Counting animals in aerial images with a crowd counting model

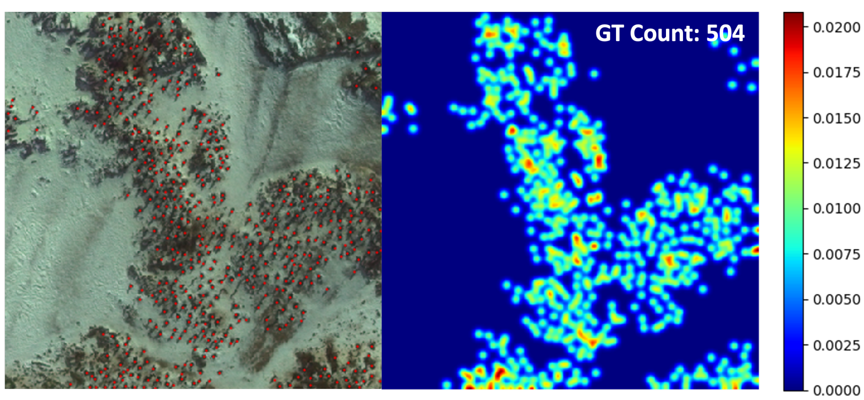
Yifei Qian[1], Grant Humphries[2], Philip Trathan[3], Andrew Lowther[4], and Carl Donovan[1]

[1]University of St Andrews
[2]Hidef Aerial Surveying Ltd
[3]British Antarctic Survey
[4]Norwegian Polar Institute

September 15, 2022

Abstract

1. Animal abundance estimation is increasingly based on drone or aerial survey photography. Manual post-processing has been used extensively, however volumes of such data are increasing, necessitating some level of automation, either for complete counting, or as a labour-saving tool. Any automated processing can be challenging when using the tools on species that nest in close formation such as Pygoscelid penguins. 2. We present here an adaptation of state-of-the-art crowd-counting methodologies for counting of penguins from aerial photography. 3. The crowd-counting model performed significantly better in terms of model performance and computational efficiency than standard Faster RCNN deep-learning approaches and gave an error rate of only 0.8 percent. 4. Crowd-counting techniques as demonstrated here have the ability to vastly improve our ability to count animals in tight aggregations, which will demonstrably improve monitoring efforts from aerial imagery.



Counting animals in aerial images with a crowd counting model

Yifei Qian[1], Grant R.W. Humphries[2], Philip N. Trathan[3, 4], Andrew Lowther[5], and Carl R.Donovan[1]

[1]School of Mathematics and Statistics, University of St Andrews, St Andrews, Fife, KY169AJ, United Kingdom

[2]HiDef Aerial Surveying Ltd, The Observatory, Dobies Business Park, Lillyhall, Cumbria CA14 4HX, United Kingdom

[3]British Antarctic Survey, High Cross, Madingley Road, Cambridge, CB3 0ET, United Kingdom

[4]Ocean and Earth Science, National Oceanography Centre Southampton, University of Southampton, University Road, Southampton, SO17 1BJ, United Kingdom

[5]Norwegian Polar Institute, Framsenteret, Postboks 6606, Stakkevollan 9296, Tromsø

* *Correspondence to* yq1@st-andrews.ac.uk

Abstract

1. Animal abundance estimation is increasingly based on drone or aerial survey photography. Manual post-processing has been used extensively, however volumes of such data are increasing, necessitating some level of automation, either for complete counting, or as a labour-saving tool. Any automated processing can be challenging when using the tools on species that nest in close formation such as Pygoscelid penguins.

2. We present here an adaptation of state-of-the-art crowd-counting methodologies for counting of penguins from aerial photography.

3. The crowd-counting model performed significantly better in terms of model performance and computational efficiency than standard Faster RCNN deep-learning approaches and gave an error rate of only 0.8 percent.

4. Crowd-counting techniques as demonstrated here have the ability to vastly improve our ability to count animals in tight aggregations, which will demonstrably improve monitoring efforts from aerial imagery.

