Counting in The Wild

Carlos Arteta[1], Victor Lempitsky[2], and Andrew Zisserman[1]

[1]Department of Engineering Science, University of Oxford, UK
[2]Skolkovo Institute of Science and Technology (Skoltech), Russia

Abstract. In this paper we explore the scenario of learning to count multiple instances of objects from images that have been dot-annotated through crowdsourcing. Specifically, we work with a large and challenging image dataset of penguins in the wild, for which tens of thousands of volunteer annotators have placed dots on instances of penguins in tens of thousands of images. The dataset, introduced and released with this paper, shows such a high-degree of object occlusion and scale variation that individual object detection or simple counting-density estimation is not able to estimate the bird counts reliably.
To address the challenging counting task, we augment and interleave density estimation with foreground-background segmentation and explicit local uncertainty estimation. The three tasks are solved jointly by a new deep multi-task architecture. Using this multi-task learning, we show that the spread between the annotators can provide hints about local object scale and aid the foreground-background segmentation, which can then be used to set a better target density for learning density prediction. Considerable improvements in counting accuracy over a single-task density estimation approach are observed in our experiments.

1 Introduction

This paper is motivated by the need to address a challenging large-scale real-world image-based counting problem that cannot be tackled well with existing approaches. This counting task arises in the course of ecological surveys of Antarctic penguins, and the images are automatically collected by a set of fixed cameras placed in Antarctica with the intention of monitoring the penguin population of the continent. The visual understanding of the collected images is compounded by many factors such as the variability of vantage points of the cameras, large variation of penguin scales, adversarial weather conditions in many images, high similarity of the appearance between the birds and some elements in the background (e.g. rocks), and extreme crowding and inter-occlusion between penguins (Figure 1).

The still ongoing annotation process of the dataset consists of a public website [27], where non-professional volunteers annotate images by placing dots on top of individual penguins; this is similar to citizen science annotators, who have also been used as an alternative to paid annotators for vision datasets (e.g. [19]). The simplest form of annotation (dotting) was chosen to scale up the annotation process as much as possible. Based on the large number of dot-annotated images, our goal is to train a deep model that can solve the counting task through density regression [4, 6, 8, 16, 25, 26].

Compared to the training annotations used in previous works on density-based counting, our crowd-sourced annotations show abundant errors and contradictions between annotators. We therefore need to build models that can learn in the presence of noisy labels. Perhaps, an even bigger challenge than annotation noise, is the fact that dot annotations do not directly capture information about the characteristic object scale, which varies wildly in the dataset (the diameter of a penguin varies between ∼15 and ∼700 pixels). This is in contrast to previous density estimation methods that also worked with (less noisy) dot annotations but assumed that the object scale was either constant or could be inferred from a given ground plane estimate.

To address the challenges discussed above, we propose a new approach for learning to count that extends the previous approaches in several ways. Our first extension over density-based methods is the incorporation of an explicit foreground-background segmentation into the learning process. We found that when using noisy dot annotations, it is much easier to train a deep network for foreground-background segmentation than for density prediction. The key insight is that once such a segmentation network is learned, the predicted foreground masks can be used to form a better target density function for learning the density prediction.

Also, the density estimates predicted for new images can be further combined with the foreground segmentation, e.g. by setting the density in the background regions to zero.

Our second extension is to take advantage of the availability of multiple annotations in two ways. First, by exploiting the *spatial* variations across the annotations, we obtain cues towards the scale of the objects. Second, by exploiting also their *counting* variability, we add explicit prediction of the annotation difficulty into our model. Algorithmically, while it is possible to learn the networks for segmentation and density estimation in sequence, we perform *joint* fine-tuning of the three components corresponding to object-density prediction, foreground-background segmentation, and local uncertainty estimation, using a deep multi-task network.

This new architecture enables us to tackle the very hard counting problem at hand. Careful analysis suggests that the proposed model significantly improves in counting accuracy over a baseline density-based counting approach, and obtains comparable accuracy to the case when the depth information is available.

2 Background

Counting objects in crowded scenes. Parsing crowded scenes, such as the common example of monitoring crowds in surveillance videos, is a challenging task mainly due to the occlusion between instances in the crowd, which cannot be properly resolved by traditional binary object detectors. As a consequence, models emerged which cast the problem as one where image features were mapped into a global object count [2, 3, 7], or local features mapped into pixel-wise object densities [4, 6, 8, 16, 24, 26] which can be integrated into the object count over

any image region. In either case, these approaches provided a way to obtain an object count while avoiding detecting the individual instances. Moreover, if the density map is good enough, it has been shown that it can be used to provide an estimate for the localization of the object instances [1, 11]. The task in this work is in practice very similar to the pixel-wise object density estimation from local features, and also executed using convolutions neural networks (CNN) similar to [16, 24, 26]. However, aside from the main differences in the underlying statistical annotation model, our model differs from previous density learning methods in that we use a CNN architecture mainly designed for the segmentation task, in which the segmentation mask is used to aid the regression of the density map. Our experiments demonstrate the importance of such aid.

