Object Counting with Small Datasets of Large Images

Shubhra Aich and Ian Stavness
Department of Computer Science
University of Saskatchewan, Canada
{s.aich,ian.stavness}@usask.ca

Abstract

We explore the problem of training one-look regression models for counting objects in datasets comprising a small number of high-resolution, variable-shaped images. To reduce overfitting when training on full resolution samples, we propose to use global sum pooling (GSP) instead of global average pooling (GAP) or fully connected (FC) layers at the backend of a convolutional neural network. Although computationally equivalent to GAP, we show via detailed experimentation that GSP allows convolutional networks to learn the counting task as a simple linear mapping problem generalized over the input shape and the number of objects present. We evaluate our approach on four different aerial image datasets – three car counting datasets (CARPK, PUCPR+, and COWC) and one new challenging dataset for wheat spike counting. Our GSP approach achieves state-of-the-art performance on all four datasets and GSP models trained with smaller-sized image patches localize objects better than their GAP counterparts.

1 Introduction

Increasingly complex and large deep learning architectures are being devised to tackle challenging computer vision problems, such as object detection and instance segmentation with hundreds of object classes [6, 17, 20]. However, as deep learning permeates specialized applications of computer vision, it is becoming common to deploy highly complex state-of-the-art architectures to solve substantially simpler tasks. Object counting is a good example of such a task because there are many practical applications that require counting the occurrences of the object instances within an image: counting cars on a freeway or in a parking lot, counting people in a crowd, and counting plants or trees from aerial images. While it is possible to apply very powerful instance segmentation [11] or object detection [27] approaches to counting problems, these architectures require detailed (and time-consuming and tedious to collect) annotations, such as instance segmentation masks or bounding boxes. However, object counting is amenable to weaker labels, such as dot annotations (one dot per instance) or a scalar count per image. Devising simpler deep learning models for less complex computer vision tasks has the benefit of less costly ground-truth labeling, smaller sized networks, more efficient training, and faster inference.

One-look regression models are a class of deep neural network that are well matched to the comparatively simpler problem of object counting. These models use a convolutional backbone combined with fully-connected (FC) or global average pooling (GAP) layers that end in a single unit to generate a scalar count of the number of object instances present in the image [1, 2, 8, 36]. Other variants of this counting network use a final classification layer, where the number of the output units are slightly more than the maximum number of possible object instances in the input [22]. This requires that the maximum number of object instances are known a priori, which may be difficult when the number of objects varies with the size of the input. Therefore, in this paper, we focus only on the single output unit models for object counting.
Counting datasets have two common characteristics that complicate the training of one-look models. First, the training set typically consists of a few very high-resolution images. Despite the computational complexity, it might be possible to train on full sized images as a whole, but there is a high probability of overfitting by blindly memorizing the scalar counts because of the small number of training samples available. Second, in datasets of aerial imagery for counting cars, shipping containers, plants, etc., objects often appear at a similar scale across images (since images are often captured at the same altitude), but the images vary in size because they are stitched or cropped to a particular region of interest. For example, agricultural studies often want to count plant characteristics within a specific portion of a field. Many architectures require a pre-defined size for training / test images, and warping aerial images to that pre-defined size may destroy the harmony of the object resolution over the whole dataset.

A common solution to overcome the challenge of high-resolution, variable-sized images is to use smaller sized, randomly cropped “patches” from the high-resolution raw training images to train the network. Dot annotations can be counted to create an object count per patch, but because the full extent of the object is not annotated, it is not possible to generate patches without partially cutting the objects at the edge of the patch. This type of label noise may be acceptable during training, but at test time, when a total count is required for the high-resolution image, tiled patches would need to be applied to the network and the counts per tile summed. Consequently, even though the model is capable of detecting or counting these partial objects, summing up the counts of the adjacent tiles to retrieve the final count might cause an unwanted overestimate or underestimate in the prediction depending on the ability of the model to detect partial or cropped instances. Previous work has attempted to resolve these patch-wise inference errors empirically, by optimizing the stride of tiled patches based on validation set performance [22]. However, this does not address the fundamental limitation of partial object instances; resulting in unavoidable per-patch counting errors, which are then propagated to the estimate of the full image count.