Keywords: crowd-counting, machine-learning, image-processing, abundance estimation

1 Introduction

Aerial imagery has become the principal surveying method for many animal populations (Butler and Muller-Schwarze, 1977; Fraser et al., 1999; Philip N. Trathan, 2004; P. N. Trathan, Ratcliffe, and Masden, 2012). While manned aircraft have been been used for several decades, the increased use of Unmanned Aerial Vehicles (UAVs) is further accelerating this form of data in ecology, due to favourable efficiency, cost, and accuracy compared to 'traditional' methods (J. Hodgson et al., 2016). UAVs have already been applied to monitor a wide variety of animals such as green sea turtles (Dunstan et al., 2020), birds (Lee, Park, and Hyun, 2019) and elephants (Vermeulen et al., 2013). While this is a very efficient way to collect large amounts of data, it also creates a large post-processing burden that is frequently borne by humans - typically consisting of laborious manual scanning of photos or videos to locate, identify, and count individual animals (Torney, Dobson, et al., 2016). Volume aside, this can be a challenging task due variously to small object sizes, complex backgrounds, and varying illuminations (see Figure 1).

To alleviate these problems, there has been extensive work to integrate computer-based image processing to assist in, or fully automate, abundance estimation. J. Hodgson et al., 2016 and J. C. Hodgson et al., 2018 offer recent examples of computer-assisted animal counting, where a combination of Fourier analysis and support vector machines are used to exclude background pixels, making the subsequent manual counting of animals easier. For fully automated estimation of animal numbers, Convolutional Neural Networks (CNNs) are commonly adopted, which are a type of deep-learning neural network with components particularly directed towards images. Their use in image-processing has been transformative, with robustness proved in classification, detection and segmentation (Simonyan and Zisserman, 2015).

Automated counting of animals within images usually involves the location, and subsequent classification, of objects within a frame. In terms of CNNs, this gives rise to two broad ap-

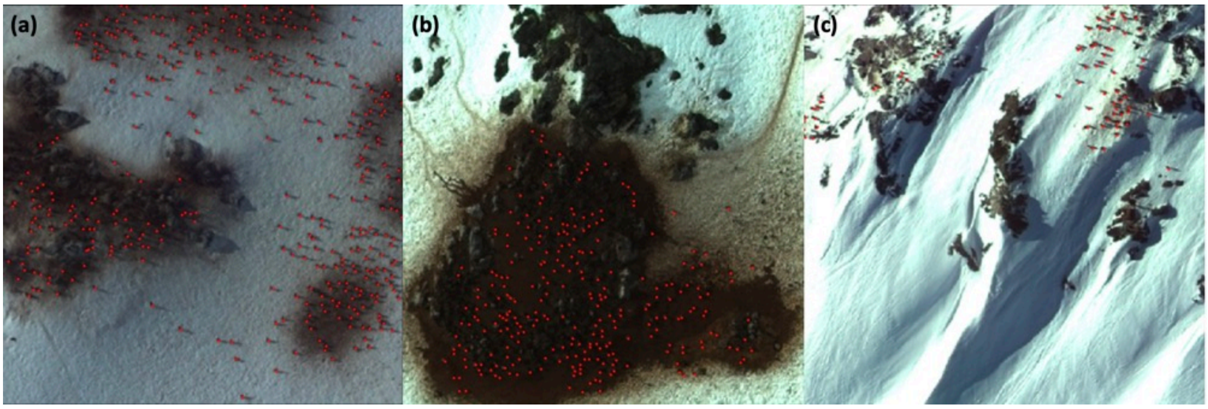


Figure 1: Selected data samples with (a) small object size, (b) complex background, and (c) varying illuminations are shown. The study object - penguins are marked with red dots.

proaches: one- and two-stage algorithms. Two-stage algorithms first propose bounding boxes for locations where objects are likely to exist, and then do the classification, where Region-based Convolutional Neural Network (RCNNs, Girshick et al., 2014) and Faster-RCNN (Ren et al., 2016) are representative examples. One-stage methods such as You Only Look Once (YOLO) (Redmon et al., 2016) and Single Shot multibox Detector (SSD) (Liu et al., 2016) process these two tasks simultaneously. In general, one-stage methods have the advantage of computing speed while two-stage methods have better accuracy.

