Counting Objects in Images using DeepLearning: Methods and Current Challenges

Adriano D'Alessandro (acdaless@sfu.ca)
Simon Fraser University
Ali Mahdavi-Amiri
Simon Fraser University
Ghassan Hamarneh
Simon Fraser University

Research Article

Keywords: Survey, Deep Learning, Object Counting, Annotation Burden, Visual Grounding

Posted Date: June 2nd, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2986682/v1

Additional Declarations: No competing interests reported.

Counting Objects in Images using Deep Learning: Methods and Current Challenges

Adriano D'Alessandro[1*], Ali Mahdavi-Amiri[1] and Ghassan Hamarneh[1]

[1]Computing Science, SFU, 8888 University Dr, Burnaby, V5A 1S6, BC, Canada.

*Corresponding author(s). E-mail(s): acdaless;
Contributing authors: amahdavi; hamarneh@sfu.ca;

Abstract

Object counting is an important computer vision application and research topic, which typically involves enumerating the number of objects in an image. Methodologies spanning a broad set of strategies have been proposed for solving object counting problems. These methods have seen an increase in relevance with the recent emergence of several highly successful deep learning techniques, which have led to significant performance improvements on a growing number of annotated counting benchmark datasets. However, despite the recent advancements in deep learning and computer vision, object counting remains a challenging problem with several open research directions. Datasets often contain objects that are highly occluded and which occur across a range of scales and perspectives. Further, popular annotation strategies, like density map annotations, suffer from annotator noise and inconsistency, which creates a performance bottleneck. These annotation strategies also have a high annotation burden, which leads to datasets that are very small when compared to common benchmark datasets in domains like image classification. Given both the significant progress and continued challenges of object counting, this task continues to be an interesting and ongoing research problem. This overview explores the historical context of object counting methods, the fundamental methodologies driving progress, the state of the art methods, and the significant open problems. In particular, we focus on recent trends that attempt to alleviate the problem of the annotation burden for object counting problems.

Keywords: Survey, Deep Learning, Object Counting, Annotation Burden, Visual Grounding

1 Introduction

Object counting is an important computer vision application and research topic, which typically involves identifying every object within an image and summarizing the number of those objects. Methodologies spanning a broad set of strategies have been proposed for solving object counting problems. These methods have seen an increase in relevance with the recent emergence of several highly successful deep learning techniques, which have led to significant performance improvements on a growing number of annotated counting benchmark datasets. The demand for high performance object counting methods is motivated by a diverse set of practical applications. For example, solutions to crowd counting [1, 5–14] problems can be applied to large event management where they provide a critical safety check by monitoring pedestrian density. Likewise, vehicle counting [4, 15, 16] provides civil engineers and city planners with tools for analyzing traffic patterns and street parking patterns. Object counting has also been applied to ecological surveying [17], where the population levels of animals like Penguins [2] and Steller Sea Lions [18] have been measured from images. Other applications include yield assessment and growth forecasting in agriculture via counting plant organs (such as fruits and leaves) [3, 19–21], and biomedical applications such as cell counting [22–25].

However, despite the recent advancements in deep learning and computer vision, object counting remains a challenging problem with several open

Fig. 1: Examples of objects from a diverse set of object counting task. From left-to-right (both rows), the objects of interest are pedestrians [1], penguins [2], apples [3], and parked vehicles [4]

research directions. Object counting datasets often contain objects that are highly occluded and which occur across a range of scales and perspectives. Further, the popular annotation strategies, like density map annotations, suffer from significant annotator noise and inconsistency, which creates a performance bottleneck. These annotation strategies also have a very high annotation burden, which leads to datasets that are extremely small when compared to common benchmark datasets in domains like image classification.

Given the demand for high performance object counting method driven by a broad range of applications, and given the challenging problems related to object counting, we seek to provide an overview of the field which helps better clarify these points. This overview explores the historical context of object counting methods, the fundamental methodologies driving progress, the state of the art methods, and the significant open problems. Further, we provide a specific focus on recent trends that attempt to alleviate the problem of the annotation burden for object counting problems. These methods can be broadly categorized as object counting with limited data, which encompasses several distinct methodologies.

2 Main Contribution

While there are several survey papers reviewing counting methods in the literature [26–30], these reviews exclusively focus on crowd counting where the object of interest is pedestrians. Our survey does not limit the scope in this way, and we opt to explore object counting across several object classes. The surveys by the authors [30] and [29] are the most up-to-date review of crowd counting methodologies. Their survey focuses on crowd counting methods, and only very briefly explore self-, semi-, and weakly-supervised learning. We explore object counting with limited data in greater detail, and focus on new trends such as few-shot object counting. Overall, we present a survey that offers a more broad look at object counting problems and which seeks to identify the major challenges in the field and the state of the art for counting with limited data.

2.1 Search Strategy

To select papers for this survey, we searched through the databases of several important machine learning and computer vision conferences. Specifically, we searched the databases for CVPR, ECCV, ICCV, ICML, ICLR, and NeurIPS using the following search query: `(count*|crowd)`. We only searched for publication records from between the years 2015 and 2022. We also searched Google Scholar using the search query: `(object counting|crowd counting|counting dataset)`. Beyond this, we collected a set of seminal papers and papers that introduce datasets from within the relevant literature returned by our search. These additional papers are occasionally selected from outside of the set of conferences highlighted above, and are selected based on the criteria that they show up in more than

one publication or appear to be highly influential. To ensure quality and focus, we exclude non peer-reviewed pre-prints found on arXiv, multi-view object counting papers, and action counting papers. We focus exclusively on single image 2D object counting problems. For surveys that explore pre-deep learning era methods, we suggest the 2008 survey by Zhan et al. [31] on crowd analysis. We also note that object localization, detection, and semantic segmentation are tasks related to object counting. We discuss the major differences between these tasks in section 4 and we explore why solutions to these problems are a poor fit for the object counting task in section 5. However, since solutions to these related problems are not strictly object counting methods, we exclude them from this report. We refer an interested reader to review the following recent surveys on object detection [32, 33], object localization [34], and semantic segmentation [35].

2.2 Limitations and Coverage

While there exists a diverse set of object counting benchmark datasets spanning several object classes, a significant proportion of the literature has focused exclusively on crowd counting problems. While several papers do evaluate on multiple types of object counting datasets, it is simply not possible to give equal weighting to all types of object counting applications while fairly surveying the literature. Further, almost all surveyed methods focus on humans, animals, or inanimate objects organized within natural scenes. A smaller proportion do focus on other settings, such as cell counting and leaf counting. However, the majority of papers selected for this review will be biased towards humans in natural scenes.

3 Object Counting Applications

3.1 Medical Imaging

The object counting task has been periodically explored within the medical image analysis research literature. Typically, object counting methodologies are utilized for either tracking disease progression, or studying the underlying mechanism of a disease. For example, detecting a change in the number of unique Gad-enhancing lesions in a T1-weighted MRI scan is a useful metric for tracking the progression of Multiple sclerosis [41–43]. Cell counting has also emerged as an important and common biomedical application [22–25, 39, 44, 45]. For example, there exists benchmark datasets for adipocyte cells counting [25], bone marrow cell counting [39], and breast cancer tissue cell counting [44] which are detailed in table 1. Synthetic cell counting datasets have also been developed to aid in cell counting research [45]. Additionally, automatic Acne vulgaris grading and lesion counting is an important application for practitioners. Recent works have established the ACNE04 dataset [40], also detailed in table 1. The number of lesions and the global grade are related

Dataset	Object Class	Annotation	Year	Size	Avg. Count
UCSD [6]	Person	Dot Maps	2008	2,000	25
Mall [7]	Person	Dot Maps	2012	2,000	31
UCF_CC_50 [10]	Person	Dot Maps	2013	50	1,279
WorldExpo'10 [8]	Person	Dot Maps	2016	3,980	50
ShanghaiTech Part A [1]	Person	Dot Maps	2016	482	501
ShanghaiTech Part B [1]	Person	Dot Maps	2016	716	123
CityUHK-X [36]	Person	Dot Maps	2017	3,191	33
UCF-QNRF [11]	Person	Dot Maps	2018	1,535	815
GCC (Synthetic) [12]	Person	Dot Maps	2019	15,212	501
JHU-CROWD++ [37]	Person	Dot Maps	2019	4,372	346
Crowd_Surv [9]	Person	Dot Maps	2019	13,945	35
NWPU-Crowd [5]	Person	Dot Maps	2020	5,109	418
TRANCOS [16]	Vehicle	Dot Maps	2015	1,641	36
PKLot [38]	Vehicle	Bounding Box	2015	12,417	57
COWC (Aerial) [15]	Vehicle	Dot Maps	2016	-	-
CARPK (Aerial) [4]	Vehicle	Bounding Box	2017	-	-
Penguins [2]	Penguins	Dot Maps	2016	80,095	7
CVPPP [20]	Leaves	Dot Maps	2016	810	-
Maize Tassel Counting [19]	Maize Tassels	Dot Maps	2017	361	-
Minneapple [3]	Apples	Bounding Box	2020	1,001	41
Global Wheat Head [21]	Wheat Heads	Bounding Box	2021	6,422	43
Adipocyte Cells [25]	Adipocyte Cells	Dot Maps	2013	200	-
MBM Cells [39]	Bone Marrow Cells	Dot Maps	2015	44	-
Dublin Cell Counting [24]	Any Cells	Dot Maps	2018	177	34
ACNE04 [40]	Acne Lesions	Bounding Box	2019	1,457	13

Table 1: Non-exhaustive list of object counting datasets used throughout the literature. Here, *dot maps* are listed as an annotation type. However, in practice, *dot maps* are often converted into *density maps*. The two are variations of the same annotation output, and so we will typically refer only to *density maps* when talking about how these annotations are used. Dataset statistics are provided where they are made available.

quantities which can be used as a measure of acne severity. Given these examples, it is clear that object counting methodologies have a demonstrated value within the medical image analysis research literature.

3.2 Plant Image Analysis

Object counting within the plant image analysis literature is a common task, with two significant use cases. The first use case is yield forecasting, which involves assessing the number of some plant organ, such as fruits or leaves, and estimating the future yield of that plant. The MinneApple dataset [3] is an example benchmark for this task, which involves the task of estimating the number of apple fruit on a plant. Similarly, the Maize Tassels Counting dataset [19] and the Global Wheat Head Detection dataset [21] are common benchmarks for assessing the number of maize tassels and wheat heads in images respectively. Automatically counting the number of maize tassels and wheat heads provides an important metric for yield forecasting. The second use case of object counting in plant image analysis is plant phenotyping, which is the task of estimating the genotype of a plant from its observable characteristics, which includes quantifying the number of some plant organ.

Plant biologists and breeders often attempt to characterize the effects of new genotypes or breeding strategies at scale, and the rapid assessment of a plant phenotype significantly reduces the burden of this task. The CVPPP Leaf Counting dataset [20] is the most popular dataset for this task, and has existed as a challenge dataset for several years. These applications highlight the value of object counting methodologies in plant image analysis.

3.3 Crowd Counting

The vast majority of object counting methodologies have been developed for the crowd counting problem. This task specifically looks at assessing the number of humans in dense crowds within natural scene. The popularity of object counting methodologies specifically targeting the crowd counting problem is likely due to the large number of available benchmark datasets. There are 11 popular benchmark datasets, as outlined in figure 1, with a large range of object counts and resolutions. Included in this list is the UCSD dataset [6], the Mall dataset [7], the WorldExpo'10 dataset [8], the ShanghaiTech datasets [1], the Crowd_Surv dataset [9], the UCF_CC_50 dataset [10], the UCF-QNRF dataset [11], the GCC dataset [12], the JHU-CROWD++ dataset [37], and the NWPU-Crowd dataset [5]. As an application, crowd counting methodologies can be applied to several important task. Crowd analysis and safety involves assessing the number of people in the crowd at a large event, such as a concert, and determining whether the number of people in the space is within a safe threshold. Similarly, crowd counting methods can be used by civil engineers for pedestrian analysis and city planning around pedestrian traffic. This highlights that the crowd counting research space is an activate area with several motivating applications and several benchmark datasets for evaluation.

3.4 Traffic Analysis

Counting the number of vehicles is a common task that can provide civil engineers with important information necessary for city planning. For example, the TRANCOS dataset [16] is a popular benchmark dataset, which focuses on fixed-camera based vehicle counting and highway traffic analysis. The CARPK dataset is a large scale dataset [4] that focuses on drone-based counting of vehicles in parking lots. The COWC dataset [15] is a satellite imagery based vehicle counting dataset for assessing road side parking, business volume, etc. Given these applications, it becomes obvious that there are a wide variety of use cases where object counting methodologies can be applied to vehicle counting.