Considering the complications of datasets with a small number of high-resolution variable-sized images, an ideal solution would be a particular kind of model that can be trained with small-sized random patches (to reduce the risk of overfitting or memorization) and then generalize its performance over arbitrarily large resolution test samples. In this paper, we devise such a model using a set of traditional convolutional and pooling layers in the front-end and replacing the fully connected (FC) layers or global average pooling (GAP) layer with the new global sum pooling (GSP) operation. We show that the use of this GSP layer allows the network to train on image patches and infer accurate object counts on full sized images. Although from a computational perspective, the summation operation in GSP is very similar to the averaging operation in GAP, GSP exhibits the non-trivial property of generalization for counting objects over variable input shapes, which GAP does not. To the best of our knowledge, this is the first work introducing the GSP operation as a replacement of GAP or FC layers. We evaluated GSP models on four different datasets — three for counting cars —
and one for the challenging task of counting wheat spikes. Results demonstrate that GSP helps to generate more localized activations on object regions (Figure 1) and achieve better generalization performance which is consistent with our hypothesis.

2 Related Work

Much of the recent literature on object counting is based on estimating different kinds of activation maps because these approaches are applicable to datasets with high-resolution images. Lempitsky and Zisserman [18] incorporate the idea of per-pixel density map estimation followed by regression for object counting. This regression approach is further enhanced by [4] by adding an interactive user interface. Fiaschi et al. [9] employ random forest to regress the density map and object count. Fully convolutional network [39] is also used for contextual density map estimation irrespective of the input shape. Proximity map, which is the proximity to the nearest cell center, is also estimated in [40] as an alternative to traditional density map approximation. Another variant of density map is proposed in the Count-ception paper [7], where the authors use fully convolutional network [21] to regress the count map followed by scalar count retrieval adjusting the redundant coverage proportional to the kernel size. Wheat spike images have been previously investigated for controlled imaging environments using density maps [24].

There also exists an extensive body of work on crowd counting [33]. Here, we review some of the recent CNN based approaches. Wang et al. [38] employed a one-look CNN model first on dense crowd counting. Zhang et al. [41] developed the cross-scene crowd counting approach. They use alternative optimization criteria for counting and density map estimation. Also, instead of single Gaussian kernels to generate a ground truth density map, they use multiple kernels along with the idea of perspective normalization. Cross-scene adaptation is done by finetuning the network with training samples similar to test scenes. Similar to the gradient boosting machines [10], Walach and Wolf [37] iteratively add additional computational blocks in their architecture to train on the residual error of the previous block, which they call layered boosting. Shang et al. [31] use an LSTM [13] decoder on GoogLeNet [34] features to extract a patchwise local count and generate a global count from them using FC layers. CrowdNet [5] uses the combination of shallow and deep networks to acquire multi-scale information in density map approximation for crowd counting. Another approach [42] for multi-scale context aggregation for density map estimation use multi-column networks with different kernel sizes. Hydra CNN [23] employ three convolutional heads to process image pyramids and combine their outputs with additional FC layers to approximate the density map at a lower resolution. Switching CNN architecture [30] proposes a switching module to decide among different sub-networks to process images with different properties.