Both methods have been previously adopted for ecological studies. Torney, Lloyd-Jones, et al. (2019) built a YOLO v3 model to detect wildebeest in aerial images, which displayed accuracy similar to manual processing whilst being quick to compute. Kellenberger, Volpi, and Tuia (2017) used a Faster-RCNN model to detect different animals in UAV images surveyed in Kuzikus Wildlife Reserve park. S.-J. Hong et al. (2019) compared the performance of different deep learning-based detection methods on a UAV aerial image dataset of wild birds and showed the potential of these techniques in monitoring wild animals. Although these detection methods work well in many cases there are constraints on object size, where objects smaller than 40 pixels will show degraded performance (S.-J. Hong et al., 2019).

Recently, Hoekendijk et al. (2021) proposed a deep CNN model to regress the count objects of interest in the image. Their model is composed of a ResNet (He et al., 2016) and two fully-connected layers. Although showing good performance, their model has a size limit on the input images, which means for a large image, it has to be cropped into the required size of patches before passing into the model. This may result in issues like replicated counts across the boundary of these image patches. Also, their results show the model only performs well up to a certain count level - when the count is out of this scope, the model exhibits poor performance.

Here we adopt a fundamentally different method for counting animals, where the detection of individual animals are avoided, with focus being the estimation of a density map - a concept initially introduced by Lempitsky and Zisserman (2010). Estimated counts are instead obtained by the subsequent integration of this density map, rather than explicit counting of objects. The density map approach has been further integrated into the deep learning framework and widely applied in crowd counting (Ma, Wei, X. Hong, and Gong, 2019; Ma, Wei, X. Hong, Lin, et al., 2021; Lin et al., 2021), where crowds are usually humans.

In this work, we create a density map estimation model based on CNNs for counting objects in aerial images. Whatever the size of objects, our model shows high accuracy in counting compared with the Faster R-CNN method which would typically be used. This is particularly relevant for our exemplar penguin data, where the objects of interest are small in terms of pixels, and detection methods are expected to show degraded performance. Our model also shows robustness when handling images of different density levels.

2 Materials and Methods

2.1 Data

2.1.1 Data collection

British Antarctic Survey currently holds an archive of colour digital aerial photography from the Antarctic Peninsula and South Shetland Islands acquired between November and December 2013, and partially re-flown in November 2015. The archive contains images from approximately

140 Pygoscelis penguin colonies selected for a range of species, population sizes and topographic settings. The images were acquired using a large-format Intergraph DMC mapping camera, with a resolution of about 12 cm or better. The images each have a footprint of about 1600 m × 1000 m and were flown with 60% overlap to allow stereo-cover. For the images to be useful as part of an automated penguin counting process they needed at least significant pre-processing to geolocate them and remove terrain distortions inherent to the perspective view of a camera image. This processing comprised: 1) the stereo-images were used to extract a Digital Elevation Model (DEM); 2) the images were ortho-rectified to the DEM to remove terrain effects; 3) the processed images were mosaicked; and then, 4) cut into standard-sized (448 × 448 pixels) tiles for counting. This process ensures that the images are accurately located and scaled to enable accurate ground area measurements and hence penguin density estimates. Without the DEM and orth-rectification pre-processing, the counts would not have a reliable ground area estimate. Stages 3) and 4) also ensure that each penguin only appears once in the dataset. The process to create the DEM is relatively complex, and utilized BAE Systems Socet GXP photogrammetry software to generate DEMs, ortho-rectify the images and prepare geo-referenced mosaics for each colony.

2.1.2 Density map generation

Our objective was to estimate the number of penguins in an image, here approached by density map estimation. The density maps are an intermediate representation generated from point annotations, with the integration of any region on these maps providing the count of target objects. The generation process is detailed here.