3.5 Ecological Surveying

Object counting methods provide significant value in ecological surveying, where methods for automatically assessing animal populations in images is an important tool for ecologists. The most popular benchmark dataset for ecological surveying is the Penguins dataset [2], which is a large dataset for

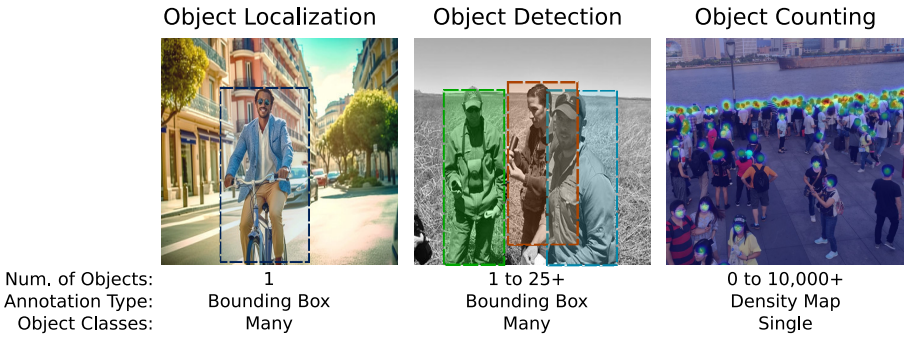


Fig. 2: There are many popular object recognition tasks. While object localization and object detection primarily focus on locating the image regions containing objects from several classes, object counting focuses on simply estimating the number of objects from a single class in dense scenes with. While object counting does not explicitly require localizing objects, it has been demonstrated that location based targets, such as density maps, improve performance. Rightmost image modified from [1]

counting penguin populations in the wild. Due to arctic locations being challenging to access, methods that allow for remote penguin counting are highly valued. Similarly, object counting has been explored for Seal population analysis [17, 18]. While the number of ecological surveying benchmark datasets is small, the Penguins dataset [2] is the largest commonly used object counting dataset available. These applications highlight the value of object counting methods for ecological surveying.

4 Object Recognition

4.1 Object Recognition Tasks

Deep learning methodologies are defined by a *dataset*, a *task*, and a *method*. A *task* is a well-defined formulation of some goal which is achieved by producing an output signal from an input signal. One important category of computer vision tasks are object recognition tasks, which describes a set of fundamental computer vision problems that involve detecting the presence of an object within an image. While there are innumerable ways to define object recognition tasks, there are several well defined object recognition tasks that have been highly explored within the computer vision literature. In this section, we will first discuss these important object recognition tasks to better define the research landscape. After, we will explore the details of the object counting task.

Object classification refers to the fundamental task of determining if some class of object exists within an image. More specifically, given an image:

$$x \in [\![0, 255]\!]^{\mathtt{H}\times\mathtt{W}\times\mathtt{C}}$$

and N object classes, the overall goal is to correctly predict one of N buckets for each image x based on the identity of the object within the image. This task has a long and rich history, with early applications including handwritten digit classification [46]. More recent examples include the ImageNet dataset [47], which involves classifying natural images into 1000 unique object categories.

Unlike object classification, which simply involves the categorization of whole images, the object localization task involves detecting which image regions specifically contain the object. This task specifically focuses on localizing a single object from a single object category within each image and involves estimating the image co-ordinates,

$$(w_{\texttt{min}}h_{\texttt{min}},\ w_{\texttt{min}}h_{\texttt{max}},\ w_{\texttt{max}}h_{\texttt{min}},\ w_{\texttt{max}}h_{\texttt{max}})$$

which tightly bound the object within each image. Here, $w_{\texttt{min}}$ is the minimum column index, $w_{\texttt{max}}$ is the maximum column index, $h_{\texttt{max}}$ is the maximum row index, and $h_{\texttt{min}}$ is the minimum row index which bound the object within the image with respect to the image origin. This box is commonly referred to as a bounding box. The most prominent example application for this task is the *ImageNet Large Scale Visual Recognition Challenge* [48], which established a localization challenge based on the ImageNet dataset. The left-most image in figure 2 gives an example of the object localization task.

While the object localization task involves estimating the coordinates of a single object from a single class within an image, the object detection task involves localizing one-or-more objects from one-or-more object classes within an image. The object detection task essentially emerges as a more challenging version of the object localization task, as it requires a method that can predict multiple bounding boxes across multiple classes given a single image. The most popular benchmark datasets for object detection problems are the PASCAL VOC dataset [49, 50] and the MS COCO [51]. The middle image in figure 2 provides an example of the object detection task. There are many additional well-defined object recognition tasks beyond those listed so far. Instance segmentation [51], object keypoint detection [52], and pose estimation [53] are just a few examples of additional tasks that fit within the category of object recognition tasks. However, object classification, object localization, and object detection are the most useful for juxtaposition against the object counting problem.

4.2 The Object Counting Task

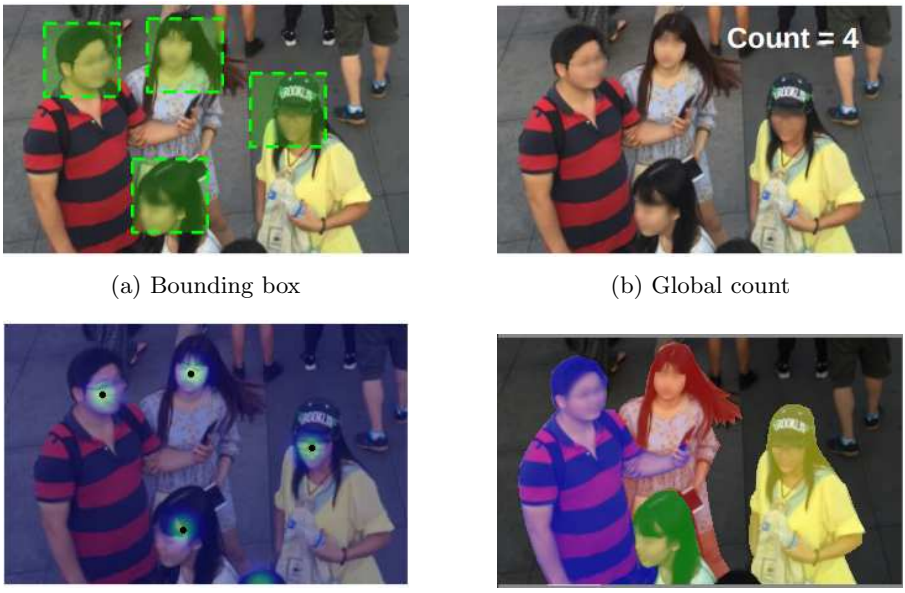
The object counting task is defined by the goal of quantifying the total number of objects visible within an image. This task almost always involves counting many instances of a single object class within a single image. Similar to the object classification task above, the object counting tasks does not explicitly require the detection of object instances within the image. It simply requires the estimation of a single value representing the correct number of objects. However, in practice, some intermediary detection step is incredibly beneficial

for performing well on the task. Given that detection is often a frequent component for object counting problems, it then makes sense to compare it directly to the object detection task. Despite the tasks having different high-level goals, they tend to share some similarities. However, speaking generally, object detection problems tend to focus on detecting a small number of objects across a wide range of object classes using bounding boxes as the image annotation. This is not necessarily true for all detection problems, but does characterize object detection in natural images. For example, a popular object detection benchmark dataset known as MS COCO [51] has an average of 7.3 objects per image and spans across 80 object classes. In contrast to this, object counting problems tend to involve counting a large number of objects across a single class. As a similar motivating example, the popular object counting dataset known as ShanghaiTech part B [1] contains an average of 123 objects per image. While these object recognition tasks share many similarities, each problem faces a distinct set of circumstances that require different methodologies. For example, if predicting bounding boxes for each object were to be used as an intermediary detection step for an object counting methodology, it is not guaranteed that bounding boxes would scale well to the number of objects commonly seen in object counting problems. In practice, the object counting task typically involves a different form of annotation, all together.

5 Dataset Annotation Strategies for Object Counting

5.1 Overview

While the goal of object counting is to simply return the total number of objects as a single value, there are several possible ways to acquire the object count through intermediary targets. Four possible annotation strategies are presented in Figure 3. Bounding boxes serve as one possible training target for counting problems. This strategy involves training a neural network in the same manner as with object detection problems, where the network performs regression over the bounding box coordinates for each object within the image. Then, during inference, the total number of bounding boxes generated by the network would simply correspond to the object count. The benefit of using bounding box regression for counting problems is that it explicitly frames counting as a localization problem, which grounds the object count by correlating it directly with the local object features in each image. However, bounding box regression tends to under perform on object counting problems. This has been demonstrated empirically [54], with the intuition that bounding box regression introduces ambiguity and noise when objects are highly occluded and cluttered in dense groups. Additionally, many object detection pipelines [55] filter multiple overlapping object proposals using non-maximum suppression (NMS). It is difficult to distinguish whether two object proposals refer to a single object or if they refer to two overlapping objects [56] and this



(a) Bounding box (b) Global count

(c) Density map (from dot map) (d) Instance segmentation

Fig. 3: Examples of different types of annotations that can be used for estimating the number of objects in an image. Images are modified from ShanghaiTech part B dataset [1]

is also aggravated by the wide range of object scales and crowd densities in object counting problems.

Global count labels are another object counting annotation type. These are the most straight forward training target, as training proceeds by simply using a neural network to predict the global count. However, these global count labels are not grounded within the image. A neural network has to learn a relationship between the true local object features and the global count without explicit guidance about the object identity. This can be problematic, as neural networks are known to incorrectly fit to patterns that spuriously correlate with the image label, especially when finding a function that fits those spurious patterns is simpler to learn than the true underlying function [57]. Given that object counting datasets are often rife with a large range of object scales and degrees of occlusion, this problem requires serious considerations.

Another potential object counting annotation strategy is instance segmentation. While this strategy is never used within the research literature, it still represents a possible avenue for acquiring object counts. This method would proceed by first training a model to perform instance segmentation, which is the task of individually segmenting each unique object. Then, during inference, each unique object captured by the output segmentation masks is summed to get the count. From a practical stand point, instance segmentation is a

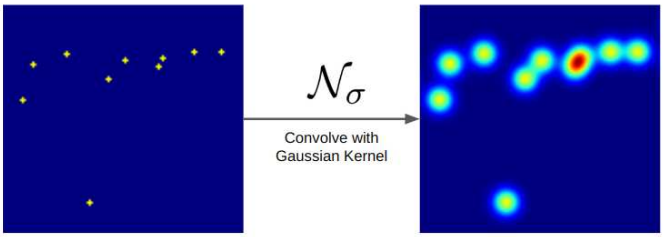


Fig. 4: Density maps annotations are produced by first having an annotator place a single dot at the location of each object, and then convolving a Gaussian kernel with the resulting dot map. This produces a location-based heat map, where larger values correspond to high object density.

very labour intensive annotation type to collect. Further, the task of instance segmentation is very challenging in different ways than the object counting problem, and spending network capacity on those challenges may not represent an ideal scenario.

The last annotation type we will consider are density map annotations. These are a special type of heatmap, where Gaussian density is placed over the location of every object in the image. This heatmap has the property that the total object count can be acquired by integrating the density map. Training proceeds by performing density map regression, and inference is performed by simply taking the summation of the density map. This strategy has the benefit of localizing object features in the image, while not being as restrictive and problematic as bounding box regression. We will now explore this annotation type in greater detail.

5.2 Density Map Annotations

Density map annotations, first proposed in 2010 by the authors of [58], are a useful optimization target for object counting methods because they representation a continuous spatial distribution of objects within an image. To produce a density map, first suppose that we have some image which we would like to annotate:

$$x \in [\![0, 255]\!]^{\mathtt{H}\times\mathtt{W}\times\mathtt{C}}$$

which contains N objects that can be realistically identified by an annotator. Here, H is the height, W is the width, and C is the number of channels. An annotator would produce a *dot map* M^{dot} with the following properties:

$$M^{dot} \in \{0, 1\}^{\mathtt{H}\times\mathtt{W}}$$

Where $M^{dot}_{ij} = 1$ for each point x_{ij} that represents the approximate centroid of each object within the image, and $M^{dot}_{ij} = 0$ everywhere else. In practice, objects tend to be highly occluded and the true centroid is not necessarily visible. Then, the centroid is either estimated or a dot is placed at the approximate

centroid of the visible portion of the object. A density map, defined as:

$$D^m \in \mathcal{R}_{0+}^{\mathtt{H}\times\mathtt{W}}$$

can then be generated by convolving an isotropic Gaussian kernel over M^{dot}, such that:

$$D^m = M^{dot} * \mathcal{N}_\sigma$$

where σ is the standard deviation of the Gaussian kernel, which determines the spread for each resulting Gaussian blob. This process is detailed in figure 4.

Given that the integral of each Gaussian blob is equal to 1, obtaining the final object count from a density map simply requires that we take the integral of the entire density map. In practice, this operation is performed by taking the sum over the entire density map:

$$y_{count} = \sum_{ij} D_{ij}^m$$

which provides the final count, y_{count}. As an optimization target, density maps have two important hyperparameters which need to be selected. First, the standard deviation of the Gaussian kernel, σ. This is often empirically determined, and assignments are dataset specific. Second, the resolution of the density map may also be modified. In practice, the resolution of the density map is usually set as $\frac{H}{8} \times \frac{W}{8}$, relative to the width and height of the image x. This may be done to either simplify the optimization target or reduce the computational resources necessary to output the density map. While this process for generating density maps is typical of many methodologies, there are many that opt to modify the density map generation procedure. For example, geometry-adaptive Gaussian kernels may be applied to treat dense regions differently than sparse regions [59].