Most of the approaches described above attempt to approximate a final activation map under different names, i.e. density map, count map, and proximity map, which is then post-processed to obtain the count information; thus resulting in a multi-stage pipeline. In this regard, one-look models are simpler and faster than these map estimation approaches. The main idea behind using one-look regression models [1, 2, 8, 15, 22, 36, 38] for object counting is to utilize weak ground truth information, such as the object counts in the images, in contrast to more sophisticated models, like object detection [25, 27] or instance-level segmentation [11, 26, 28] that require stronger and more tedious to collect object labels. The domain knowledge of spatial collocation of cars in the car parks is exploited in the layout proposal network [14] to detect and count cars. The COWC dataset paper [22] uses multiple variants of the hybrid of residual [12] and Inception [35] architectures, called ResCeption, as the one-look model for counting cars patchwise. During inference, the authors determine the stride based on the validation set. This kind of hybrid models are also used in [1] to estimate plant characteristics from images. However, recent work on heatmap regulation (HR) [3] describes the philosophical limitation of using one-look models and tries to improve its performance by regulating the final activation map with a Gaussian approximation of the ground-truth activation map. In this paper, using GSP and training with smaller samples, we obtain similar final activation maps to HR without using any extra supervision channel in our model. Further, HR can be easily used in conjunction with GSP.

3 Our Approach

In order to overcome the generalization challenges of object counting from a small number of high-resolution, variable sized images, and to avoid problems with partial objects in counting from small
patches, we propose an architecture that can learn to count from images irrespective of their shape. Architectures with final FC layers pose strict requirements about the input images’ shape, whereas architectures that combine CNN with additional nonlinear and normalization layers are more flexible. We take inspiration from recent image classification architectures [12, 15, 19, 35] that replace FC layers with a simple GAP layer. Using GAP greatly reduces the number of parameters (to help reduce overfitting), emphasizes the convolutional front-end of the models, permits training and testing on variable size input images, and provides intuitive visualizations of the activation maps [43]. For an object counting task, however, the averaging operation of GAP lacks the ability to generalize over different sized input images.

This point can be illustrated by a hypothetical example. Assuming the resolution of object instances falls within a fixed range over a dataset, which is common for many counting problems that arise in the real-world (e.g., fixed altitude aerial imagery), and assuming (without loss of generality) that objects are uniformly distributed, the number of objects within an image is expected to scale with the image resolution. For example, if a $200 \times 200$ region contains 4 objects, then a $1000 \times 1000$ region would be expected to contain 100 objects. If we train a network containing a stack of convolution layers followed by a GAP layer on $200 \times 200$ samples, our models will learn to generate the expected count of 4 with an equivalent vector representation as the output of GAP. During inference, with a $1000 \times 1000$ image, the last convolution layer will generate 25 adjacent, spatial feature responses, each representing the expected count of 4. This convolutional representation is appropriate to predict an expected count of 100. However, the GAP layer will average over all the 25 spatial sub-regions and obtain an equivalent representation of 4. Hence, the averaging operation is not suitable for modeling the proportional scaling of the number of objects with the size of input. Instead of average-pooling the final feature maps, we propose a summation or mere aggregation of the input over the spatial locations only. From the previous example, this aggregation of 25 similar sub-regions, each with a count of 4, would produce the desired expected value of 100. Following the nomenclature of GAP, we call this operation global sum pooling (GSP). Although GAP and GSP are computationally similar operations, conceptually GSP provides the ability to use CNN architectures for generalized training and inference on variable shaped inputs in a simple and elegant way.

**Linear mapping:** Learning to count regardless of the input image shape necessarily means that the convolutional front-end of the network should learn a linear mapping task, where the output vector of GSP will scale proportionally with the number of objects present in the input image. Figure 2 shows a sample $720 \times 1280$ image from the CARPK [14] aerial car counting dataset. On the right of Figure...
2 we plot the largest 48 activations of the 512-vector output of the GSP layer of our model described later, for different sized sub-regions of the same sample image. Here, the elements are sorted in descending order for the full resolution image, and the same ordering is used for the activations of the sub-regions. The model producing these activations was trained on 224 × 224 randomly cropped samples. From this figure, it is evident that our model is able to learn a linear mapping function from the image space to the high-dimensional feature space, where the final count is a simple linear regression or combination of the extracted feature values.