Given an image $\mathbf{I}$ with $\mathbf{M}$ pixels and a set of 2D annotated points $\mathbf{P} = \{p_1, p_2, ..., p_n\}$, its ground-truth density map D_{gt} can be obtained by

$$D_{gt}(I_m) = \sum_{n=1}^{N} \mathcal{N}(I_m; p_n, \sigma_n^2) \tag{1}$$

where I_m denotes a two-dimensional pixel location, $m = 1, 2, ...M$ and $\mathcal{N}(I_m; p_n, \sigma_n^2)$ represents the n^{th} annotated two-dimensional Gaussian distribution, p_n is the coordinate of n^{th} point annotation and σ_n^2 indicates the isotropic variance. The setting of σ_n^2 is flexible and often dataset dependent. It can be either fixed (Lempitsky and Zisserman, 2010) or adaptive (distance to nearest neighbours) (Zhang et al., 2016). When using the kernel with fixed bandwith, we are assuming objects are independently distributed on the image plane, while the adaptive bandwith is normally used to characterize the geometry distortion led by the perspective effect.

The choice of σ_n^2 is crucial for generating density maps, and using an improperly generated density map as a learning target may compromise the model's counting performance (Wan and Chan, 2019). Ideally, the pixels with density values should reflect consistent features, which in our case means only pixels belonging to a penguin will have density values. However, this is hard to achieve, given the typical size of a penguin is only about 4×4 pixels, while using a very small Gaussian kernel will lead to a very unbalanced sparse matrix with most values of 0, and will make the network hard to train (B. Wang et al., 2020). To achieve the balance, our generation method is given as follows: given the penguins are almost identical in size and shape in aerial images, the Gaussian kernel with fixed bandwidth is applied to the centre point of each penguin and the value of σ is set as 4. An example of these density maps is given in Figure 2. Although we don't give the location of each penguin, these density maps still retain some location information, which can indicate the region where the penguin may exist.

2.2 Specification of the density map estimation model

2.2.1 Model structure

The overall model structure is shown in Figure 3. It is a simple structure with only a backbone network and two branches. Since VGG-19 (Simonyan and Zisserman, 2015) has good performance in most computer vision tasks, such as detection and classification, and consumes relatively few computing resources, we adopt it as the backbone. However, VGG-19 learns