6 Challenges of Object Counting

Object counting problems seek to determine the number of objects in an image under a variety of conditions related to the dynamics between objects and the environment. These dynamics can lead to highly complex scenes that present significant challenges for object counting methodologies. Further, object counting problems require human annotators to provide accurate information, which becomes difficult when the scene are complex or even ambiguous. In the remainder of this section, we will explore these concepts further, and provide an overview the major challenges of object counting

6.1 Object Variance

6.1.1 Scale and Perspective

Perspective in imaging is related to the process of projecting a 3D scene onto the 2D sensor of a camera. This relationship is governed by the geometry

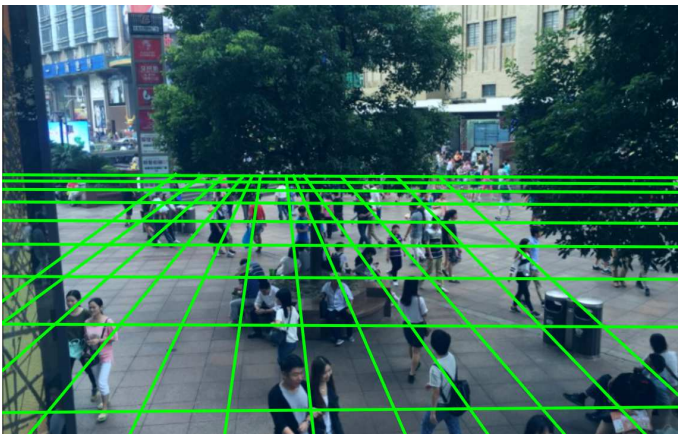


Fig. 5: Images of natural scenes have a perspective, which impacts the scale of objects within the image. The leads to a situation where objects are either close to the camera and highly resolved, or very far away from the camera and poorly resolved. Grids represent an artistic approximation of the perspective in the scene. Images are modified from ShanghaiTech part B dataset [1]

of scene elements and their distance from the camera. Within the context of object counting problem, perspective defines the distribution of objects within a scene and can lead to a large variance in object scale. Figure 5 highlights an example of this problem, where objects closest to the camera appear larger and are better resolved and objects furthest from the camera appear smaller and are poorly resolved. This emerges as a significant problem for object counting methodologies, as a model must learn to extract features across scales, and must learn how to disentangle individual objects in dense crowds when those objects are small and poorly resolved. As will be discussed in later sections, many researchers have proposed multiple solutions that attempt to alleviate the problem of scale and perspective within images. However, it still remains an open and challenging problem. Further, while this is problem exists for many object counting problems in natural images, perspective issues will not necessarily emerge as a problem for all object counting problems.

6.1.2 Inter-Object Variance

While scale variance emerges as a natural property of perspective within complex scenes, inter-object variance emerges due to a distinct property of an object class. For example, consider that humans may exhibit diversity across a wide set of traits, including height, age, gender, clothing choice, etc. Or, consider that vehicles may come in a wide variety of colors, sizes, and shapes. This creates an additional challenge for object counting methodologies, which may need to capture this variance across object classes. This may be exasperated by several factors, including an imbalance in the distribution of object traits within some dataset. When considered alongside the problem of perspective

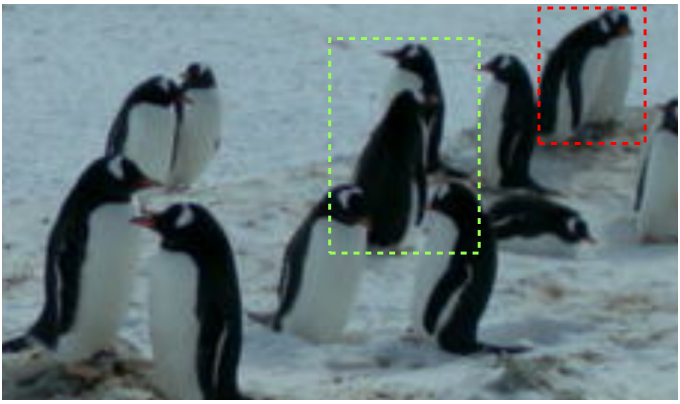


Fig. 6: There are two penguins in the green box, with one penguin partially occluding the other in a vertical arrangement, potentially appearing as a single long penguin. There are also two penguins in the red box, with one penguin almost entirely occluding the other, with only parts of the penguin's stomach and feet being visible. Images modified from [2]

based variance, the complexity of object counting problems becomes clear. Counting objects requires methods to learn complex functions that model the potentially large variance within an object class.

6.2 Occlusion

Occlusion refers to a problem in images where parts of objects are covered by other objects or scene elements, such that those portions of the object are no longer visible within the image. This reduces the information available when attempting to resolve and detect that object for the purposes of counting, and becomes a significant challenge for object counting methodologies. In particular, it requires that an object counting model learn a diverse set of object features and still recognize the object when some of those features are not present. Figure 6 provides examples of object occlusion within an image. Beyond the problems created by the removal of object features, when objects are occluded by other objects, the two objects become more difficult to resolve from one another. This acts a second complicating factor. Finding ways to delineate objects under the conditions of occlusion is a significant open problem that exists across object recognition tasks more broadly.

6.3 Annotation Burden

In sections 5 we discussed the details of density maps as a form of object counting annotation. These annotations require that an annotator place a point at the approximate location of each object within an image. Cholakkal et al. [60] conducted a study to measure the speed at which annotators could produce a density map for images taken from the MS COCO dataset [51]. They

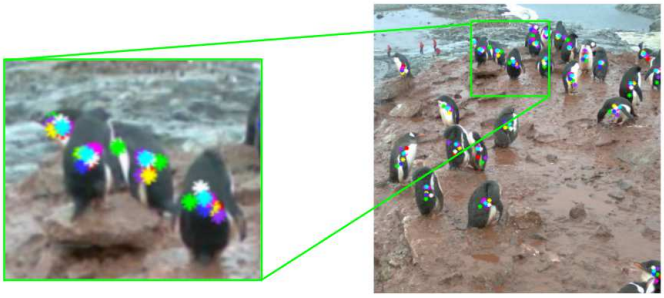


Fig. 7: Object dot assignments made by of several annotators for several objects, where each color indicates an individual annotators contribution. There is significant inconsistencies between where different annotators place dots. Further, annotators may not agree on whether an object exists at a location at all. Here, the *green* annotator places a single dot where no other annotator has selected one. Images modified from [2]

determined that it takes an annotator 1.1 seconds per point while annotating an image. However, the MS COCO dataset contains significantly fewer objects per image on average than the majority of common object counting datasets, and this measure of annotator speed may not hold for scenes with dense crowds, large variance in object scale, and heavy occlusion. However, using this measure as a baseline, we can estimate the annotation burden for the ShanghaiTech part B dataset. This dataset contains 716 images and 88,488 annotated individual objects. Using an annotator speed of 1.1 seconds, it can be estimated that it would take approximately 27 hours for an annotator to annotate every image. Cholakkal et al. [60] also identified that object class labels could be collected at a rate of roughly 1.1 seconds to 1.4 seconds per image. In the time it takes to annotate the 716 images in the ShanghaiTech part B data, one could also annotate nearly 88,488 images with object classes.

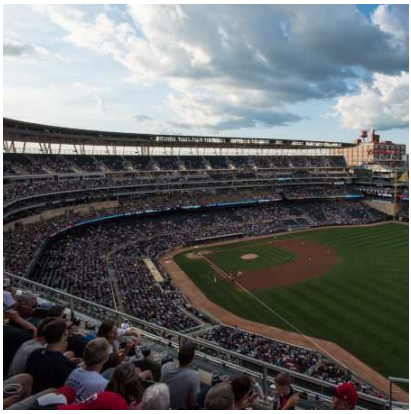
6.4 Annotation Imprecision

When collecting density maps, annotators are typically expected to place a dot at a consistent location, such as the approximate centroid of each object or the head of a person. However, this introduces a few significant problems. First, an annotator must make a decision on how to select the location of an object when parts of the object are occluded. Second, annotators may not be able to correctly select that location consistently between objects. And third, different annotators may make different types of errors when annotating the objects within a scene. Figure 7 shows an example of this problem. Different annotators select slightly different points as the centroid of each penguin, leading to significant inter-annotator variance. This can potentially lead to a performance bottleneck, where a neural network expends capacity fitting the annotator noise.

(a) Person in an advertisement

(b) Reflection in the mirror

(c) Occluded but implied by context



(d) Partial view of scene

Fig. 8: Ambiguity can emerge in counting problems from various sources. What do we consider to be a distinct object, rather than a representation of that object? Image (a) is modified from ShanghaiTech part B dataset [1]

6.5 Ambiguity

Ambiguity emerges due to a lack of precision in either the problem definition or the object category under consideration. For example, in Figure 8, we provide four examples of images where the total object count is ambiguous. In Figure 8a, a person is depicted within an advertisement. However, this is a depiction of a person, rather than the actual person. In Figure 8b, a person is reflected in a mirror. This appears as two people within a 2D image when the context of the mirror is not considered. This type of ambiguity poses a problem for object counting methods, as the source of ambiguity may not be present in the training set, or the annotator may be unclear on how to correctly annotate the image. Thus, it becomes difficult to represent the object

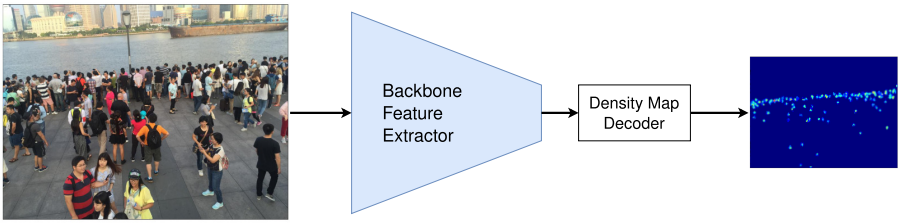


Fig. 9: The majority of object counting papers follow the same basic recipe for their architectural design. These networks are comprised of some feature extractor, usually some version of VGG16 [61] or ResNet50 [62]. They then have an additional parameterized function which decodes the image features to generate the final density map. Figure uses modified image from ShanghaiTech part B dataset [1]

category context necessary to determine the outcome in the presence of these ambiguous examples.

Further, ambiguity can be introduced when the downstream problem is not concisely defined. For example, if our goal is to estimate the number of penguins in an area, then we must make decisions about whether or not to attempt to estimate the presence of penguins within the area which are not present within the image. In Figure 8c, a person is sufficiently occluded behind another person, such that they do not appear in the image. However, their presence is implied by the context of the image, as people are lined up in a formation. In Figure 8d, only a partial view of a crowd is available. Counting only the visible crowd does not provide the total size of the crowd within the stadium. Counting the number of *visible* objects within the image provides an underestimate of the total number of objects within the image.

7 Architectures

Deep learning methodologies for object counting typically utilize a common neural network architecture formulation, as seen in figure 9. This formulation involves a backbone feature extraction neural network, which is responsible for extracting useful features for solving the object counting problem. These features are then decoded by a separate neural network, which is specifically designed to be relevant to the type of object counting annotation being utilized. One commonly used feature decoder is CSRNet [59], which is detailed in figure 11. This decoder passes the image features through several dilated convolutional layers before producing a density map estimate. The goal of this method is to use dilated convolutions to learn long range dependencies within an image, as crowds may have both local and long range properties that are important for summarizing the count.

VGG16 [61], detailed in figure 10, is by far the most common backbone feature extractor used for object counting problems. Figure 12 highlights this trend, with VGG type architectures appearing more frequently in the reviewed

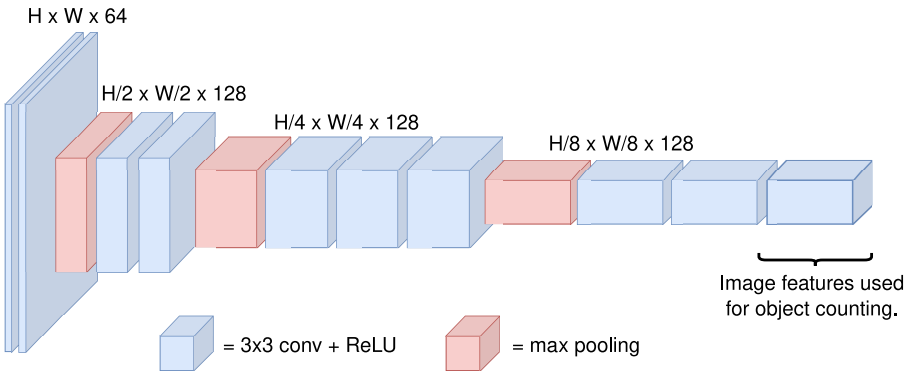


Fig. 10: VGG16 [61] is a common backbone feature extractor used in object counting architectures. VGG16 downsamples the image using consecutive convolutional layers followed by max pooling layers. The features are commonly extracted from the point where the resolution of the features is 1/8th of the original image.

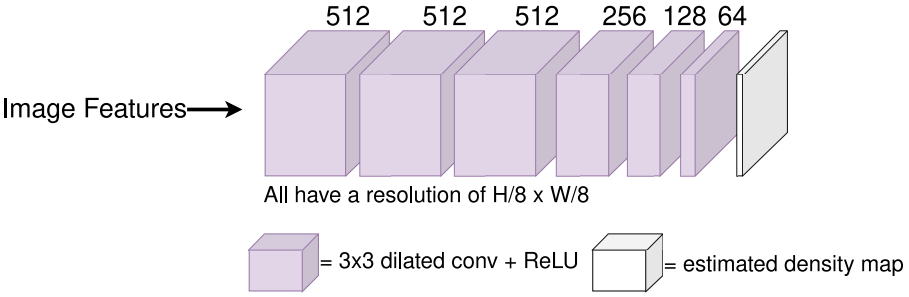


Fig. 11: The CSRNet decoder [59] is an example of a popular density map decoder used for object counting problems. It uses a stack of six dilated convolutional layers to learn long range dependencies within the image. Image reproduced from the text of [59].

papers than all other backbone feature extractors combined. First proposed by Karen Simonyan and Andrew Zisserman at ICLR 2015 [61], VGG16 is a deep convolutional neural network comprised of two distinct types of layers. The convolution layers process image features using parameterized kernels with a resolution of 3x3 before passing the new features through a ReLU activation function to introduce non-linearity. Image features are zero padded to maintain the same resolution between convolutional layers. Down sampling is performed by 2x2 max pooling layers, which pass forward only the maximum value within a 2x2 block. Convolution layers and max pooling layers are applied in reptition until the desired depth is achieved. The purpose of this process is to create a network which is deep, with the goal of achieving rich features representing the input image.