Weak instance detector and region classifier: An advantage of training on small input sizes is that it guides the network to behave like a weak object instance detector even though we only provide weak labels (a scalar count per image region). Training on sub-regions of a large input image helps the network to better disambiguate the true object regions from the object-like background sub-regions, resulting in improved performance. For example, when training the network with full images, all of which have a non-zero object count, the network never faces a complete background sample from which it can extract background information similar to any binary region classification problem. On the other hand, when we train with small randomly-cropped regions of the input image many background-only samples are fed to the network, instantiating a more rigorous learning paradigm even with weak count labels. Class-activation map (CAM) [43] visualizations illustrate that the GSP model trained with small sub-regions better captures localization information (Figure 1). Training the GAP or GSP models with full-resolution images results in a less uniform distribution of activation among object regions and less localized activations inside object regions as compared to the GSP model trained with smaller patches.

Architecture: We attach a GSP layer after the convolutional front-end of VGG16 [32] model pretrained on ImageNet [29]. GSP produces a 512-dimensional vector, which is converted to a scalar count by a linear layer. We faced no problems with the potential numerical instability caused by large, unnormalized values after spatial summation, even when training the GSP models with full resolution images.

4 Experiments

Datasets: We evaluate object counting with GSP on 4 datasets: CARPK [14] (overhead view of different car parks), PUCPR+ [14] (slant view of a single car park), COWC [22] (overhead view of cars in residential areas and highways), and a wheat spike (WS) dataset [16] (overhead view of mature wheat plants). CARPK and PUCPR+ contain 720 × 1280 resolution images, whereas COWC and WS have large images with variable resolutions. All datasets have comparatively few training and test images. Statistics of these datasets are listed in Table 1.

Metrics: We adopt the same evaluation metrics reported in the HR [3] and COWC [22] papers along with one additional metric: the percentage of MAE over the mean of the target count, which we call the relative MAE (RMAE %).

Models & Training: We compare counting performance for our GSP model trained on full-sized images (GSP-Full) to GSP trained on randomly cropped patches: GSP-N where the patch size is $N \times N$. We also include statistics for the GAP version of our model trained with full resolution images (GAP-Full) and GAP trained with the best patch size found by GSP (GAP-N). Testing was performed on full sized test images for all conditions, except for GAP-Patch-N, where counts are predicted on tiled patches from the test image and summed to get a total count per image. We also compare to published baselines for the car counting datasets.

In order to train on image patches, we compute a count per patch based on the number of central object regions within the patch. The CARPK and PUCPR+ datasets provide bounding boxes, which

| Dataset          | #Images (Train, Test) | Resolution | Total Count (Train, Test) | Range of Count (Train, Test) |
|------------------|-----------------------|------------|---------------------------|-----------------------------|
| CARPK            | (989, 459)            | (720 × 1280) | (42274, 47500) | ([1, 87], [2, 188]) |
| PUCPR+           | (100, 25)             | (720 × 1280) | (12995, 3920) | ([0, 331], [1, 328]) |
| COWC             | (32, 20)              | ∼(18k × 18k) – (2k × 2k) | (37890, 3456) | ([45, 13086], [10, 881]) |
| Wheat-Spike      | (10, 10)              | ∼(1000, 3000) | (10112, 9989) | ([796, 1287], [749, 1205]) |
Figure 3: Activation maps for CARPK (left pair) and PUCPR+ (right pair) generated by the baseline model (left) and GSP-224 model (right). Activations are more uniformly distributed and more concentrated inside object regions for the GSP-224 model.