Learning from multiple annotators. The increasing amount of available data has been a key factor in the recent rapid progress of the learning-based computer vision. While the data collection can be easily automated, the bottleneck in terms of cost and effort mostly resides in the data annotation process. Two complementary strategies help the community to alleviate this problem: the use of crowds for data annotation (e.g. through crowdsourcing platforms such as Amazon Mechanical Turk); and the reduction in the level of difficulty of such annotations (e.g. image-level annotations instead of bounding boxes). Indeed, both solutions create in turn additional challenges for the learning models. For example, crowdsourced annotations usually show abundant errors, which create the necessity of building models that can learn in the presence of noisy labels. Similarly, dealing with simpler annotations demands more complex models, such as in learning to segment from image-level labels instead of pixel-level annotations, where the model also needs to infer on its own the difference between the object and the background. Nevertheless, regardless of the added complexity, coping with simpler and/or noisy supervision while taking advantage of vast amounts of data is a scalable approach.

Dealing with multiple annotators has been generally approached by modelling different annotation variables with the objective of scoring and weighting the influence of each of the annotators [12, 22, 23], and finding the ground-truth label that is assumed to exist in the consensus of the annotators [10, 14, 21]. However, in cases such as the penguin dataset studied in this paper, most of the annotations are performed by tens of thousands of different and mostly anonymous users, each of which provides a very small set of annotations, thus reducing the usefulness of modelling the reliabilities of individual annotators. Moreover, ambiguous examples are extremely common in such crowded and occluded scenes, which not only means that it is often not possible to agree on a ground-truth, but also that the errors of the individual annotators, most notably missing instances in the counting case, can be so high that a ground-truth cannot be determined from the annotations alone as all of them are far from it. On the positive side, the variability between annotators is proportional to the image difficulty, thus we chose to learn to predict directly the uncertainty or agreement of the annotators, and not only the most likely instance count. Therefore, we argue that providing a confidence band for the object count still fulfils

the objective of the counting task, taking advantage of the multiple annotators. We note that this predictive uncertainty is different from the uncertainty in the model parameters, which could also be determined from a learned architecture similar to the one used in this work (e.g. [5]), but that is not used here. Instead, the approach taken in this paper is more similar to [23] where uncertainty of the annotator is directly used in the learning model, although it is determined by the annotator recording their uncertainty in the annotation system, as opposed to deriving it from the disagreement between annotators.

Learning from dot-annotations. Dot annotations are an easy way to label images, but generally require additional cues in order to be used in learning complex tasks. For example, [15] showed how to use dots in combination with an objectness prior in order to learn segmentations from images that would otherwise only have image-level labels. Dots have also been used in the context of interactive segmentation [18, 20] with cues such as background annotations, which are easy to provide in an interactive context. The most common task in which dot-annotations are used is that of counting [1, 4, 8, 25], where they are used in combination with direct information about the spatial extent of the object in order to define object density maps that can be regressed. However, in all of these cases the dots are introduced by a single annotator. We show that when dot annotations are crowdsourced, and several annotators label each image, the required spatial cues can be obtained from the point patterns, which can then be used for object density estimation or segmentation.

3 The penguin dataset

The penguin dataset [13] is a product of an ongoing project for monitoring the penguin population in Antarctica. The images are collected by a series of fixed cameras in over 40 different sites, which collect images every hour. Examples can be seen in Figure 1. The data collection process has been running for over three years, and has produced over 500 thousand images with resolutions between 1MP and 6MP. The image resolution in combination with the camera shots, translate into penguin sizes ranging from under 15 pixels to over 700 pixels in length.

Among the information that the zoologists wish to extract from these data, a key piece is the trend in the size of the population of penguins on each site, which can then be studied for correlation with different factors such as climate change. Therefore, it is necessary to obtain the number of penguins present in each of the frames. The goal of making this dataset available to the vision community is to contribute to the development of a framework that can accurately parse this continuous stream of images.

So far, the annotation process of the dataset has been carried out by human volunteers in a citizen science website [27], where any person can enter to place dots inside the penguins appearing in the image. Currently, the annotation tool has received over 35 thousand different volunteers. Once an image has been annotated by twenty volunteers, it is removed from the annotation site.