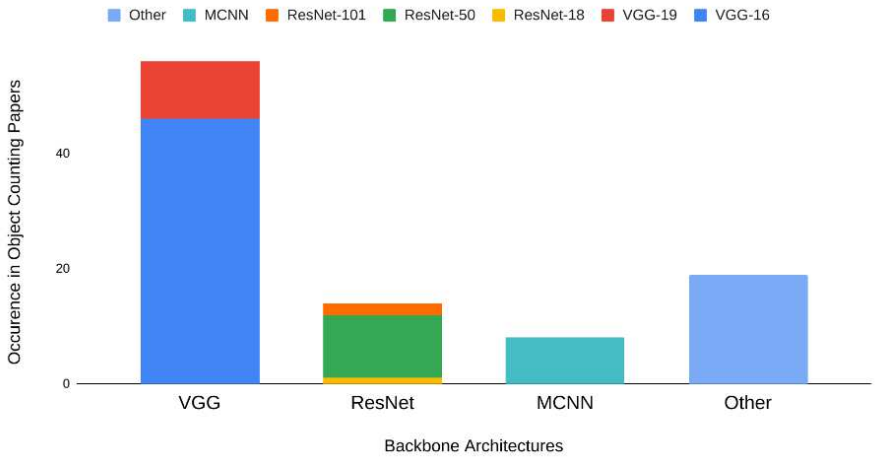


Fig. 12: Among all of the object counting papers summarized in this review, the vast majority of papers use a variant of VGG to perform feature extraction. ResNet and MCNN are also common. All other papers use a unique backbone architecture that can't be neatly categorized, and these tend to be the oldest papers.

In 2017, the seminal work on transformers fundamentally changed the field of natural language processing [63]. Transformer-like architectures are incredibly successful, and scale well with data size and architecture size. In 2021, the benefits of transformer-like architectures were brought to computer vision problems with the introduction of the Vision Transformer (ViT) [64]. These architectures differ from the convolutional networks described above, in that operate on patches rather than whole images, and they do not contain any inductive biases related to local neighborhood structure. The introduced ViT method introduced a strategy for operating on image patches, as well as feature patches generated from a traditional convolutional network. The latter attempted to merge the benefits of both architectures. However, transformers have seen limited application in object counting problems. Recently, the authors of [65] proposed the seminal work on the successful application of transformers to crowd counting problems, representing an emerging research direction.

There exists a wide variety of approaches to object counting architecture selection, ranging from neural architecture search [66] to simply using established off-the-shelf architectures like VGG16.

8 Evaluation Methods

8.1 Metrics

Object counting methods are typically evaluated by approximating their generalization error using a test set and two popular metrics. The first metric is mean absolute error, defined as:

$$E_{\texttt{MAE}} = \frac{1}{N}\sum_{i}^{N}|y_{gt}^{(i)} - \tilde{y}_{pred}^{(i)}|$$

where $y_{gt}^{(i)}$ is the ground truth label for the i-th example, $\tilde{y}_{pred}^{(i)}$ is the prediction made by the method on the i-th example, and N is the number of examples in the test set. This metric measures the average absolute difference in the number of objects predicted by the method and the true number of objects. The second metric is the root mean squared error, defined as:

$$E_{\texttt{RMSE}} = \sqrt{\frac{1}{N}\sum_{i}^{N}(y_{gt}^{(i)} - \tilde{y}_{pred}^{(i)})^2}.$$

This metric is typically used due to the fact that it is more sensitive to outliers given that the error scales quadratically with the difference between the prediction and ground true rather than linearly. This metric is important, as counting datasets are often highly imbalanced and it can help diagnosis problems related to poor generalization in regions with fewer examples.

Additionally, there are other niche metrics that are occasionally used to evaluate object counting methods. For example, the quality of the density map can be evaluated using metrics such as PSRN and SSIM. These metrics essentially measure how close the predicted density maps are to the ground truth density map on a per-pixel basis. As another example, the TRANCOS [16] dataset introduces an evaluation metric known as the Grid Average Mean Absolute Error (GAME). This metric is defined as follows:

$$\texttt{GAME}_{\texttt{L}} = \frac{1}{N}\sum_{i}^{N}\sum_{l}^{4^{\texttt{L}}}|y_i^l - \tilde{y}_i^l|$$

where the $\texttt{GAME}_{\texttt{L}}$ metric splits the image into $4^{\texttt{L}}$ regions and calculates the mean absolute error for each region. Here, y_i^l is the ground truth count for the l-th region of the i-th, and $\tilde{y}_i^l$ is the method prediction for the l-th region of the i-th example. This metric essentially attempts to determine how well the method localizes the objects. Other metrics do not discern between contributions to the count attributed by true objects and by non-objects.

While the above metrics are more common among object counting papers, there exists other metrics that may be useful for evaluating object counting

methods but which are not frequently utilized. For example, the mean absolute percentage error is defined as:

$$E_{\texttt{MAPE}} = \frac{1}{N}\sum_{i}^{N} |\frac{y_{gt}^{(i)} - \tilde{y}_{pred}^{(i)}}{y_{gt}^{(i)}}|$$

This metric measures the absolute difference between the true and estimated count as a percentage of the true count. As an example, consider a method that outputs a prediction of 5 when the true count is 0, and also outputs 1005 when the true count is 1000. Both of these errors would be treated the exact same using the previous metrics. However, mean absolute percentage error would penalize the former significantly more than the latter. We may prefer a metric which is more forgiving of small relative errors for larger total counts given that we may consider a model with a large relative error for small counts to be a critical failure.

8.2 Methods

The most common method for evaluating object counting methods is in-distribution domain generalization. [Ali: two methods in one sentence does not look good.] Typically, a dataset is collected from a single domain and randomly split into a training and test set. A model is then trained on one split and evaluated on the other. However, there are other ways to benchmark object counting methods. For example, the authors of [67] evaluated the robustness of their method to spatial annotator noise. They randomly shifted the location of each individual point in a dot map to simulate varying levels of annotator noise during training. The authors of [1] evaluated the domain generalization performance of their method by evaluating their method on various transfer learning tasks. They explored how well their method transferred from a source domain to a target domain. The authors of [68] considered inference time. Object counting methods may be deployed remotely on edge devices, and inference time performance may be necessary for reasonable performance on such devices.

9 Fully-Supervised Object Counting

9.1 Multi-Scale Feature Extraction

Object counting problems are complex, yet they are hindered by the available datasets being relatively small. A common research direction involves making custom neural network architecture design choices that better align the network design with the specifics of the problem. Typically, this involves careful consideration of the data and the task and using these intuitions to introduce expert knowledge about the problem, such that the network does not have to learn that knowledge from the dataset. One common approach, which shows up frequently within the literature, is the strategy of multi-scale feature fusion.

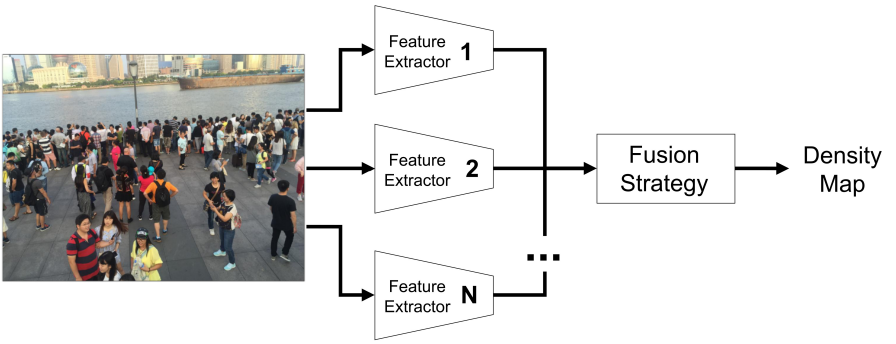


Fig. 13: An overview of the multi-scale feature extraction approach to object counting architecture design. Image features are extracted and aggregated using a fusion strategy. The feature extractors may be comprised of different architectures meant to manage different scales, or a single shared architecture with features sampled at different depths. Figure uses modified image from ShanghaiTech part B dataset [1]

Object scale is an incredibly important property of objects within counting datasets which requires careful consideration. A neural network must learn different representations of an object across the variety of scales. Previous research has approached this through multi-scale feature fusion. A high-level overview of these methods is presented in figure 13. These methods seek to extract different types of images features, typically focused on different scales and crowd densities, and fuse those features together to produce the final density map. For example, Zhang et al. [1] proposed the multi-column convolution neural network (MCNN), which utilizes three separate network paths. The first path uses 9x9 and then 7x7 kernels. The second path uses 7x7 and then 5x5 kernels. And the final path uses 5x5 and then 3x3 kernels. The resulting features from each path are then fused by concatenating them together. The intuition behind using different network paths with different kernel sizes is that each kernel will be suited for detecting objects at a specific scale. Boominathan et al. [69] proposed a similar approach, which involved using a deep network and a shallow network to capture both the high-level and semantically rich features and the low-level highly localized features. These features were also fused using concatenation. Onoro-Rubio et al. [70] proposed an approach that operated on a pyramid of image patches and passed the patches through different network paths meant to process different scales. Babu Sam et al. [71] proposed Switch-CNN, which is conceptually similar to MCNN [1], except that it decomposes an input image into patches and has an additional parameterized model that attempts to determine which network path is best suited to handle the patch. Jiang et al. [72] proposed a novel trellis encoder-decoder network which aggregated multi-scale features from different depths in a single network, and then fused features together at different scales to produce four

different density maps with different resolutions. The loss function for the network would then compare each of these four density maps to a ground truth generated at each respective resolution. Liu et al. [73] introduced a strategy for aggregating multi-scale features using a multi-scale pooling operation that allows for adaptive re-weigghting of features using local and global context. The common thread between all of these methods is that they focus on finding ways to selectively process image features across different scales using different network pathways.

Babu Sam et al. [74] and Sindagi et al. [75] explored a different strategy for feature fusion which used concepts known as top-bottom and bottom-top feature fusion. Bottom-top feature fusion fuses together features from early layers in the network with later layers. Top-bottom feature fusion follows the opposite strategy, taking the deepest features and gradually merging them upwards. This strategy attempts to use the high level semantic information of deeper layers to correct the information found in shallower layers.

Multi-scale feature extraction strategies are a common and well-explored approach for attempting to incorporate additional knowledge about the presence of scale in object counting datasets. The goal of including this information is to alleviate the model from needing to learn about scale from the data, as the model is designed to handle scale from initialization. However, these design choices are often based on intuition about scale. Explicit design choices tailored towards multi-scale feature extraction have become less common within the literature, and have been replaced by more sophisticated strategies.

9.2 Task-Specific Convolutions

Convolutional layers in neural networks take a small weighted matrix, with shape $N \times M$, and convolves it with an much larger input feature. This can be thought of as similar to a sliding window algorithm. The weighted matrix is then applied across the input when calculating the resulting features. Custom convolutions extend the basic idea of convolutional layers, and typically employ special kernels or additional operations which attempt to better align the convolutional layer with the task.

Li et al. [59] proposed CSRNet, a neural network based on a VGG16 backbone feature extractors. As overviewed in figure 11, their method for decoding VGG16 features into the final density map involves a stack of dilated convolutional layers. Dilated convolutions are sparse weighted matrices that increase the size of the kernel without adding additional parameters. Using dilated convolutions allows a neural network to more easily learn long range dependencies without drastically increasing the number of parameters, as dilated convolutions are essentially sparse equivalents to larger kernels. Liu et al. [76] proposed using deformable convolutions [77] for object counting in dense scenes. This method uses a learned offset field to produce a deformable matrix which processes signals non-uniformly. The regions sampled by the deformable kernel are

not necessarily aligned with the grid, and so features are processed using bilinear interpolation. This method allows a network to better adapt to complex local features when compared to the standard grid-like convolution.

Custom convolutions are an interesting approach, which explore how convolutional layers might be better aligned with the object counting task. The success of convolutional layers has been widely attributed to image-specific inductive biases, such as translation invariance and locality [64]. Likewise, there may be additional object counting specific inductive biases which can be introduced through novel custom convolutions. The methodologies presented above highlight the value of these custom operations and how they might introduce better inductive biases.

9.3 Perspective

Perspective information has been utilized frequently throughout the object counting literature [6, 58, 78, 79]. Unlike the multi-scale methods which were discussed in section 9.1, including perspective information explicitly provides a neural network with a rich source of information about the distribution of object scale within the images. Including perspective information reduces the complexity of the problem, as the model no longer has to attempt to learn this property from the data.

The recent works of Shi et al. [78] explored the problem of estimating perspective maps and including them within their object counting method. Their method exploited the basic principles of how cameras work. Provided with their assumptions about the camera model, the location of the head of an person is given as a function of the focal length, the camera height, the person's depth, and the person's height:

$$y_h = \frac{f(C - H)}{z_1}$$

Similarly, the location of the person's feet in the image is given by:

$$y_f = \frac{fC}{z_1}$$

By re-arranging these two equations, the following relationship between the camera height, the true height of the object, and the scale of the object within the image can be determined:

$$h = \frac{H}{C - H} y_h$$

which can then be used to define a perspective map as follows:

$$p^g = \frac{h}{H} = \frac{1}{C - H} y_h$$

Generating the perspective map then requires estimating H and C. To do this, Shi et al. [78] estimated the height of humans, H, as 1.75 m. They then manually annotated the heights of a few objects per image to estimate the fixed camera height C and acquire the ground truth perspective maps.