Table 2: Results on CARPK and PUCPR+ datasets

| Method      | MAE  | RMAE (%) | RMSE | %O  | %U  | MAE  | RMAE (%) | RMSE | %O  | %U  |
|-------------|------|----------|------|-----|-----|------|----------|------|-----|-----|
| GSP-64      | 32.09| 31.01    | 36.02| 30.81| 0.21| 37.44| 23.88    | 48.15| 23.88| 0.00|
| GSP-96      | 10.63| 10.27    | 11.37| 8.61 | 1.66| 19.64| 12.53    | 24.69| 11.56| 0.97|
| GSP-128     | 6.70 | 6.48     | 10.21| 1.79 | 4.69| 6.28 | 4.01     | 8.10 | 2.76 | 1.25|
| GSP-224     | 5.46 | 5.28     | 8.09 | 1.14 | 4.14| 4.52 | 2.88     | 7.30 | 1.33 | 1.56|
| GSP-HR-224  | 6.95 | 6.72     | 9.87 | 0.12 | 6.60| 4.20 | 2.68     | 6.16 | 1.12 | 1.56|
| GAP-Patch-224| 7.65 | 7.39    | 9.59 | 0.84 | 6.56| 5.64 | 3.60     | 7.01 | 2.07 | 1.53|
| GAP-Full    | 19.61| 18.95    | 21.65| 0.24 | 18.71| 6.32 | 4.16     | 9.25 | 0.28 | 3.88|
| GSP-Full    | 32.94| 31.83    | 36.23| 0.42 | 31.42| 7.64 | 4.80     | 9.56 | 0.13 | 4.74|
| LPN \[14\]  | 13.72| -        | 21.77| -   | -   | 8.04 | -        | 12.06| -   | -   |
| GAP-HR-Full \[3\]| 7.88 | -        | 9.30 | 0.71 | 6.91| 5.24 | -        | 6.67 | 2.73 | 0.61|

we shrink down to 25% along each dimension and to define a central region for each car instance. The shrinking prevents object regions from overlapping and makes it so that we only count objects that are mostly inside the cropped patch. The COWC and Wheat-Spike datasets provide dot annotations instead of bounding boxes, which we dilate to define a central region for each object instance, in order to increase the probability of counting partial objects in the cropped patches. The dots are assumed to be in the center of each object, which is not necessarily the case due to annotation error regarded as label noise.

**CARPK and PUCPR+ datasets:** For these car park datasets, we found that GSP models trained with $128 \times 128$ and $224 \times 224$ samples perform much better than the same model trained with smaller patches, like $64 \times 64$ and $96 \times 96$ (Table 2). Adding HR \[3\] to the GSP-244 model provided the best overall performance for PUCPR+. Figure 3 compares CAM heatmaps superimposed on original images for the baseline GAP-Full model and our best performing GSP-N model (N=224). The activation maps of the GAP model are variable over the object regions, indicating that some of the objects are being highly emphasized than others, whereas the GSP-224 activations are more uniform, showing that all the instances are getting more or less equal attention from the network. Moreover, the GSP-224 activations better localized within object sub-regions than GAP model, which demonstrates that GSP-N models with small N work as a better object detector or binary region classifier than the baseline models.

We believe that the poor performance of GSP-N models for smaller N (64 and 96) is not a characteristic of the model itself. Instead, the poor performance can be attributed to the training procedure that we followed in this paper. As already stated, we shrink the bounding boxes for CARPK and PUCPR+ datasets to disambiguate the overlapping bounding boxes. However, such shrinking poses restrictions on using arbitrarily small sample sizes in training. If the patch size is close to the object resolution (the average resolution of the bounding boxes in the training set of CARPK dataset is about 40
pixels) and the objects are close together (which cars are in a parking lot), a patch is likely to include one complete object with several other instances partially cut at the edge of the patch. Because we disambiguate object counts by shrinking the boxes, depending on the relative orientation between the object and its encompassing box, and its portion inside the cropped patch, it might be taken into account for counting or not. Therefore, this aspect of our training paradigm is a bit randomized. For comparatively larger sample size, such as 128 and 224, we anticipate that this problem of random consideration of the partial objects in the border is less frequent than the smaller sized patches, such as 64 and 96. In this regard, the optimal sample size depends on the average resolution of the object instances and their relative placement in the images of a particular dataset. Furthermore, for the PUCPR+ dataset, the performance is already very good with the baseline GAP model. This dataset contains images with a static background of the same car park and therefore could be considered a most straightforward object counting task. While the GAP-Patch-N model appears to perform reasonably well, this is due to large per-patch overestimates and underestimates canceling out when patch-wise counts are summed at test time. The per-patch RMAE was large for both datasets: 15.89% and 15.05% for CARPK and PUCPR+, respectively.