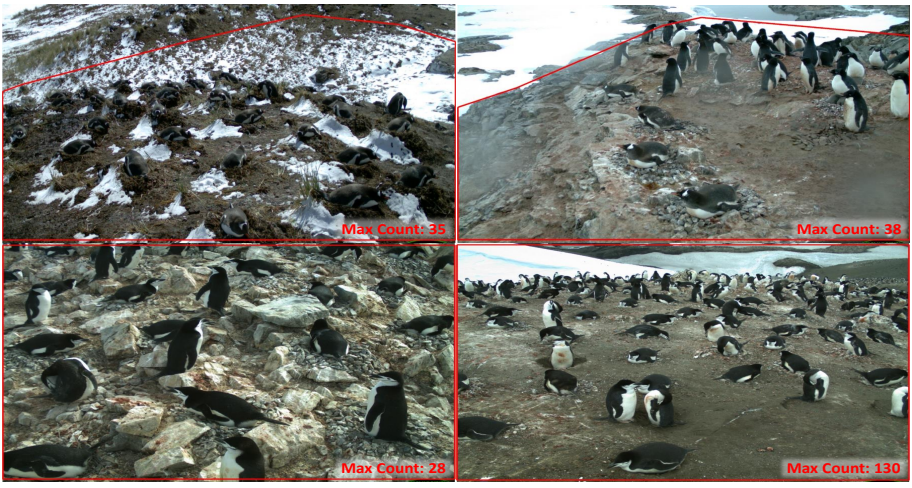

Fig. 1: *(Example images of the penguin dataset).* The challenging penguin dataset consistently shows heavy occlusion and complex background patterns that can be easily mistaken with the penguins. Aside from the difficult image conditions, the dataset is only annotated with dots. Regions of interest are provided for each site, shown in this figure with red lines. We also show in the bottom right of each image the maximum penguin count provided in the crowdsourced annotations.

The distribution of annotation-based count around the ground-truth is far from Gaussian normal. Instead, as the level of difficulty in the image regions increases, the annotators proportionally under-count (i.e. false negatives are far more frequent than false-positives). This becomes evident after experiencing the annotation process. In general, it is much easier for a human to miss an instance than to confuse it with something else. Furthermore, cluttered images (e.g. with over 30 instances) make the annotators tired, making them less careful, and thus, more prone to missing instances. This fact motivates the design of the learning target described in Section 4, as well as the evaluation metrics discussed in Section 5.

Each of the sites in the penguin dataset has different properties, which result in different levels of difficulty. For example, some cameras are placed to capture very wide shots, where masses of penguins appear in very low effective resolution. Other cameras are placed in such a way that the perspective creates constant occlusion. Factors external to the cameras, such as the weather on site, also represent difficulty factors that must be dealt with. In order to allow a more detailed evaluation of this and future methods, we have split the different sites into four categories according to their difficulty: lower-crowded, medium/lower-crowded, medium/higher-crowded and higher-crowded. Additionally, a region of interest is provided for each of the cameras which aims to discard far-away regions.

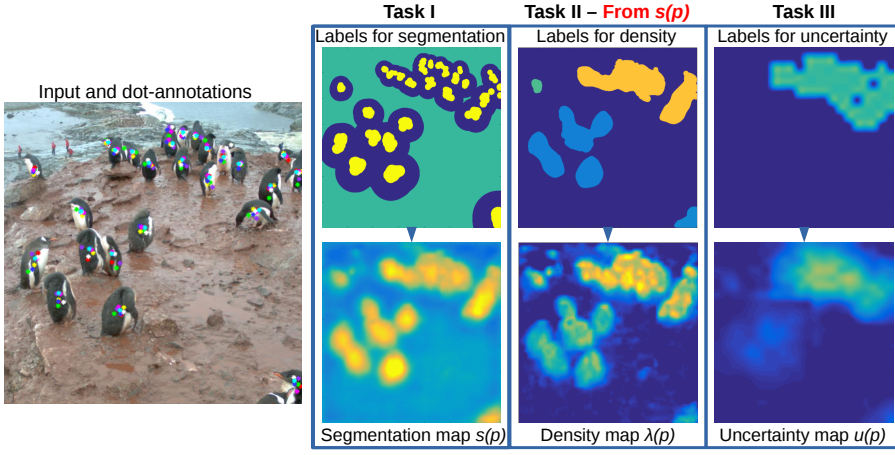

Fig. 2: *(Sketch of the training procedure in the multi-task network).* The proposed solution for the counting problem from crowdsourced dot-annotations consist of a convolutional network that predicts three output maps: segmentation $s(p)$, object density $\lambda(p)$, and a map $u(p)$ representing the agreement between the multiple annotators. The labels for the three tasks are generated using the dot patterns introduced by the annotators, as described in Section 4. Particularly, we note that the shape of the target label used to regress the object density map is defined using the segmentation map $s(p)$ as detailed in the text. Using the segmentation map to generate more spatially accurate labels for the density regression is a key element in out method. The segmentation, on the other hand, can be learned from less accurate labels (i.e. the trimaps).

4 Learning model

Our aim here is to train a ConvNet to estimate the density function $\lambda(p)$ on a novel image. If the learning has been successful, integrating over any region of the function $\lambda(p)$ corresponding to an image $\mathcal{I}$(p) will return the estimated number of instances of the object in such regions. Also, the prediction of the agreement map $u(p)$ can be used in a novel image to estimate how much multiple annotators would agree in the number of instances in any region of such image if they were to annotate it – which is also an indication of the image/region difficulty, and provides a type of confidence band for the values of $\lambda(p)$.

As discussed in Section 2, regressing the object density function from dot-annotations requires additional knowledge of the extent of each instance throughout the entire image in order to define the target density. Therefore, when the camera perspective significantly affects the relative size of the objects, it is necessary to either have a depth map along with some additional cue of area covered by the object, or bounding box annotations for each instance. In this paper we present an alternative approach to defining the target density map by using the

object (foreground) segmentation, as this can be an easier task to learn with less supervision than the density regression. We present such an approach with and without any depth information, preceded by an overview of the general learning architecture.