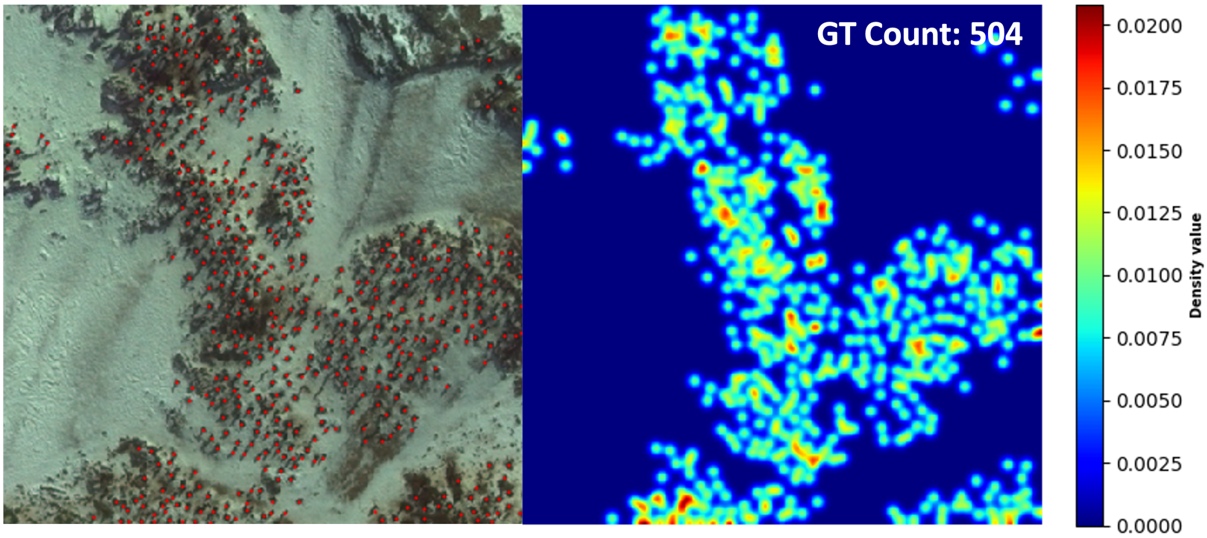


Figure 2: The left is a random image (penguins are labelled with red dots) picked from the dataset and its corresponding density map is on the right.

salient features by gradually downsampling the feature maps. To maintain high resolution of the output density map, we remove its last max pooling layer and all subsequent layers. Additionally, an up-sampling layer is added to keep the final size of the output at 1/8 of the original input. Here, bi-linear interpolation is used as the up-sampling method.

The model are designed to process two tasks: density map estimation and segmentation. Density map estimation can be seen as a two-step problem by nature, firstly is to locate regions that contains objects of interest and then regress the density values. While segmentation is to classify if a pixel belongs to the object of interest. These two tasks are interrelated and can assist the backbone to learn robust intermediate features for each other. Further, the segmentation result is used to further guide the density regression. Specifically, to prevent background features from misleading the regressor, the weights of these features are reduced before being fed into the regressor. To achieve this, we generate a mask M_d based on the predicted segmentation map S_{pred}:

$$M_d = \mathbb{1}(S_{pred} \geq 0.5) + \alpha \mathbb{1}(S_{pred} < 0.5), \tag{2}$$

where α is the dampening factor and $\mathbb{1}$ is the indicator function. We set α as 0.1. The generated