**COWC dataset:** COWC contains very few training images (32) and the image sizes vary substantially (2220 × 2220 to 18400 × 18075), therefore it is an ideal test case for the main features of GSP. Unlike the parking lot datasets, the COWC dataset contains images covering highways and residential areas and therefore cars in these images often appear to be entirely isolated objects in the roads or highways or parked in the residential streets. Each pixel covers 15 cm, resulting in the resolution of the cars ranging from 24 to 48 pixels. Because of the sparsity of the objects in the ultra-high-resolution training images, we extract ∼ 8000 samples of resolution 288 × 288 centered on object sub-regions from the images prior to training. We do this to avoid training on a large number of negative samples that would be the case for random cropping. Therefore, the GSP-288 represents the GSP model trained with “full-resolution” images for this dataset.

For COWC, the smallest patch size (GSP-64) provides comparable performance to previously published results (Table 3). We also see that the activations for GSP-64 are more concentrated on the objects in the image compared to that of GSP-224 (Figure 4). This observation is consistent with our claim that the GSP models trained with smaller sample size tend to localize objects better, particularly when they are relatively isolated from each other. We omit MAE (%) and RMSE (%)
Table 4: Results on Wheat-Spike dataset

| Method       | MAE   | RMAE (%) | RMSE   | %O     | %U     |
|--------------|-------|----------|--------|--------|--------|
| GSP-64       | 111.38| 11.15    | 130.07 | 9.76   | 1.39   |
| GSP-96       | 80.00 | 8.01     | 100.63 | 0.00   | 8.01   |
| GSP-128      | 61.81 | 6.19     | 78.34  | 0.28   | 5.91   |
| GSP-HR-128   | 56.09 | 5.62     | 73.93  | 0.89   | 4.73   |
| GAP-Patch-128| 91.00 | 9.11     | 106.07 | 5.89   | 3.22   |
| GSP-224      | 108.19| 10.83    | 134.01 | 0.0    | 10.83  |
| GSP-HR-224   | 65.81 | 6.59     | 87.69  | 1.89   | 4.70   |
| GSP-Full     | 161.63| 16.18    | 178.11 | 4.92   | 11.26  |

from the tables reporting other datasets due to their misleading numerical value when the target count in the test sample is close to zero.

**Wheat-Spike dataset:** This dataset [16] is a comparatively challenging one for object counting because of the irregular placement or collocation of wheat spikes. Out of 10 training samples, we use 8 for training and 2 for validation. Like COWC, the Wheat-Spike dataset is an ideal case study for GSP because of the low number of high-resolution training samples. Since the images are high-resolution and sub-regions inside a single image vary quite a bit in terms of brightness, perspective, and variable object shape resulting from natural morphology and wind motion, there are many features inside a single image that any suitable architecture should exploit without memorization or overfitting.

Table 4 reports the performance of different GSP-N models on this dataset. The error for GSP model trained with full-resolution images is quite high – 161.63, about 16% MAE compared to the average count of 1000. GSP-128 provides the best performance with MAE of 61.81 (6% of average count), where GSP-128 equipped with HR performs even better. Figure 5 shows cropped samples, their superimposed activation maps from GSP-Full model (middle), and GSP-128 (right). The GSP-128 model is able to identify salient regions in the image well, but for GSP-Full models, it tries to blindly memorize the count from only eight high-resolution images, which is clearly evident from the very uniform heatmap distribution all over the image regardless of foreground and background. Similar to car counting, we observed large per-patch counting errors for the GAP-Patch-128 model with a per-patch RMAE of 25.73%.