To utilize these generated perspective maps, Shi et al. [78] trained a neural network to first estimate the ground truth perspective map. They then used the estimated perspective maps to adaptively re-weight three multi-scale density maps before combining them to get the final density map estimate. Yang et al. [79] followed up on this work by noticing that the perspective maps predicted by the network developed by Shi et al. [78] were overly noisy and a potential source of error. Instead, Yang et al. proposed an alternative formulation of the problem which involved using the perspective maps to deform the input images using a uniform grid. They trained a neural network to estimate the perspective maps, and then used those perspective maps to deform the input images during training with the goal of reducing the scale complexity of the problem by attempting to make all objects roughly the same scale.

Perspective maps are a rich source of information which can be exploited to simplify the object counting problem in scenes that are characterized by diverse perspectives and a high variance in object scale specifically due to perspective. However, perspective information is not always relevant to object counting problems. Some problems may focus on images taken from an aerial view, where perspective has a reduced impact on object scale. Other problems may deal with objects that have high inter-object scale variance which is a natural property of the object class and not related to perspective at all. Further, perspective maps must be collected by annotating images, which introduces an additional annotation burden to the problem. Given this, perspective maps can be viewed as a powerful additional tool for solving counting problems in situations where perspective problems are common and the burden of collecting perspective maps is tolerable.

9.4 Attention

Attention mechanisms refer to a broad class of deep learning methods. Stated as simply as possible, attention mechanisms are strategies for learning to focus on the important parts of an input. We will discuss three forms of attention as they relate to object counting methods: feature-wise self-attention, input space attention, and gating.

The self-attention mechanism was first proposed by Vaswani et al. [63] in their seminal work on the topic. Self-attention utilizes a scaled dot-product strategy to learn long-range dependencies across the set of features. Given the ability to learn long-range dependencies, self-attention mechanisms are very applicable to object counting problems. Zhang et al. [80] extended upon the idea of self-attention by introducing a local self-attention and global self-attention module for learning short-range and long-range interactions across an image. Lin et al. [65] proposed a multifaceted attention network, which

utilizes three forms of attention. First, a self-attention mechanism. Second, a learnable region attention. And third, instance attention.

Gating refers to an attention strategy which involves calculating the element-wise product between a learned weighted gating vector and image features, such that only important parts of the image features are passed forward. Chen et al. [81] proposed a technique known as variational attention for domain-specific attention. They opt to model domain-specific latent variables and utilize the resulting variable as a gating vector to perform domain specific channel-wise re-weighting. Liu et al. [82] proposed DecideNet, which utilizes an additional network that produces a gating vector for deciding between two different density map proposals. The first proposal is based on a Faster R-CNN framework and outputs bounding box proposals which are converted into density maps. The second proposal is based on patch-based local counting. Zhang et al. [83] proposed attentional neural fields, which utilize an attention mechanism and conditional random fields to aggregate multi-scale features.

Input space attention is a strategy for focusing on important regions of the input space. This could involve dropping irrelevant input image regions or manipulating relevant regions in some way. Liu et al. [76] proposed an attention mechanism for generating attention maps which were used to mask the input before passing it to an additional density map estimating neural network. Liu et al. [84] proposed a strategy for generating attention maps that highlight ambiguous image regions and then they recurrently zoom in on these regions to better resolve the count. Jiang et al. [85] proposed an attention scaling mechanism, which masks image regions based on object density and processes each region with an object density specific network.

Attention is an important technique within deep learning methodologies and provides significant improvements when applied to object counting tasks. Guiding the network towards the most important features, and learning important relationships between features, is a useful inductive bias for improving solutions to these problems. However, it is still unclear how to best include attention mechanisms into object counting frameworks, as the vast majority of the literature explores attention for classification or similar recognition tasks. Given this, attention remains and interesting aspect of object counting methods and an open problem.

9.5 Loss Formulations

The previous sections have focused primarily on architectural choices or ways to provide additional information during network training. An separate branch of methods and techniques involve manipulating the output, typically using a novel loss formulation or by modifying the predicted density map before applying the loss. We will discuss three categories of loss formulations that exist within the object counting literature: modified outputs, multi-task loss functions, and generalized loss functions.

Shu et al. [68] proposed an output modification strategy, which involved transforming the predicted and ground truth density maps into the frequency

domain. Rather than calculating the pixel-wise error between the predicted and ground truth density map, their strategy involves taking the L1-norm between the characteristic functions of both in the frequency domain. They further prove that this loss formulation acts as an upper bound on the pseudo sup norm metric between the predicted and ground truth density maps and that minimizing the loss minimizes this upper bound.

Wan et al. [86] and Wang et al. [87] explored the idea of reframing learning from density maps as an optimal transport problem. Wan et al. [86] proposed this idea as a generalized loss function for object counting with density maps. They argued that the L2-norm between a predicted and ground truth density map may not be the ideal learning signal for the counting task. They opt to treat the problem as an unbalanced optimal transport problem and formulate a novel loss function based on this property. Further, they demonstrate that previous loss functions, such as the L2-norm between the predicted and ground truth density map, are suboptimal solutions to the optimal transport problem.

Multi-task losses explore the idea of adding additional training signals as additional loss functions. The intuition behind multi-task learning is that by targeting multiple related tasks with different loss formulations, the model may learn a more robust representation that generalizes better. Zhao et al. [88] proposed a multi-task loss which decomposed the counting problem into several core components. Their method jointly predicts geometric, semantic, and numeric attributes. Idrees et al. [11] proposed a multi-task loss that targeted multiple types of density maps, formulated with different σ values. Shi et al. [89] proposed a multi-task loss that predicted a segmentation mask, the density map, and the global density. These methods typically use additional losses, related to localization but different than density map regression, to learn more useful features for solving the counting problem.

9.6 Using Dot Maps

Density maps are generated from dot maps, which simply contain a single dot at the center of each object. One line of research has explored using dot maps directly, rather than using density maps as an intermediary. Laradji et al. [90] designed a model which predicted segmentation maps containing blobs around the ground truth points in the dot map. They treated the dot maps as a sparsely annotated segmentation map, with the points representing annotated pixels in the true underlying segmentation map. They introduced a split-level loss, which forces the model to break apart blobs that contain more than one ground truth point. Additionally, they introduce a false-positive loss, which penalizes the model for predicting blobs which contain no ground truth points. And finally, they propose a point-level loss which ensures that the model always correctly predicts the few sparsely annotated pixels available during training. Ma et al. [91] argued that using density maps as the training target is sub-optimal due to occlusion and irregular crowd densities. They instead proposed a novel loss formulation which targeted the expected count at each annotation point, calculated using Bayes' theorem, which takes the summation

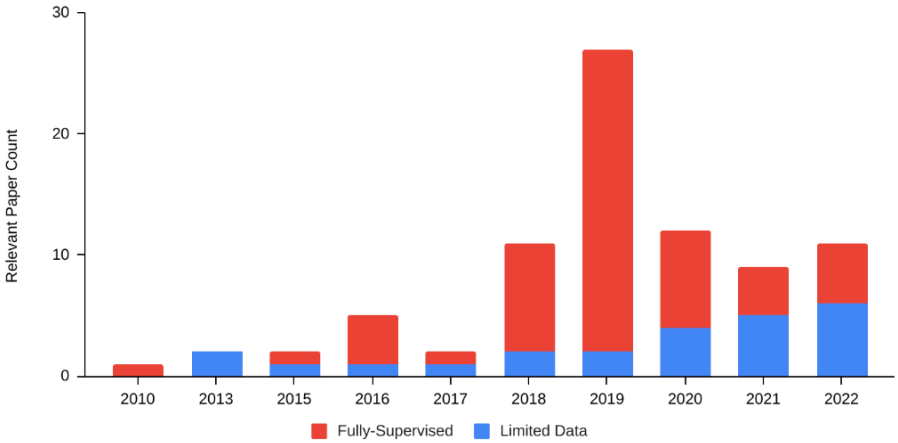


Fig. 14: Among the papers which we have reviewed in this report, there is a clear trend that solutions to object counting problems with limited data are becoming increasingly more common within the literature while fully-supervised methods are becoming less common.

of contribution probability at each point. Liu et al. [92] proposed the idea of using dot map annotations to generate pseudo bounding boxes targets, and then iteratively update those bounding box proposals during training to better include object size information. Song et al. [93] proposed a purely point based approach, which adopts methods similar to those found in the key-point matching literature, where they directly regress point coordinates and use the Hungarian algorithm to optimally match predicted points to ground truth points.

These methods seek to use the localization signal present in dot maps directly by forgoing density map regression. Instead, they present a novel strategy for utilizing the dots as anchor points in alternative formulations of the problem. These papers take the perspective that density maps are not ideal targets, and that different problem formulations may lead to models that generalize better.

10 Counting with Limited Data

Fully-supervised object counting problems rely on density map annotations, which carry a significant annotation burden and are labour intensive to collect. As highlighted earlier, Cholakkal et al. [60] determined that each dot takes approximately 1.1 s for an annotator to collect. Finding ways to solve object counting problems when only limited data is available becomes an important problem for increasing the usage of object counting methodologies in problem spaces where annotations are sparse or challenging to collect.

Fig. 15: An example of the visual grounding problem. The symbol for this image is the object class label "Fish". However, there are multiple conflicting visual stimuli in the image that may be spuriously correlated with the object class label.

Counting with limited data refers to the task of solving the counting problem when some information is missing as compared to the typical fully-supervised setup. For example, this could involve utilizing a dataset where the examples are annotated with a weaker signal. Similarly, this could involve using an automatically annotated synthetic dataset and finding a way to adapt features learned on this synthetic dataset to an unlabelled real dataset. There are many potential formulations of low-data versions of the object counting problem. However, there is a shared underlying problem that emerges when attempting to solve these problems. While density maps create an explicit correspondence between object locations and the predicted counts, limited data setups may be working with some portion of the data where the signal has no correspondence with location at all. This is known as the visual grounding problem.

10.1 The Visual Grounding Problem and Object Recognition

Visual grounding is the problem of correctly determining what stimulus in an image gives a symbol meaning, where a symbol is any set of attributes. Examples of symbols in computer vision problems include a set of object class labels (such as the 1000 ImageNet classes [47]), captions/language, or global count labels. An example of an image, a set of visual stimuli, and an object class label are presented in figure 15. Here we can see that a problem emerges when attempting to attribute a stimulus to the object class label. For this particular example, a dataset may contain many images within the "Fish" object class that carry additional stimuli such as water, fishing nets, and people. If these stimuli co-occur frequently enough with the "Fish" object class, then

it becomes incredibly challenging for a neural network to learn how to select the appropriate stimuli. This problem is commonly referred to as a spurious correlation, as an irrelevant feature spuriously co-occurs with the label due to biases in the dataset.

Choe et al. [94] analyzed this problem within the context of weakly supervised object localization, which is the task of correctly attributing image pixels to the object class label using only the object class label during training. Their analysis demonstrates that if spurious features are more strongly correlated with the label than some actually relevant object part, then the localization task cannot be solved. Assume that for some problem you have a set of image patches, where each patch is defined by a high-level semantic cue, M, such as "water", "fish tail", "fish head", "net", "man holding something", etc. And suppose that the true posterior distribution, $p(Y = \texttt{Coho,Salmon}|M)$, is known ahead of time. The posterior distribution is the percentage of examples where the semantic cue M co-occurs with the label Y. If there exists a situation, such as

$$p(Y = \texttt{Coho,Salmon}|M = \texttt{water}) > p(Y = \texttt{Coho,Salmon}|M = \texttt{fish tail})$$

then we can see that there exists no threshold on the posterior distribution that perfectly selects all of the relevant cues while ignoring all of the spurious cues.

An additional challenge that makes solving the visual grounding problem difficult is the simplicity bias. Described by Shah et al. [57], the simplicity bias refers to a tendency for neural networks to preferentially learn overly simple functions, even when a more complex function would lead to a lower training error on a given dataset. As discussed above, distinguishing the correct set of relevant visual stimuli is potentially not possible, even when you have access to the true posterior distribution. The tendency for neural networks to preferentially learn simple functions means that access to the true posterior distribution of high-level semantic cues may not be accessible at all. These two problems make localizing the true signal within an image using only global labels an incredibly challenging task.

Returning to the object counting problem, the intuition behind selecting density maps as a supervisory signal becomes apparent. By re-framing the object counting problem as a density based object localization problem, the training signal is explicitly grounded within the image. Given this, the challenge of solving object counting problems with limited data must give consideration to the problem of visual grounding. The remainder of this section will discuss the various ways that methodologies for solving object counting problems with limited data attempt to also solve the problem of visual grounding, with many methods exploring clever ways to continue utilizing density maps while lowering the annotation burden.