Learning architecture. The learning architecture is a multi-task convolutional neural network which aims to produce *(i)* a foreground/background **s**egmentation $s(p)$, *(ii)* an object density function $\lambda(p)$, and *(iii)* a prediction of the agreement between the annotators $u(p)$ as a measure of **u**ncertainty. While the usual motivation for the use of multi-task architectures is the improvement in the generalization, here we additionally reuse the predicted segmentations to change the objective for other branches as learning progresses.

The segmentation branch consists of a pixel-wise binary classification path with a hinge loss. For the second task of regressing $\lambda(p)$, where more precise pixel-wise values are required, we use the segmentation mask $s(p)$ from the first task as a prior. That is, the target map for learning $\lambda(p)$ is constructed from an approximation $\hat{\lambda}(p)$ that is built using the class segmentation $s(p)$. The density target map is regressed with a root-mean-square loss. Finally, the same regression loss function is used in the third and final branch of the CNN in order to predict a map $u(p)$ of agreement between the annotators, as described below.

Labels for learning. A fundamental aspect of this framework is the way the labels are defined for the different learning tasks based on the multiple dot annotations. The details will depend on the specific model used, described later in Section 4.1 and Section 4.2, but we fist introduce the general aspects of them. Given a set of dots $D = d_1, d_2, \ldots, d_K$, we define a trimap $t(p)$ of 'positive', 'negative' and 'ignore' regions, which respectively are likely to correspond to regions mostly contained inside instances of the object, regions corresponding to background, and uncertain regions in between. Example trimaps are shown in Figure 3.

Regression targets. A key aspect is defining the regression target for each task as this in turn defines the pixel-wise loss. For the *segmentation map* target, the positive and negative regions in the trimap are used to define the foreground and background pixel labels, whereas the ignore regions do not contribute in the computation of the pixel classification loss (i.e. the derivatives of the loss in those spatial locations are set to zero for the backpropagation through the CNN). As the network learns to regress this target, the predicted foreground regions can extend beyond the positives of the trimap into the ignore regions to better match the true foreground/background segmentation, as can be seen in Figure 2.

The *density map* target is obtained from the predicted segmentation and the user annotations. First, connected components are obtained from the predicted segmentation. Then, for each connected component, an integer score is assigned as the maximum over the different annotators. We pick the maximum as a way to counter-balance the consistent under-estimation of the count (e.g. as opposed to the mean) as discussed in Section 3. The density target is defined for each pixel of the connected component by assigning it the integer score divided by

the component area (so that integrating the density target over the component area gives the maximum annotation).

Finally, the *uncertainty map* target for annotator (dis)agreement consist of the variance of the annotations within each of the connected component regions. More principled ways of handling the annotation bias along with the uncertainty are briefly discussed in Section 6 (applicable to crowdsourced dot-annotations in general), but we initially settle for the more practical MAX and VAR approaches described above.

Implementation details. The core of the CNN is the segmentation architecture *FCN8s* presented in [9], which is initialized from the *VGG-16* [17] classification network, and adds skip and fusion layers for a finer prediction map which can be evaluated at the scale of the input image.

We make extensive use of scaling-based data augmentation while training the ConvNet by up-scaling each image to six different scales and taking random crops of 700×700 pixels, our standard input size while training. This is done with the intention of gaining the scale invariance required in the counting task (i.e. the spatial region in the density map corresponding to a single penguin should sum to one independently of its size in pixels).

We train for the three tasks in parallel and end-to-end. The overall weight of the segmentation loss is set to be higher than the remaining two losses as we want this easier task to have more influence over the filters learned; we found that this helps to avoid the divergence of the learning that could happen during the iterations where the segmentation prior is far from local optima. At the start of the training it is necessary to provide an initial target for the density map loss, since the segmentation map $s(p)$ is not yet defined. Again the trimap is used, but more loosely here than in the segmentation target, with the union of the positive *and* ignore maps used to define the connected components. The density is then obtained by assigning annotations to the connected components in the same manner as used during training. At the end of this initialization the density target will generally spread beyond the objects since it includes the ignore region. The initial trimap can be estimated in two different ways depending on whether the rough estimate of depth information is available. We now discuss these two cases.

4.1 Learning from multiple dot-annotations and depth information

We wish to use the dot-annotations provided by multiple annotators for an image to generate a trimap $t(p)$ for that image. The trimap will be used for the intermediate learning step of a segmentation mask $s(p)$.

Due to perspective effects the penguins further from the camera are smaller, and this typically means that penguins become smaller moving from the bottom to the top of the image for the camera placements used in our dataset. We assume here that we have a depth map for the scene, together with an estimate of the object class size (e.g. penguins are roughly of a similar real size), and thus can predict the size of a penguin at any point in the image.