**Acknowledgment**

This research was undertaken thanks in part to funding from the Canada First Research Excellence Fund and the Natural Sciences and Engineering Research Council (NSERC) of Canada. We also thank Steve Shirtliffe, Anique Josuttes, and the UofS field crew for providing the Wheat-Spike dataset.

**5 Conclusions and Future work**

In this paper, we introduce the global sum pooling operation as a way to train one-look counting models without overfitting on datasets containing few high-resolution images. With detailed experimental results on several datasets, we show that our GSP model trained with smaller input samples
provides more accurate counting results than existing approaches. At the same time, these models learn to localize objects better. Based on these observations, as future work we plan to evaluate GSP models on other counting datasets that include larger numbers of training samples. Although we have only addressed object counting in this study, we believe that GSP should be investigated for other computer vision tasks, such as classification or object detection, where the scaling property of GSP may be able to utilize the features of image sub-regions over multiple spatial scales better than models that employ GAP or FC layers.

References

[1] S. Aich, A. Josuttes, I. Ovsyannikov, K. Strueby, I. Ahmed, H. S. Duddu, C. Pozniak, S. Shirtliffe, and I. Stavness. Deepwheat: Estimating phenotypic traits from crop images with deep learning. In 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pages 323–332, March 2018.

[2] S. Aich and I. Stavness. Leaf counting with deep convolutional and deconvolutional networks. In 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), pages 2080–2089, Oct 2017.

[3] S. Aich and I. Stavness. Improving object counting with heatmap regulation. CoRR, abs/1803.05494, 2018.

[4] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman. Interactive object counting. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, Computer Vision – ECCV 2014, pages 504–518, Cham, 2014. Springer International Publishing.

[5] L. Boominathan, S. S. S. Kruthiventi, and R. V. Babu. Crowdnet: A deep convolutional network for dense crowd counting. In Proceedings of the 2016 ACM on Multimedia Conference, MM ’16, pages 640–644, New York, NY, USA, 2016. ACM.

[6] H. Caesar, J. R. R. Uijlings, and V. Ferrari. Coco-stuff: Thing and stuff classes in context. CoRR, abs/1612.03716, 2016.

[7] J. P. Cohen, G. Boucher, C. A. Glastonbury, H. Z. Lo, and Y. Bengio. Count-ception: Counting by fully convolutional redundant counting. In 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), pages 18–26, Oct 2017.

[8] A. Dobrescu, M. V. Giuffrida, and S. A. Tsaftaris. Leveraging multiple datasets for deep leaf counting. In 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), pages 2072–2079, Oct 2017.

[9] L. Fiaschi, U. Koethe, R. Nair, and F. A. Hamprecht. Learning to count with regression forest and structured labels. In Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), pages 2685–2688, Nov 2012.

[10] J. H. Friedman. Greedy function approximation: A gradient boosting machine. Ann. Statist., 29(5):1189–1232, 10 2001.

[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In 2017 IEEE International Conference on Computer Vision (ICCV), pages 2980–2988, Oct 2017.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778, June 2016.

[13] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural Comput., 9(8):1735–1780, Nov. 1997.

[14] M. R. Hsieh, Y. L. Lin, and W. H. Hsu. Drone-based object counting by spatially regularized regional proposal network. In 2017 IEEE International Conference on Computer Vision (ICCV), pages 4165–4173, Oct 2017.

[15] G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2261–2269, July 2017.

[16] A. Josuttes, S. Aich, I. Stavness, C. Pozniak, and S. Shirtliffe. Utilizing deep learning to predict the number of spikes in wheat (triticeum aestivum). In Phenome 2018 Posters, 2018.