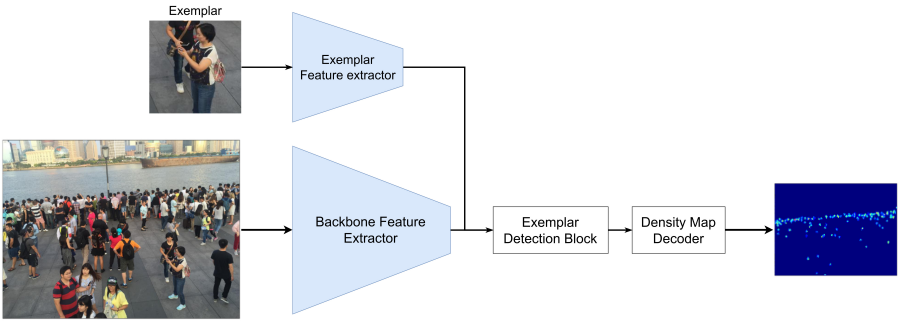


Fig. 16: Overview of a typical architecture formulation for a class-agnostic counting problem. An exemplar, which is meant to be an informative prototype for the objects of interest, is passed through a feature extractor and used to guide the main counting network in detecting objects in the given image that match the exemplar. Figure uses modified image from ShanghaiTech part B dataset [1]

10.2 Few-Shot Object Counting

Few-shot Object Counting refers to the problem of training an object counting network which is class agnostic. The typical formulation, detailed in figure 16, involves training an object counting network using image and exemplar pairs while using a density map as the training target. An exemplar is defined as an additional image containing a single example of the object which can be used by the network as a prototype for detecting that object in the whole image. Essentially, one network is responsible for extracting rich object-centric features from an image, and a second network is responsible for generating a prototype that can be used to select the most relevant objects from the resulting features. These methods don't explicitly learn from limited data. Instead, this strategy involves using a lot of fully-supervised data to learn a class agnostic counting model that can then later be applied to unseen object categories with limited additional data being required. In this way, it is a limited data method with respect to the unseen object categories.

Ranjan et al. [95] produced the seminal work on the topic of few-shot object counting by introducing the FSC-147 dataset. This dataset contains 6135 images across 147 object categories with an average of 56 objects per image. Each image was annotated with a ground truth density maps. Their work also introduced FamNet, which utilized ResNet50 [62] as a feature extractor. Their method then produced a feature correlation map using the image features and the exemplar features. This feature correlation map was then decoded into a density map using an additional network. Shi et al. [96] explored the idea that computing the similarity between image features and exemplar features via a fixed inner product is not optimal. Instead, they proposed a learnable bilinear similarity metric. Gong et al. [97] approached the challenge of intraclass diversity, which includes factors such as color, shape and scale. Essentially, an

exemplar may not contain enough information to capture the full diversity of an object class. They introduced the concept of Exemplar Feature Augmentation to generate a more diverse set of exemplars. Further, they introduce the concept of Edge Matching, which introduces shape priors into the training process. Nguyen et al. [98] approached a slightly more challenging version of the problem, where they were focused on generating both density maps and bounding box proposal. They introduced an uncertainty aware training strategy for generating pseudo ground truth bounding boxes from density map proposals, which were then used as the targets during a second round of training. Few-shot object counting is an active area of exploration, and open problems primarily focus on finding better ways to learn features that robustly define object classes and methods for correlating image features and exemplars.

10.3 Domain Adaptation

Domain adaptation refers to the problem of training a model using a dataset from a source domain and then transferring that knowledge to a similar target domain which may have an underlying distribution shift. In practice, a source object counting dataset tends to be collected from a few environments and the underlying properties of that dataset may not represent the general case for those objects. As a hypothetical example, a crowd counting dataset could have been collected entirely from indoor environments while a target distribution could have been collected entirely outdoors. The underlying distribution shift between indoor environments and outdoor environments could potentially change the image features enough to prevent the model from generalizing to the new environment. However, methodologies that learn robust features on a source domain which can be adapted to a new domain is a promising research direction, as it alleviates the need to collect additional fully-supervised data for every new environment where the objects might be found.

Zhang et al. [99] proposed a method for object counting domain adaptation which involved first training a network on a source domain. Then, they use their network to locate patches within the source domain that looked most like the target domain to further fine-tune the network. Essentially, their goal was to detect a subset of the source distribution that best matched the target distribution with the goal of improving domain adaptation. Marsden et al. [24] proposed a local patch based method for regressing the object count coupled with a residual adapter module that performs domain-specific normalisation and scaling. With the goal of learning from several domains, they then trained their model on four distinct domains; people, penguins, vehicles, and cells. Ma et al. [100] explored domain adaptation from the perspective that object counting methods are severely impacted by scale shifts between distributions. They propose a scale alignment module which derives an optimal re-scaling factor between scale distributions.

(a) Synthetic data from GCC dataset [12]

(b) Real data from ShanghaiTech B dataset [1]

Fig. 17: Zoomed in crops of crowd counting images from a real and a synthetic dataset. The underlying domain gap makes task transfer from synthetic to real domains a challenging problem.

10.3.1 Synthetic Data

Learning to count objects from a synthetic image distribution is a special case of the domain adaptation problem. In this version of the problem, a large synthetic dataset is generated automatically using various 3D assets, which also allows for the automatic production of density map annotations. However, as can be viewed in figure 17, the difference between synthetic and real distributions can be significant. Thus, it becomes worthwhile to talk about synthetic domain adaptation as a distinct problem.

Historically, synthetic datasets have been applied to many computer vision problems. For example, the Synthia dataset [101] contains synthetic images utilized for semantic segmentation in urban scenes. Likewise, the Virtual KITTI dataset [102] contains synthetic images utilized for multi-object tracking. Both of these datasets contain pedestrians, and can potentially be utilized for crowd counting problems. However, both of these datasets contain a low density of pedestrians, and are not necessarily representative of the real world crowd densities that are regularly explored in crowd counting problems.

Wang et al. [12] proposed the GCC dataset, a synthetic crowd counting dataset, which utilised character assets from the Grand Theft Auto V video game. Their approach to the problem of synthetic domain adaptation involved using SE Cycle GAN, a method typically used for style transfer in images, to transform the synthetic images such that they appear in the style of the target domain. Liu et al. [103] approach the problem by jointly training a neural network on the synthetic data in a fully-supervised fashion and on the real data in a self-supervised fashion. Their network attempts to solve a proxy task, which involves predicting whether or not an image from the real

distribution has been flipped upside down. Objects within a scene tend to have a well-defined orientation, and act as a strong signal for image orientation. The authors found that by jointly training a network on these two tasks, they were able to better generalize to the target distribution. Gong et al. [104] introduced a method for task-driven data alignment between source and target domains, and developed a strategy for finding the optimal set of source domain image transformations.

Domain adaptation is a promising direction, which attempts to alleviate the need for to acquire new data for similar objects under previously unseen conditions. Synthetic domain adaptation, in particular, has emerged as an interesting direction. However, while 3D assets are often cheap to collect, they still suffer from a large domain gap. Further, there exists many object classes for which 3D assets are not readily available.

10.4 Self, Weak and Semi-Supervision

Self-, weak-, and semi-supervised object counting methods refer to a broad framework of weak supervision strategies. These methods attempt to learn from limited, if any, fully supervised examples. They utilize a significantly less informative signal and attempt to extract as many relevant features as possible from the data, typically by using a clever prior.

Semi-supervised object counting refers to the task of learning from a small amount of fully-supervised density maps and a large quantity of completely unlabelled examples. They typically approach the problem by attempting to extract a weak signal from the unsupervised data to help reduce the set of possible functions that fit the small amount of fully-supervised examples. Change et al. [105] and Zhao et al. [106] proposed active learning frameworks, which focused on identifying the most informative images for annotation, and then learned from the remaining images in an unsupervised way. Change et al. [105] enforced a manifold constraint on the unlabelled images using a Laplacian regularised least squares strategy by taking the assumption that the data generation process is similar between the labelled and unlabelled portions of the dataset. Liu et al. [107] proposed a self-supervised proxy task for learning from the unlabelled portion of the data. Their strategy involved recognizing that for any image, a sub-crop of that image must have as many or fewer objects. They then proposed a learning strategy for ranking images and image sub-crops pairs while jointly optimizing their network using the available fully-supervised density map annotations. Sam et al. [108] proposed a strategy for learning unsupervised features using a Grid Winner-Take-All autoencoder, and then fine-tuning the learned features on a small amount of fully-supervised annotations. Liu et al. [109] utilized self-training by learning a proxy segmentation task using the available fully-supervised data and then used the pseduo-segmentation masks generated on the unlabelled data as a target during training. Sindagi et al. [110] proposed an iterative self-training method using a Gaussian Process for estimating the pseudo ground truth on unlabelled

data. Meng et al. [111] used a spatial uncertainty aware teacher-student framework to learn from a surrogate task over the unlabelled data. Lei et al. [112] proposed a weakly-semi supervised approach, which utilized a small amount of fully-supervised density maps and a large quantity of global object counts.

Object counting with sparse density map annotations involves learning to count when only a subset of each image has been annotated. In the typical formulation, 10% of an image region is annotated while the remaining 90% of an image is unannotated. Xu et al. [113] approached this problem by proposed a module which encouraged visual features to flow from annotated regions to unannotated regions. This approach works by extracting a training signal from the annotated regions in a supervised way and the unannotated regions in an unsupervised. More importantly, it exploits the fact that an annotated region contains information about the object density in the neighboring unannotated regions. This distinction separates the problem from a typical semi-supervised formulation, where only a subset of the images are fully annotated. Similar approaches have been applied to weak object detection problems [114]. The intuition appears to be that knowing a little bit about 10 images can potentially be more useful than knowing a lot about 1 image. It has been postulated that inter-image differences in the traditional semi-supervised setup can prevent the learning of useful features from the unlabelled data.

Weakly-supervised object counting methods utilize a weaker signal, which still requires manual annotation to collect, but which is less informative than the fully-supervised examples. Borstel et al. [115] proposed a method for learning from local count patches, rather than density maps, using a Gaussian process prior on a latent function. Yang et al. [116] proposed a method for learning exclusively from global object counts. Their method used a sorting loss which enforced that the network learn inter-example relationships across the dataset.

Self-supervised object counting attempt to learn how to count with no labels whatsoever. This is an incredibly challenging problem, and requires sophisticated and highly curated priors to achieve anything that could be considered a reasonable result. Babu et al. [117] proposed a fully self-supervised approach to crowd counting. Their method first trains a neural network on a self-supervised proxy task, such as predicting image rotation. They identified that the distribution of object counts in patches from crowd counting images follows a power law distribution, and they use an optimal transport law to strictly force the features of the self-supervised model to follow this power law distribution. However, they tested their method exclusively on dense scene and avoided evaluating on sparse datasets where crowds may not be uniformly distributed across every image.

These methodologies, which attempt to alleviate the annotation burden by learning from unlabelled or weakly labelled data, represent a promising direction for object counting problems. However, there is still significant progress necessary to bring the performance of these methods closer to fully-supervised methods.

11 Future Directions

11.1 Considering Downstream Usage

Enumerating the number of objects within an image is often an intermediate step in solving a downstream domain-specific task. Therefore, for future work, it will be interesting to collect downstream task related ground truth labels and train end-to-end model for solving those tasks, while using the count labels as an additional source of information for regularization.

For example, the Non-Violent Action Lab is a lab which monitors and disseminates knowledge pertaining to social movements and protests. They specifically monitor crowd sizes at political events as a method for estimating the extent of public support for political movements. However, estimating the object count in a single 2D image often does not capture the entire crowd. This is due to the fact that for sufficiently large crowds, a single image cannot capture every object within the crowd. Given that crowd size estimation is used to estimate abstract properties such as political support and counting is used as a proxy metric, there may be value in more explicitly coupling the count to that task. Therefore, there may be additional ground truth labels which could accompany crowd counting labels. For example, tracking the number of related social media posts at the time of an event may be a useful additional label for estimating the total extent of the crowd, in addition to the per-image crowd density.

Similarly, estimating the quantity of a plant organ, such as fruits or leaves, is useful for forecasting the expected yield for that crop. However, the per-image count is not the actual goal in this scenarios. For example, in the example of yield forecasting, the goal is the estimate productivity of that crop for the farmer. This is based on several complex factors, including temperature, fertilizer application, watering schedule, etc. There may be value in collecting yield ground truth, and estimating the yield directly from images and the environmental factors.

In domains such as medical image analysis, counts of lesions or cells can be used within the decision-making process. For example, tracking the number of acne lesions is a useful metric when making decisions related to disease management. If the goal is to use counts to make decisions, then there may be value in including those decision as ground truth labels.

11.2 Domain Considerations

The object counting literature has been biased towards crowd counting problems, which represents only a fraction of the relevant domains explored within the object counting literature. Crowd counting methods have focused on solving problems related to occlusion and scale due to perspective-based object variance, which serves as the central challenge in the available datasets. However, these methods may not be relevant in other domains. For example, cell counting problems [24, 25, 39] and aerial view vehicle counting problems [4, 15]

do not share the same perspective profile as crowd counting problems. This indicates that there may be unexplored research directions which specifically tackle domains with unique challenges.

11.3 Synthetic Data

While synthetic datasets such as GCC [12] have been developed for counting problems, they are still limited due to their scope and lack of realism. GCC only tackles crowd counting problems, likely due to simple 3D assets for humans being highly accessible. Further, there is still a large domain gap between these synthetic images and real images, which has prevented performance gains [103]. Generating high-quality and realistic synthetic images without relying on 3D assets would allow the synthetic counting problem to easily extend to other object classes, and would significantly reduce the annotation burden.

Guided image synthesis is an emerging field that has seen significant strides forward with the development of technologies such as stable diffusion [118]. Recent examples, such as SDEdit [119], have focused on guided image synthesis through stroke-based guidance and compositing based guidance. Stroke-based guidance utilizes user guidance to select image regions using a digital brush-stroke, with the user having brushes with distinct semantic meaning. The diffusion based editing system then completes the scene by generating a synthetic image that matches the semantics of the strokes. Compositing based guidance involves copying image patches representing content from one image, and integrating it into a second image. For example, copying an image patch with glasses and placing it on an image of a person's face. Diffusion based guidance can then integrate the patch and produce a realistic combination of the two. Guided image synthesis has not yet been applied to object counting problems, and represents a novel research direction for reducing the annotation burden by streamlining the process of high-quality synthetic image generation.