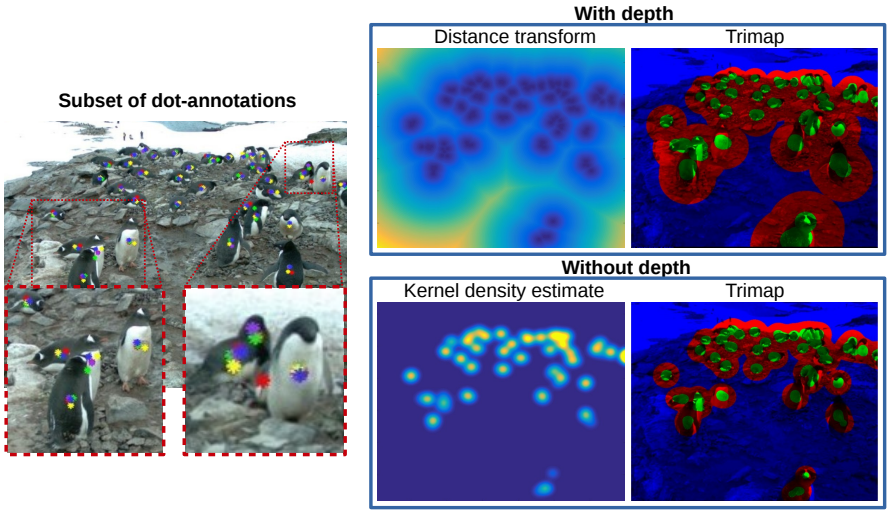

Fig. 3: *(Generation of the trimaps for the training labels).* The left image shows an example of penguin dataset annotations from which we generate the trimaps that are used during training, as described in Section 4; the dots are color-coded according to each annotator, and we only plot a small portion of the annotations to avoid further cluttering. Our training labels are generated with the help of trimaps, which can be obtained with and without the use of depth information by exploiting the multiple annotations and the randomness in them. On the right, the trimap obtained in each way is used to shade the input image in green, red and blue, corresponding to the positive, ignore, and negative regions of the trimap.

The trimap is then formed using a simple computation: first, a distance transform is computed from all dot annotations, such that the value at any pixel is the distance to the nearest dot annotation. Then the trimap positive, negative and ignore regions are obtained by thresholding the distance transform based on the predicted object size. For example, pixels that are further from a dot annotation than three quarters of the object size are negative. An example trimap with depth information is shown in Figure 3(top).

4.2 Learning from multiple dot-annotations without depth

In the case of not having an estimate for the varying size of the penguins in an image, we need a depth-independent method for defining the trimap.

Learning from multiple crowdsourced dot annotations without a direct indication of the spatial extent of the instances can be enabled by leveraging the variability in the annotators placement of dots. As one might expect, annotators have different ideas of where to place dots when annotating, which along with the spatial randomness of the process, can provide a sufficient cue into the spa-

tial distribution of each instance. The more annotators are available, the better the spatial cue can get.

We harness this spatial distribution by converting it into a density function $\rho(p)$, and then thresholding $\rho(p)$ at two levels to obtain the positive, ignore and negative regions of the trimap $t(p)$. The density is simply computed by placing a Gaussian kernel with bandwidth h at each provided dot annotation: $\rho(p) = \sum_{j=1}^{N} \frac{1}{h} K(\frac{p-d_j}{h})$ where $d_1 ... d_N$ is the set of provided points. We note that this can be seen as a generalization of the approach for generating the target density map used in previous counting work for the case of a single annotator and a Gaussian kernel [1].

The only question remaining is how to determine the size of the Gaussian kernel h. We rely on a simple heuristic to extract from the dot-patterns a cue for the selection of h: annotations on a larger instance tend to be more distributed than annotations on smaller objects. In fact, the relation between point pattern distribution and object size is not a clear one as it is affected by other factors such as occlusion, but it is sufficient for our definition. The estimation of h consists of doing a rough reconciliation of the dot patterns from multiple annotators (to determine which dots should be assigned to the same penguin), followed by the computation of a single value of h that suits an entire image. The reconciliation process is done by matching the dots between pairs of annotators using the Hungarian algorithm, with a matching cost given by Euclidean distance. This produces a distribution of distances between dots that are likely to belong to the same instance. After combining all pairs of annotators, h is then taken to be the median of this distribution of distances. An illustration of $\rho(p)$ can be seen in Figure 3(bottom) using a Gaussian kernel, which we keep for our experiments of Section 5.

Finally, the trimap $t(p)$ is obtained from $\rho(p)$ by thresholding as above. As can be seen in Figure 3(bottom), this approach has less information than using depth and results in slightly worse trimaps (i.e. with more misplaced pixels), which in our experiments translate to slower convergence of the learning.

5 Experiments

Metrics for counts from crowdsourced dot-annotations. As discussed in Section 3, benchmarking on the penguin dataset is a challenging task due to the lack of ground-truth. Moreover, it is a common case in the penguin dataset that the true count, under any reasonable definition, might lie far from what the annotators have indicated, and is generally an under-estimation. Therefore, we propose to evaluate the performance on this dataset using metrics that not only reflect the similarity of the automatic estimations w.r.t. what the annotators introduced, but also the uncertainty in them; ultimately, both aspects are useful information regarding the image.