[17] I. Krasin, T. Duerig, N. Alldrin, V. Ferrari, S. Abu-El-Haija, A. Kuznetsova, H. Rom, J. Uijlings, S. Popov, S. Kamali, M. Malloci, J. Pont-Tuset, A. Veit, S. Belongie, V. Gomes, A. Gupta, C. Sun, G. Chechik, D. Cai, Z. Feng, D. Narayanan, and K. Murphy. Openimages: A public dataset for large-scale multi-label and multi-class image classification. Dataset available from https://storage.googleapis.com/openimages/web/index.html, 2017.
[18] V. Lempitsky and A. Zisserman. Learning to count objects in images. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems* 23, pages 1324–1332, Curran Associates, Inc., 2010.

[19] M. Lin, Q. Chen, and S. Yan. Network in network. *CoRR*, abs/1312.4400, 2013.

[20] 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 D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.

[21] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, June 2015.

[22] 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 B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, pages 785–800, Cham, 2016. Springer International Publishing.

[23] D. Oñoro-Rubio and R. J. López-Sastre. Towards perspective-free object counting with deep learning. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, pages 615–629, Cham, 2016. Springer International Publishing.

[24] M. P. Pound, J. A. Atkinson, D. M. Wells, T. P. Pridmore, and A. P. French. Deep learning for multi-task plant phenotyping. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 2055–2063, Oct 2017.

[25] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, June 2016.

[26] M. Ren and R. S. Zemel. End-to-end instance segmentation with recurrent attention. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 293–301, July 2017.

[27] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99, Curran Associates, Inc., 2015.

[28] B. Romera-Paredes and P. H. S. Torr. Recurrent instance segmentation. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, pages 312–329, Cham, 2016. Springer International Publishing.

[29] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[30] D. B. Sam, S. Surya, and R. V. Babu. Switching convolutional neural network for crowd counting. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4031–4039, July 2017.

[31] C. Shang, H. Ai, and B. Bai. End-to-end crowd counting via joint learning local and global count. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 1215–1219, Sept 2016.

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[33] V. A. Sindagi and V. M. Patel. A survey of recent advances in cnn-based single image crowd counting and density estimation. *Pattern Recognition Letters*, 2017.

[34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, June 2015.

[35] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, June 2016.

[36] J. R. Ubbens and I. Stavness. Deep plant phenomics: A deep learning platform for complex plant phenotyping tasks. *Frontiers in Plant Science*, 8:1190, 2017.

[37] E. Walach and L. Wolf. Learning to count with cnn boosting. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, pages 660–676, Cham, 2016. Springer International Publishing.
[38] C. Wang, H. Zhang, L. Yang, S. Liu, and X. Cao. Deep people counting in extremely dense
crowds. In Proceedings of the 23rd ACM International Conference on Multimedia, MM ’15,
pages 1299–1302, New York, NY, USA, 2015. ACM.

[39] 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, 0(0):1–10, 2016.

[40] Y. Xie, F. Xing, H. Su, and L. Yang. Beyond classification: Structured regression for robust cell
detection using convolutional neural network. In N. Navab, J. Hornegger, W. M. Wells, and
A. F. Frangi, editors, Medical Image Computing and Computer-Assisted Intervention – MICCAI
2015, pages 358–365, Cham, 2015. Springer International Publishing.

[41] C. Zhang, H. Li, X. Wang, and X. Yang. Cross-scene crowd counting via deep convolutional
neural networks. In 2015 IEEE Conference on Computer Vision and Pattern Recognition
(CVPR), pages 833–841, June 2015.

[42] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma. Single-image crowd counting via multi-column
convolutional neural network. In 2016 IEEE Conference on Computer Vision and Pattern
Recognition (CVPR), pages 589–597, June 2016.

[43] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for dis-
criminative localization. In 2016 IEEE Conference on Computer Vision and Pattern Recognition
(CVPR), pages 2921–2929, June 2016.