, 2008.

[25] Y. Li, K. He, J. Sun, et al. R-fcn: Object detection via region-based fully convolutional networks. In *NIPS*, pages 379–387, 2016.

[26] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014.

[27] X. Liu, Z. Wang, J. Feng, and H. Xi. Highway vehicle counting in compressed domain. In *CVPR*, pages 3016–3024, 2016.

[28] L. Maddalena, A. Petrosino, and F. Russo. People counting by learning their appearance in a multi-view camera environment. *Pattern Recognition Letters*, 36:125–134, 2014.

[29] T. N. Mundhenk, G. Konjevod, W. A. Sakla, and K. Boakye. A large contextual dataset for classification, detection and counting of cars with deep learning. In *ECCV*, pages 785–800, 2016.

[30] D. Onoro-Rubio and R. J. López-Sastre. Towards perspective-free object counting with deep learning. In *ECCV*, pages 615–629, 2016.

[31] V. Pham, T. Kozakaya, O. Yamaguchi, and R. Okada. Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation. In *ICCV*, pages 3253–3261, 2015.

[32] V. Rabaud and S. Belongie. Counting crowded moving objects. In *CVPR*, volume 1, pages 705–711, 2006.
[33] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016.
[34] M. Ren and R. S. Zemel. End-to-end instance segmentation and counting with recurrent attention. *CoRR*, 2016.
[35] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.
[36] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.
[37] D. Ryan, S. Denman, C. Fookes, and S. Sridharan. Scene invariant multi camera crowd counting. *Pattern Recognition Letters*, 44:98–112, 2014.
[38] S. Seguí, O. Pujol, and J. Vitria. Learning to count with deep object features. In *CVPR Workshops*, pages 90–96, 2015.
[39] J. Shao, D. Wang, X. Xue, and Z. Zhang. Learning to point and count. *CoRR*, 2015.
[40] W. Szpankowski. Inclusion-exclusion principle. *Average Case Analysis of Algorithms on Sequences*, pages 49–72, 2001.
[41] B. Tamersoy and J. K. Aggarwal. Counting vehicles in highway surveillance videos. In *ICPR*, pages 3631–3635, 2010.
[42] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013.
[43] J. van Gemert, C. Verschoor, P. Mettes, K. Epema, L. Koh, and S. Wich. Nature conservation drones for automatic localization and counting of animals. In *ECCV Workshop on Computer Vision in Vehicle Technology*, 2014.
[44] M. von Borstel, M. Kandemir, P. Schmidt, M. K. Rao, K. Rajamani, and F. A. Hamprecht. Gaussian process density counting from weak supervision. In *ECCV*, pages 365–380, 2016.
[45] E. Walach and L. Wolf. Learning to count with cnn boosting. In *ECCV*, pages 660–676, 2016.
[46] J. Wawerla, S. Marshall, G. Mori, K. Rothley, and P. Sabzmeydani. Bearcam: Automated wildlife monitoring at the arctic circle. *Machine Vision and Applications*, 20(5):303–317, 2009.
[47] W. Xie, J. A. Noble, and A. Zisserman. Microscopy cell counting and detection with fully convolutional regression networks. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, pages 1–10, 2016.
[48] Y. Xue, N. Ray, J. Hugh, and G. Bigras. Cell counting by regression using convolutional neural network. In *ECCV*, pages 274–290, 2016.
[49] C. Zhang, H. Li, X. Wang, and X. Yang. Cross-scene crowd counting via deep convolutional neural networks. In *CVPR*, pages 833–841, 2015.
[50] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma. Single-image crowd counting via multi-column convolutional neural network. In *CVPR*, pages 589–597, 2016.
[51] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *CPVR*, 2016.
[52] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, pages 391–405, 2014.