Considering the under-counting bias of the annotators, we firstly propose to compare with a region-wise max of the annotations. That is, we first define a set of "positive" regions based on the dot-annotations, as done in the learning

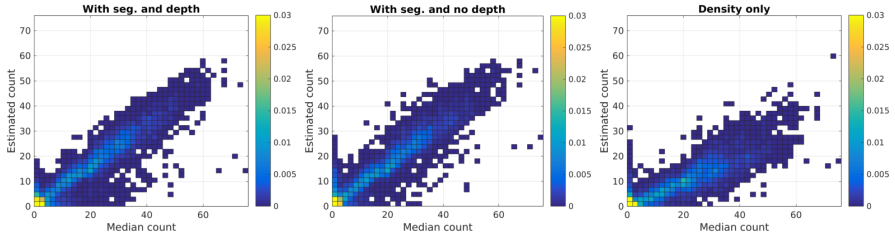

Fig. 4: (*(Histogram of coincidence).* The figure presents a 40-by-40 normalized histogram of counts accumulating the predicted values from the proposed methods, w.r.t. median count of the annotators in the *mixed-sites* dataset split. Even though the reference is noisy, it is visible that the proposed methods using the segmentation-aided count show a tighter agreement with the annotators. The two main failure modes of the methods are visible in this plot, as detailed in Section 5.

label generation with depth information described in Section 4.1. Then, for each connected component, we define the annotated density as one which integrates to the maximum over what each of the annotators introduced. Different from an image-wise maximum, the region-based evaluation approach allows for the possibility of the annotations being complementary, thus reducing the overall under-counting bias. Additionally, we present the annotated values using the median instead of the maximum for comparison.

Penguin counting experiments. We now compare the two learning models proposed in Section 4 with the human annotations in the task of counting, according to the metrics described above. As a first attempt to propose a solution applicable to the penguin dataset, we work in this paper with the lower-crowded and medium/lower-crowded sites of the penguin dataset, which add up to ∼82k images. We split these images into training and testing sets in two different ways, which reflect two similarly valuable use-cases: the *mixed-sites* split, in which images from the same camera can appear in both the training and testing set, and the *separated-sites* split, in which images in each set strictly belong to different cameras. In both cases, the size of the training and testing sets account for ∼70% and ∼30% of the ∼82k images respectively.

To the best of our knowledge, this the first work to address the problem of counting from crowdsourced dot-annotations, and the penguin dataset is the first one suitable for this task, thus there is no method for direct comparison. We expect this would change after the introduction of the penguin dataset. In the meantime, we propose a simple baseline that extends the case of previous counting work such as [1, 4, 8, 25], where a single set of dot-annotations was available, along with an estimate of the object size. For this *density-only* baseline, we generate a target density map for regression using a kernel density estimate (similar to Section 4.2) but define the bandwidth of the kernel using the depth information instead of the heuristic of inter-dot distances. Then, the target map is regressed directly without the help of the segmentation prediction.

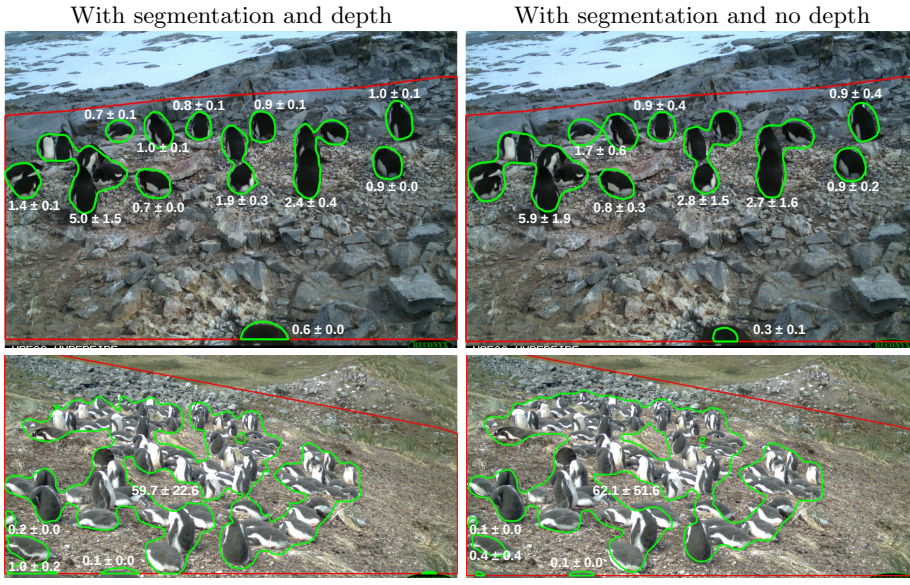

Fig. 5: *(Example results on the penguin dataset)*. The segmentation-aided counting methods proposed outputs a segmentation of the region along with the instance count and an estimation of the the annotators agreement given as range in the count. In these examples both proposed methods generate similar outputs. The top row is a relatively easy example, while the bottom row present severe occlusion in the very crowded area, and thus, it is segmented as a single large region with a very wide uncertainty band. The region of interest for this image, as annotated by the penguinologist collecting the data, is shown with a red line.