Automatic generation of complex 3D scenes represents an additional option for synthetic dataset generation. Kubric [120] has been recently proposed as a Blender based dataset generator for the fast production of 3D multi-object synthetic data which also provides a wide variety of complex automatic annotations. These dataset generators could be used for the development of synthetic few-shot learning methods, similar to the real datasets presented in section 10.2. A method trained using a synthetic few-shot dataset may transfer to an real object of interest without needing to generate an object-specific synthetic datasets

11.4 Multi-Class Counting

Multi-class object counting [54, 60, 121] and few-shot object counting [95] have only been explored to a limited extent. Cholakkal et al. [60, 121] explored multi-object counting in sparse scenes using global object counts and object class labels. However, their work predominately focused on sparse scenes, with object counts within or close to the subitizing range for humans. Similarly,

Chattopadhyay et al. [54] also focused on multi-class object counting problems within the subitizing range. While few-shot object counting [95] have approached the problem by focusing on multi-class dataset with dense object counts, well outside of the subitizing range, these problems predominately focus on a single object category per image, designated by a class defining exemplar. Given this, we argue that there is a need for dense multi-class object counting datasets where multiple object categories within an image are interacting.

11.5 Multi-Modal Counting

Counting objects within images primarily relies on only a single modality. However, there are many additional modalities which complement the image-based counting problem. For example, crowd counting with audiovisual [122], thermal imaging [123, 124], and depth [125] have been explored. However, this research topic has only been minimally explored, with each distinct sub-domain offering unique modalities. For example:

- Crowd size estimation has been performed using mobile phone and Twitter data [126]. This additional modality may be combined with crowd imaging data.
- Yield forecasting in agriculture has frequently relied on remote sensing [127] to provide measurements about croplands. This modality could be combined with in-field image-based counting of plant organs.
- Medical imaging problems typically involve multiple modalities such as hybrid PET/CT scan, and multi-model MRI (e.g. T1 and T2 weighted). Medical image counting problems, such as counting multiple sclerosis lesions [41–43], may benefit from these additional modalities.

Thus, we argue that multi-modal object counting is an under explored problem, and there exists a vast number of modalities that may potentially benefit domain specific object counting problems.

12 Ethical Considerations

Solutions to object counting problems, and especially object counting problems with limited data, have a wide variety of applications that benefit different fields. However, crowd analysis has continued to be the dominant application within the research community, with pedestrian specific datasets representing the majority of benchmark datasets used for evaluating object counting methods. Given that the surveillance of people is such a dominant topic in the space, it is worth considering the ethical implications of making surveillance cheaper. Surveillance methods can potentially be used by unethical governments to violate personal privacy and autonomy, suppress dissent, and gain tactical military advantages. However, crowd density estimation also provides useful information in large event management, disaster management, and public safety. Further, object counting methods typically only provide an estimate

of the count, and do not necessarily localize faces and make it easier to re-identify individuals. Face detection and recognition methods, which can be used to re-identify individuals, are already a well-defined and separate research topic. Thus, access to accurate crowd analysis is not necessarily easily utilized for unethical means. Further, other methods often perform better at tasks that may be exploited for unethical means. As a positive example, the methods developed by Cheng et al. [67] were used to perform crowd analysis of the January 6th capital riot, for which they were awarded the 2022 Pulitzer Prize for public service. Given this, crowd counting methods seem to have significant positive benefits that are worth pursuing from a research perspective and limited unethical use cases. However, researchers should remain vigilant when providing contributions to the space and make themselves aware of who is using their research and how it is being applied.

13 Conclusions

Object counting is an important computer vision task, which has several applications across a wide variety of domains. Progress within the field has historically been driven by density map annotations. However, these annotations are labour intensive to collect and contain significant noise, errors, and inter-annotator variance. More recent approaches have attempted to alleviate the burden of density map annotations by proposed limited data approaches, such as few-object counting, domain adaptation, semi-supervised learning, weakly supervised learning, and self-supervised learning. The field has shown a trend towards these approaches and away from fully supervised approaches. However, these limited data methodologies do not achieve similar performance when compared to fully supervised methods, and significant work must be done to find new strategies for improving these methods.

13.0.1 Acknowledgements

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), Mitacs, and the Simon Fraser University Graduate Fellowships and Graduate Dean's Entrance Scholarship.

13.0.2 Statements and Declarations

Funding.

Funding was supplied in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) scholarship, Mitacs Accelerate Project IT10016, Mitacs Accelerate Project IT18522, the Simon Fraser University Graduate Fellowships and the Simon Fraser University Graduate Dean's Entrance Scholarship..

References

[1] Zhang, Y., Zhou, D., Chen, S., Gao, S., Ma, Y.: Single-image crowd counting via multi-column convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 589–597 (2016)

[2] Arteta, C., Lempitsky, V., Zisserman, A.: Counting in the wild. In: European Conference on Computer Vision (2016)

[3] Häni, N., Roy, P., Isler, V.: Minneapple: a benchmark dataset for apple detection and segmentation. IEEE Robotics and Automation Letters **5**(2), 852–858 (2020)

[4] Hsieh, M.-R., Lin, Y.-L., Hsu, W.H.: Drone-based object counting by spatially regularized regional proposal network. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4145–4153 (2017)

[5] Wang, Q., Gao, J., Lin, W., Li, X.: Nwpu-crowd: A large-scale benchmark for crowd counting and localization. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020). https://doi.org/10.1109/TPAMI.2020.3013269

[6] Chan, A.B., Liang, Z.-S.J., Vasconcelos, N.: Privacy preserving crowd monitoring: Counting people without people models or tracking. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–7 (2008). IEEE

[7] Chen, K., Loy, C.C., Gong, S., Xiang, T.: Feature mining for localised crowd counting. In: Bmvc, vol. 1, p. 3 (2012)

[8] Zhang, C., Kang, K., Li, H., Wang, X., Xie, R., Yang, X.: Data-driven crowd understanding: A baseline for a large-scale crowd dataset. IEEE Transactions on Multimedia **18**(6), 1048–1061 (2016)

[9] Yan, Z., Yuan, Y., Zuo, W., Tan, X., Wang, Y., Wen, S., Ding, E.: Perspective-guided convolution networks for crowd counting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 952–961 (2019)

[10] Idrees, H., Saleemi, I., Seibert, C., Shah, M.: Multi-source multi-scale counting in extremely dense crowd images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2547–2554 (2013)

[11] Idrees, H., Tayyab, M., Athrey, K., Zhang, D., Al-Maadeed, S., Rajpoot,

N., Shah, M.: Composition loss for counting, density map estimation and localization in dense crowds. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 532–546 (2018)

[12] Wang, Q., Gao, J., Lin, W., Yuan, Y.: Learning from synthetic data for crowd counting in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8198–8207 (2019)

[13] Sindagi, V., Yasarla, R., Patel, V.M.: Jhu-crowd++: Large-scale crowd counting dataset and a benchmark method. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)

[14] Wang, Q., Gao, J., Lin, W., Li, X.: Nwpu-crowd: A large-scale benchmark for crowd counting and localization. IEEE transactions on pattern analysis and machine intelligence **43**(6), 2141–2149 (2020)

[15] Mundhenk, T.N., Konjevod, G., Sakla, W.A., Boakye, K.: A large contextual dataset for classification, detection and counting of cars with deep learning. In: European Conference on Computer Vision, pp. 785–800 (2016). Springer

[16] Guerrero-Gómez-Olmedo, R., Torre-Jiménez, B., López-Sastre, R., Maldonado-Bascón, S., noro-Rubio, D.O.: Extremely overlapping vehicle counting. In: Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA) (2015)

[17] Hoekendijk, J., Kellenberger, B., Aarts, G., Brasseur, S., Poiesz, S.S., Tuia, D.: Counting using deep learning regression gives value to ecological surveys. Scientific reports **11**(1), 1–12 (2021)

[18] DataCanary, M.R. Katie: NOAA Fisheries Steller Sea Lion Population Count. Kaggle (2017). https://kaggle.com/competitions/noaa-fisheries-steller-sea-lion-population-count

[19] Lu, H., Cao, Z., Xiao, Y., Zhuang, B., Shen, C.: Tasselnet: counting maize tassels in the wild via local counts regression network. Plant methods **13**(1), 1–17 (2017)

[20] Minervini, M., Fischbach, A., Scharr, H., Tsaftaris, S.A.: Finely-grained annotated datasets for image-based plant phenotyping. Pattern recognition letters **81**, 80–89 (2016)

[21] David, E., Serouart, M., Smith, D., Madec, S., Velumani, K., Liu, S., Wang, X., Pinto, F., Shafiee, S., Tahir, I.S., et al.: Global wheat head detection 2021: an improved dataset for benchmarking wheat head detection methods. Plant Phenomics **2021** (2021)

[22] Xie, W., Noble, J.A., Zisserman, A.: Microscopy cell counting and detection with fully convolutional regression networks. Computer methods in biomechanics and biomedical engineering: Imaging & Visualization **6**(3), 283–292 (2018)

[23] Paul Cohen, J., Boucher, G., Glastonbury, C.A., Lo, H.Z., Bengio, Y.: Count-ception: Counting by fully convolutional redundant counting. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 18–26 (2017)

[24] Marsden, M., McGuinness, K., Little, S., Keogh, C.E., O'Connor, N.E.: People, penguins and petri dishes: Adapting object counting models to new visual domains and object types without forgetting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8070–8079 (2018)

[25] Lonsdale, J., Thomas, J., Salvatore, M., Phillips, R., Lo, E., Shad, S., Hasz, R., Walters, G., Garcia, F., Young, N., *et al.*: The genotype-tissue expression (gtex) project. Nature genetics **45**(6), 580–585 (2013)

[26] Li, B., Huang, H., Zhang, A., Liu, P., Liu, C.: Approaches on crowd counting and density estimation: A review. Pattern Analysis and Applications **24**(3), 853–874 (2021)

[27] Sindagi, V.A., Patel, V.M.: A survey of recent advances in cnn-based single image crowd counting and density estimation. Pattern Recognition Letters **107**, 3–16 (2018)

[28] Thasveen M., S., Mredhula, L.: Real time crowd counting: A review. In: 2020 International Conference on Futuristic Technologies in Control Systems & Renewable Energy (ICFCR), pp. 1–5 (2020). https://doi.org/10.1109/ICFCR50903.2020.9249984

[29] Jingying, W.: A survey on crowd counting methods and datasets. In: Advances in Computer, Communication and Computational Sciences, pp. 851–863. Springer, ??? (2021)

[30] Khan, M.A., Menouar, H., Hamila, R.: Revisiting crowd counting: State-of-the-art, trends, and future perspectives. Image and Vision Computing, 104597 (2022)

[31] Zhan, B., Monekosso, D.N., Remagnino, P., Velastin, S.A., Xu, L.-Q.: Crowd analysis: a survey. Machine Vision and Applications **19**(5), 345–357 (2008)

[32] Zaidi, S.S.A., Ansari, M.S., Aslam, A., Kanwal, N., Asghar, M., Lee, B.: A survey of modern deep learning based object detection models. Digital

Signal Processing, 103514 (2022)

[33] Zou, Z., Chen, K., Shi, Z., Guo, Y., Ye, J.: Object detection in 20 years: A survey. Proceedings of the IEEE (2023)

[34] Zhang, D., Han, J., Cheng, G., Yang, M.-H.: Weakly supervised object localization and detection: A survey. IEEE transactions on pattern analysis and machine intelligence **44**(9), 5866–5885 (2021)

[35] Asgari Taghanaki, S., Abhishek, K., Cohen, J.P., Cohen-Adad, J., Hamarneh, G.: Deep semantic segmentation of natural and medical images: a review. Artificial Intelligence Review **54**, 137–178 (2021)

[36] Kang, D., Dhar, D., Chan, A.: Incorporating side information by adaptive convolution. Advances in Neural Information Processing Systems **30** (2017)

[37] Sindagi, V.A., Yasarla, R., Patel, V.M.: Pushing the frontiers of unconstrained crowd counting: New dataset and benchmark method. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1221–1231 (2019)

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

[39] Kainz, P., Urschler, M., Schulter, S., Wohlhart, P., Lepetit, V.: You should use regression to detect cells. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 276–283 (2015). Springer

[40] Wu, X., Wen, N., Liang, J., Lai, Y.-K., She, D., Cheng, M.-M., Yang, J.: Joint acne image grading and counting via label distribution learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10642–10651 (2019)

[41] Dworkin, J.D., Linn, K.A., Oguz, I., Fleishman, G.M., Bakshi, R., Nair, G., Calabresi, P.A., Henry, R.G., Oh, J., Papinutto, N., *et al.*: An automated statistical technique for counting distinct multiple sclerosis lesions. American Journal of Neuroradiology **39**(4), 626–633 (2018)

[42] Chung, K.K., Altmann, D., Barkhof, F., Miszkiel, K., Brex, P.A., O'Riordan, J., Ebner, M., Prados, F., Cardoso, M.J., Vercauteren, T., *et al.*: A 30-year clinical and magnetic resonance imaging observational study of multiple sclerosis and clinically isolated syndromes. Annals of neurology **87**(1), 63–74 (2020)

[43] Karimaghaloo, Z., Arnold, D.L., Arbel, T.: Adaptive multi-level conditional random fields for detection and segmentation of small enhanced pathology in medical images. Medical image analysis **27**, 17–30 (2016)