The quantitative comparison between the approaches is shown in Table 1. We observed that the proposed methods, mainly differing in the usage of an auxiliary segmentation network, produce lower counting errors than the baseline in our metrics and different data splits, while being very similar in performance between them. The latter would indicate that the density prediction can be done with similar results without requiring explicit object size information during training. The quantitative difference in performance between methods can be better detailed in Figure 4. As Table 1 indicated, the performance of the proposed methods is similar, both showing good agreement with the region-wise median count of the annotators. Figure 4 also reveals two *failure modes* present in the experiments. The first is reflected in the mass accumulated under the diagonal, meaning that the examples contain a considerable number of instances that were missed. This failure mode correspond to images containing instances smaller than those the network is able to capture (e.g. penguins with a length of $\sim$15 pixels). The second failure mode is one that mainly affects the density-only baseline network, and it is visible as the mass accumulated above the diagonal

	MCE-Median	MCE-Max
Density-only baseline	7.09/5.01	9.81/8.11
With seg. and depth	3.42/3.99	5.74/6.38
With seg. and *no* depth	3.26/3.41	5.35/5.77

Table 1: *Comparison of counting experiments on the penguin dataset.* We compare the counting accuracy of the proposed counting methods and baseline against the count of the annotators based on two single-value criteria described in Section 5: the mean counting error w.r.t. the median (MCE-Median) and maximum (MCE-Max) of the annotations. Results are shown for two splits of the dataset: mixed and separated sites, presented as *mixed/separated*, and also described in Section 5. We observe that the segmentation-aided counting methods fall closer to the reference than the density-only baseline in all metrics, whereas the two segmentation-aided methods are comparable between them.

	0 Pen.	1-10 Pen.	11-20 Pen.	21-30 Pen.	31-113 Pen.
Density-only baseline	0.89/0.73	2.92/3.30	9.69/13.02	14.72/20.81	24.45/34.24
With seg. and depth	0.93/0.57	2.11/2.80	5.23/10.83	7.89/18.15	14.21/26.00
With seg. and *no* depth	1.41/0.46	2.17/2.68	4.81/10.20	7.12/16.56	12.54/23.24

Table 2: *Counting performance as a function of penguin density.* We show a breakdown of the results presented in Table 1 (MCE-Max metric) w.r.t. the density of penguins in the images. As expected, the counting accuracy decreases with the increase in the number of penguins in the images. Note, however, that the accuracy in the annotations is affected in the same way, and thus, the comparison becomes less reliable for crowded images. The two results shown in each cell correspond to the *mixed-* and *separated-* sites splits of the dataset.

and near to the y-axis. This mode corresponds to the cases where the network was not able to differentiate between complex background (e.g. mostly rocks) and the penguins, thus erroneously counting instances. We hypothesize that the discrimination capacity brought by the segmentation loss helps the other networks to reduce or suppress this effect.

To further examine the methods, we show in Table 2 a breakdown of their errors as a function of the number of penguins in the testing images. One expected observation is that the error for all methods grows with the penguin density in the images. However, we must consider that the annotation error is also greatly affected by such factor. It is also noticeable the influence of the first failure mode discussed above. The density-only baseline is more sensitive to this problem due to not having the discriminative power of the foreground-background segmentation. Therefore, it has a less favourable trade-off between error rates throughout the density spectrum than the methods relying on the segmentation mask.

Figure 5 shows example qualitative results on the testing set for each of the proposed methods. To generate these images, we simply threshold on the output of the segmentation map $s(p)$, and then obtain the count on each connected

component by integrating the corresponding region over the density map $\lambda(p)$. Finally, we add the learned measure of annotator uncertainty $u(p)$ as a bound for the estimated count. Qualitatively, both methods obtain similar results regardless of their training differences. Further examples are available at [13].

Effect of the number of annotators. Finally, we examine how the performance of the proposed counting method is affected by the number of annotators in the training images. In the previous experiments we used all images that had at least five annotators, with an average of 8.8. Instead, we now perform the training with the same set of images but limiting the number of annotators to different thresholds; the testing set is kept the same as before. The experiment was done on the variant of our method that uses the site depth information (*with seg. and depth* in Table 1), and taking three random subsets of the annotators for each image. The results using the MCE-max metric were 7.12±0.20, 6.37±0.25 and 6.14±0.29 when limiting the number of annotators to 1, 3 and 5 respectively. We recall that the MCE-max was 5.74 when using all the annotators available. This experiment confirms an expected progressive improvement in the counting accuracy as the number of annotators per image increases.