[44] Gurcan, M.N., Madabhushi, A., Rajpoot, N.: Pattern recognition in histopathological images: An icpr 2010 contest. In: International Conference on Pattern Recognition, pp. 226–234 (2010). Springer

[45] Lehmussola, A., Ruusuvuori, P., Selinummi, J., Huttunen, H., Yli-Harja, O.: Computational framework for simulating fluorescence microscope images with cell populations. IEEE transactions on medical imaging **26**(7), 1010–1016 (2007)

[46] LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural computation **1**(4), 541–551 (1989)

[47] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009). Ieee

[48] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., *et al.*: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)

[49] Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International journal of computer vision **88**(2), 303–338 (2010)

[50] Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html

[51] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision, pp. 740–755 (2014). Springer

[52] Biggs, B., Boyne, O., Charles, J., Fitzgibbon, A., Cipolla, R.: Who left the dogs out?: 3D animal reconstruction with expectation maximization in the loop. In: ECCV (2020)

[53] Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B.: 2d human pose estimation: New benchmark and state of the art analysis. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014)

[54] Chattopadhyay, P., Vedantam, R., Selvaraju, R.R., Batra, D., Parikh, D.: Counting everyday objects in everyday scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1135–1144 (2017)

[55] Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015)

[56] Bodla, N., Singh, B., Chellappa, R., Davis, L.S.: Soft-nms – improving object detection with one line of code. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017)

[57] Shah, H., Tamuly, K., Raghunathan, A., Jain, P., Netrapalli, P.: The pitfalls of simplicity bias in neural networks. Advances in Neural Information Processing Systems **33**, 9573–9585 (2020)

[58] Lempitsky, V., Zisserman, A.: Learning to count objects in images. Advances in neural information processing systems **23** (2010)

[59] Li, Y., Zhang, X., Chen, D.: Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1091–1100 (2018)

[60] Cholakkal, H., Sun, G., Khan, S., Khan, F.S., Shao, L., Van Gool, L.: Towards partial supervision for generic object counting in natural scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)

[61] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)

[62] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

[63] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)

[64] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021). https://openreview.net/forum?id=YicbFdNTTy

[65] Lin, H., Ma, Z., Ji, R., Wang, Y., Hong, X.: Boosting crowd counting via multifaceted attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19628–19637 (2022)

[66] Hu, Y., Jiang, X., Liu, X., Zhang, B., Han, J., Cao, X., Doermann, D.: Nas-count: Counting-by-density with neural architecture search. In: European Conference on Computer Vision, pp. 747–766 (2020). Springer

[67] Cheng, Z.-Q., Dai, Q., Li, H., Song, J., Wu, X., Hauptmann, A.G.: Rethinking spatial invariance of convolutional networks for object counting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19638–19648 (2022)

[68] Shu, W., Wan, J., Tan, K.C., Kwong, S., Chan, A.B.: Crowd counting in the frequency domain. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19618–19627 (2022)

[69] Boominathan, L., Kruthiventi, S.S., Babu, R.V.: Crowdnet: A deep convolutional network for dense crowd counting. In: Proceedings of the 24th ACM International Conference on Multimedia, pp. 640–644 (2016)

[70] Onoro-Rubio, D., López-Sastre, R.J.: Towards perspective-free object counting with deep learning. In: European Conference on Computer Vision, pp. 615–629 (2016). Springer

[71] Babu Sam, D., Surya, S., Venkatesh Babu, R.: Switching convolutional neural network for crowd counting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5744–5752 (2017)

[72] Jiang, X., Xiao, Z., Zhang, B., Zhen, X., Cao, X., Doermann, D., Shao, L.: Crowd counting and density estimation by trellis encoder-decoder networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6133–6142 (2019)

[73] Liu, W., Salzmann, M., Fua, P.: Context-aware crowd counting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5099–5108 (2019)

[74] Sam, D.B., Babu, R.V.: Top-down feedback for crowd counting convolutional neural network. In: Thirty-second AAAI Conference on Artificial Intelligence (2018)

[75] Sindagi, V.A., Patel, V.M.: Multi-level bottom-top and top-bottom feature fusion for crowd counting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1002–1012 (2019)

[76] Liu, N., Long, Y., Zou, C., Niu, Q., Pan, L., Wu, H.: Adcrowdnet: An attention-injective deformable convolutional network for crowd understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3225–3234 (2019)

[77] Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 764–773 (2017)

[78] Shi, M., Yang, Z., Xu, C., Chen, Q.: Revisiting perspective information for efficient crowd counting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7279–7288 (2019)

[79] Yang, Y., Li, G., Wu, Z., Su, L., Huang, Q., Sebe, N.: Reverse perspective network for perspective-aware object counting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4374–4383 (2020)

[80] Zhang, A., Shen, J., Xiao, Z., Zhu, F., Zhen, X., Cao, X., Shao, L.: Relational attention network for crowd counting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6788–6797 (2019)

[81] Chen, B., Yan, Z., Li, K., Li, P., Wang, B., Zuo, W., Zhang, L.: Variational attention: Propagating domain-specific knowledge for multi-domain learning in crowd counting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 16065–16075 (2021)

[82] Liu, J., Gao, C., Meng, D., Hauptmann, A.G.: Decidenet: Counting varying density crowds through attention guided detection and density estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5197–5206 (2018)

[83] Zhang, A., Yue, L., Shen, J., Zhu, F., Zhen, X., Cao, X., Shao, L.: Attentional neural fields for crowd counting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5714–5723 (2019)

[84] Liu, C., Weng, X., Mu, Y.: Recurrent attentive zooming for joint crowd counting and precise localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1217–1226 (2019)

[85] Jiang, X., Zhang, L., Xu, M., Zhang, T., Lv, P., Zhou, B., Yang, X., Pang, Y.: Attention scaling for crowd counting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,

pp. 4706–4715 (2020)

[86] Wan, J., Liu, Z., Chan, A.B.: A generalized loss function for crowd counting and localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1974–1983 (2021)

[87] Wang, B., Liu, H., Samaras, D., Nguyen, M.H.: Distribution matching for crowd counting. Advances in neural information processing systems **33**, 1595–1607 (2020)

[88] Zhao, M., Zhang, J., Zhang, C., Zhang, W.: Leveraging heterogeneous auxiliary tasks to assist crowd counting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12736–12745 (2019)

[89] Shi, Z., Mettes, P., Snoek, C.G.: Counting with focus for free. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4200–4209 (2019)

[90] Laradji, I.H., Rostamzadeh, N., Pinheiro, P.O., Vazquez, D., Schmidt, M.: Where are the blobs: Counting by localization with point supervision. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 547–562 (2018)

[91] Ma, Z., Wei, X., Hong, X., Gong, Y.: Bayesian loss for crowd count estimation with point supervision. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6142–6151 (2019)

[92] Liu, Y., Shi, M., Zhao, Q., Wang, X.: Point in, box out: Beyond counting persons in crowds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6469–6478 (2019)

[93] Song, Q., Wang, C., Jiang, Z., Wang, Y., Tai, Y., Wang, C., Li, J., Huang, F., Wu, Y.: Rethinking counting and localization in crowds: A purely point-based framework. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3365–3374 (2021)

[94] Choe, J., Oh, S.J., Lee, S., Chun, S., Akata, Z., Shim, H.: Evaluating weakly supervised object localization methods right. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3133–3142 (2020)

[95] Ranjan, V., Sharma, U., Nguyen, T., Hoai, M.: Learning to count everything. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3394–3403 (2021)

[96] Shi, M., Lu, H., Feng, C., Liu, C., Cao, Z.: Represent, compare, and

learn: A similarity-aware framework for class-agnostic counting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9529–9538 (2022)

[97] Gong, S., Zhang, S., Yang, J., Dai, D., Schiele, B.: Class-agnostic object counting robust to intraclass diversity. In: European Conference on Computer Vision, pp. 388–403 (2022). Springer

[98] Nguyen, T., Pham, C., Nguyen, K., Hoai, M.: Few-shot object counting and detection. In: European Conference on Computer Vision, pp. 348–365 (2022). Springer

[99] Zhang, C., Li, H., Wang, X., Yang, X.: Cross-scene crowd counting via deep convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 833–841 (2015)

[100] Ma, Z., Hong, X., Wei, X., Qiu, Y., Gong, Y.: Towards a universal model for cross-dataset crowd counting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3205–3214 (2021)

[101] Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3234–3243 (2016)

[102] Gaidon, A., Wang, Q., Cabon, Y., Vig, E.: Virtual worlds as proxy for multi-object tracking analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4340–4349 (2016)

[103] Liu, W., Durasov, N., Fua, P.: Leveraging self-supervision for cross-domain crowd counting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5341–5352 (2022)

[104] Gong, S., Zhang, S., Yang, J., Dai, D., Schiele, B.: Bi-level alignment for cross-domain crowd counting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7542–7550 (2022)

[105] Change Loy, C., Gong, S., Xiang, T.: From semi-supervised to transfer counting of crowds. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2256–2263 (2013)

[106] Zhao, Z., Shi, M., Zhao, X., Li, L.: Active crowd counting with limited supervision. In: European Conference on Computer Vision, pp. 565–581 (2020). Springer

[107] Liu, X., Van De Weijer, J., Bagdanov, A.D.: Leveraging unlabeled data for crowd counting by learning to rank. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7661–7669 (2018)

[108] Sam, D.B., Sajjan, N.N., Maurya, H., Babu, R.V.: Almost unsupervised learning for dense crowd counting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 8868–8875 (2019)

[109] Liu, Y., Liu, L., Wang, P., Zhang, P., Lei, Y.: Semi-supervised crowd counting via self-training on surrogate tasks. In: European Conference on Computer Vision, pp. 242–259 (2020). Springer

[110] Sindagi, V.A., Yasarla, R., Babu, D.S., Babu, R.V., Patel, V.M.: Learning to count in the crowd from limited labeled data. In: European Conference on Computer Vision, pp. 212–229 (2020). Springer

[111] Meng, Y., Zhang, H., Zhao, Y., Yang, X., Qian, X., Huang, X., Zheng, Y.: Spatial uncertainty-aware semi-supervised crowd counting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 15549–15559 (2021)

[112] Lei, Y., Liu, Y., Zhang, P., Liu, L.: Towards using count-level weak supervision for crowd counting. Pattern Recognition **109**, 107616 (2021)

[113] Xu, Y., Zhong, Z., Lian, D., Li, J., Li, Z., Xu, X., Gao, S.: Crowd counting with partial annotations in an image. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 15570–15579 (2021)

[114] Li, H., Pan, X., Yan, K., Tang, F., Zheng, W.-S.: Siod: Single instance annotated per category per image for object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 14197–14206 (2022)

[115] Borstel, M.v., Kandemir, M., Schmidt, P., Rao, M.K., Rajamani, K., Hamprecht, F.A.: Gaussian process density counting from weak supervision. In: European Conference on Computer Vision, pp. 365–380 (2016). Springer

[116] Yang, Y., Li, G., Wu, Z., Su, L., Huang, Q., Sebe, N.: Weakly-supervised crowd counting learns from sorting rather than locations. In: European Conference on Computer Vision, pp. 1–17 (2020). Springer

[117] Babu Sam, D., Agarwalla, A., Joseph, J., Sindagi, V.A., Babu, R.V., Patel, V.M.: Completely self-supervised crowd counting via distribution matching. In: European Conference on Computer Vision, pp. 186–204 (2022). Springer

[118] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10684–10695 (2022)

[119] Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.-Y., Ermon, S.: SDEdit: Guided image synthesis and editing with stochastic differential equations. In: International Conference on Learning Representations (2022). https://openreview.net/forum?id=aBsCjcPu_tE

[120] Greff, K., Belletti, F., Beyer, L., Doersch, C., Du, Y., Duckworth, D., Fleet, D.J., Gnanapragasam, D., Golemo, F., Herrmann, C., Kipf, T., Kundu, A., Lagun, D., Laradji, I., Liu, H.-T.D., Meyer, H., Miao, Y., Nowrouzezahrai, D., Oztireli, C., Pot, E., Radwan, N., Rebain, D., Sabour, S., Sajjadi, M.S.M., Sela, M., Sitzmann, V., Stone, A., Sun, D., Vora, S., Wang, Z., Wu, T., Yi, K.M., Zhong, F., Tagliasacchi, A.: Kubric: A scalable dataset generator (2021)

[121] Cholakkal, H., Sun, G., Khan, F.S., Shao, L.: Object counting and instance segmentation with image-level supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)

[122] Sajid, U., Chen, X., Sajid, H., Kim, T., Wang, G.: Audio-visual transformer based crowd counting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, pp. 2249–2259 (2021)

[123] Liu, L., Chen, J., Wu, H., Li, G., Li, C., Lin, L.: Cross-modal collaborative representation learning and a large-scale rgbt benchmark for crowd counting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4823–4833 (2021)

[124] Peng, T., Li, Q., Zhu, P.: Rgb-t crowd counting from drone: A benchmark and mmccn network. In: Proceedings of the Asian Conference on Computer Vision (ACCV) (2020)

[125] Lian, D., Li, J., Zheng, J., Luo, W., Gao, S.: Density map regression guided detection network for rgb-d crowd counting and localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)

[126] Botta, F., Moat, H.S., Preis, T.: Quantifying crowd size with mobile phone and twitter data. Royal Society open science **2**(5), 150162 (2015)

[127] Bastiaanssen, W.G., Ali, S.: A new crop yield forecasting model based on satellite measurements applied across the indus basin, pakistan.

Agriculture, ecosystems & environment **94**(3), 321–340 (2003)