6 Discussion

We have presented an approach that is designed to address a very challenging counting task on a new dataset with noisy annotations done by citizen scientists. We augment and interleave density estimation with foreground-background segmentation and explicit local uncertainty estimation. All three processes are embedded into a single deep architecture and the three tasks are solved by joint training. As a result, the counting problem (density estimation) benefits from the robustness that the segmentation task has towards noisy annotation. Curiously, we show that the spread between the annotators can in some circumstances help image analysis by providing a hint about the local object scale.

While we achieve a good counting accuracy in our experiments, many challenges remain to be solved. In particular, better models are required for uncertainty estimation and for crowdsourced dot-annotations. The current somewhat unsatisfactory method (using MAX and VAR as targets in training) could be replaced with a quantitative model of the uncertainty, e.g. using Generalized Extreme Value distributions to model the crowdsourced dot-annotations, with their consistent under-counting. Alternatively, dot-annotations could be modelled more formally as a spatial point processes with a rate function $\lambda(p)$. In addition, a basic model of crowdsourced dot-annotations is required in order to better disentangle errors related to the estimation model, from those errors arising from the noisy annotations.

Acknowledgements. We thank Dr. Tom Hart and the Zooniverse team for their leading role in the penguin watch project. Financial support was provided by the RCUK Centre for Doctoral Training in Healthcare Innovation (EP/G036861/1) and the EPSRC Programme Grant Seebibyte EP/M013774/1.

References

[1] Arteta, C., Lempitsky, V., Noble, J.A., Zisserman, A.: Interactive object counting. In: ECCV (2014)

[2] Chan, A.B., Liang, Z.S.J., Vasconcelos, N.: Privacy preserving crowd monitoring: Counting people without people models or tracking. In: CVPR (2008)

[3] Chan, A.B., Vasconcelos, N.: Bayesian poisson regression for crowd counting. In: CVPR (2009)

[4] Fiaschi, L., Nair, R., Köethe, U., Hamprecht, F.: Learning to count with regression forest and structured labels. In: ICPR (2012)

[5] Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. arXiv preprint arXiv:1506.02142 (2015)

[6] Idrees, H., Soomro, K., Shah, M.: Detecting humans in dense crowds using locally-consistent scale prior and global occlusion reasoning. Pattern Analysis and Machine Intelligence, IEEE Transactions on (2015)

[7] Kong, D., Gray, D., Tao, H.: A viewpoint invariant approach for crowd counting. In: ICPR (2006)

[8] Lempitsky, V., Zisserman, A.: Learning to count objects in images. In: NIPS (2010)

[9] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015)

[10] Ma, F., Li, Y., Li, Q., Qiu, M., Gao, J., Zhi, S., Su, L., Zhao, B., Ji, H., Han, J.: Faitcrowd: Fine grained truth discovery for crowdsourced data aggregation. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM (2015)

[11] Ma, Z., Yu, L., Chan, A.B.: Small instance detection by integer programming on object density maps. In: CVPR (2015)

[12] Ouyang, R.W., Kaplan, L.M., Toniolo, A., Srivastava, M., Norman, T.: Parallel and streaming truth discovery in large-scale quantitative crowdsourcing. IEEE Transactions on Parallel and Distributed Systems (2016)

[13] Penguin research webpage, www.robots.ox.ac.uk/∼vgg/research/penguins

[14] Raykar, V.C., Yu, S., Zhao, L.H., Valadez, G.H., Florin, C., Bogoni, L., Moy, L.: Learning from crowds. The Journal of Machine Learning Research (2010)

[15] Russakovsky, O., Bearman, A.L., Ferrari, V., Li, F.F.: What's the point: Semantic segmentation with point supervision. arXiv preprint arXiv:1506.02106 (2015)

[16] Shao, J., Kang, K., Loy, C.C., Wang, X.: Deeply learned attributes for crowded scene understanding. In: CVPR (2015)

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

[18] Straehle, C., Koethe, U., Hamprecht, F.A.: Weakly supervised learning of image partitioning using decision trees with structured split criteria. In: ICCV (2013)

[19] Van Horn, G., Branson, S., Farrell, R., Haber, S., Barry, J., Ipeirotis, P., Perona, P., Belongie, S.: Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In: CVPR (2015)

[20] Wang, T., Han, B., Collomosse, J.: Touchcut: Fast image and video segmentation using single-touch interaction. Computer Vision and Image Understanding (2014)

[21] Welinder, P., Branson, S., Perona, P., Belongie, S.J.: The multidimensional wisdom of crowds. In: NIPS (2010)

[22] Whitehill, J., Wu, T.f., Bergsma, J., Movellan, J.R., Ruvolo, P.L.: Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In: NIPS (2009)
[23] Wolley, C., Quafafou, M.: Learning from multiple naive annotators. In: Advanced Data Mining and Applications. Springer (2012)
[24] Xie, W., Noble, J.A., Zisserman, A.: Microscopy cell counting with fully convolutional regression networks. In: MICCAI 1st Workshop on Deep Learning in Medical Image Analysis (2015)
[25] 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 (2016)
[26] Zhang, C., Li, H., Wang, X., Yang, X.: Cross-scene crowd counting via deep convolutional neural networks. In: CVPR (2015)
[27] Zooniverse: penguinwatch.org