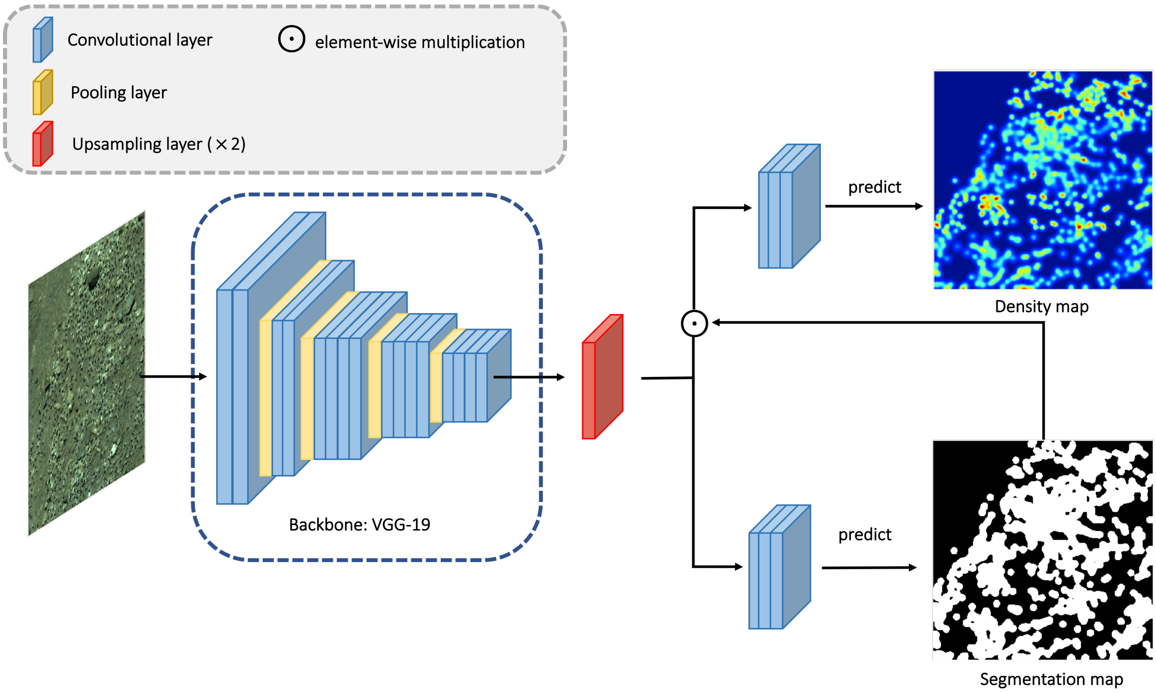


Figure 3: This figure shows the overall structure of our density map estimation model. The backbone extracts features from the input image and these intermediate features are further fed to two branches to predict density map and segmentation map.

mask M_d is then applied on the intermediate features by point-wise multiplication.

We downsample the D_{gt} by aggregating the density values to match the output size. The resulted learning target D_{target} is further used in the generation of the ground-truth segmentation map (S_{gt}):

$$S_{gt} = \mathbb{1}(D_{target} > \epsilon), \tag{3}$$

where ϵ is a density threshold and is set as 1×10^{-3} here.

Density branch & Segmentation branch

The two branches in the model share a similar structure. They both consist of three convolutional layers, the first two have a kernel size of 3 while the last one has a kernel size of 1. These layers gradually reduce the number of channels of the extracted features from 512 to 1. The Rectified Linear Unit (ReLU) (Zeiler et al., 2013) is used as the activation function for the first two layers, with the activation function for the last layer of the two branches being different. The density branch is activated with the ReLU function to make sure every point on the output is non-negative, whereas for the segmentation branch, the sigmoid (Han and Moraga, 1995) function is used to limit the range to between 0 and 1.

2.2.2 Loss function

Our overall loss function consists of two parts. Firstly, we adopt the structural loss (SL) proposed by Rong and Li (2021) to supervise the density branch, defined as:

$$SL = \frac{1}{N}\sum_{i=1}^{N}(1 - SSIM(Pool_i(D_{pred}), Pool_i(D_{target}))), \tag{4}$$

where D_{pred} represents the predicted density map, and $Pool$ stands for average pooling which down-samples the map by a factor of $\frac{1}{2^{i-1}}$. $SSIM$ is short for the Structural Similarity Index Measures (Z. Wang et al., 2004) that can describe the similarity of two images, expressed as:

$$SSIM(X, Y) = 1 - \frac{(2\mu_X\mu_Y + C_1)(2\sigma_{XY} + C_2)}{(\mu_X^2 + \mu_Y^2 + C_1)(\sigma_X^2 + \sigma_Y^2 + C_2)}, \tag{5}$$

where μ and σ denote mean and variance while σ_{XY} represents the covariance of X and Y. C_1 and C_2 are constants, set to 0.01 and 0.03 by default. The higher the SSIM index, the more similar the two images are. N is set as 3 following Z. Wang et al.' s work.

The SL function improves the structural similarity between the prediction and the target by SSIM of high-resolution maps, and the count accuracy is ensured by SSIM of the pooled density maps. Further, we make a minor change on the original loss function to improve counting accuracy, expressed as:

$$SL^* = \frac{1}{N}\sum_{i=1}^{N}(1 - SSIM(Pool_i(D_{pred} \odot S_{gt}), Pool_i(D_{target} \odot S_{gt}))), \tag{6}$$

where $\odot$ denotes point-wise multiplication. This change eliminates the contribution to the loss value from points which have negligible values on the density maps. The original SL

function pushes the value of each pixel on the predicted density map as close to the corresponding value on the target map as possible. However, in aerial images, if points are classfied into two categories based on whether they have non-zero density values, the two classes are imbalanced. Most of the points are with small values or even zero and since they are in large quantities, the regressor will compromise and tend to estimate them correctly, meanwhile underestimate points with large density values. But in fact, large density values contribute greatest to the count, so the counting accuracy will be harmed in unduly accommodating low density regions. By masking points with small values, the regressor focus is on large density values and reduces their influence. During the inference stage, integrated with the result of segmentation, we can safely discard the regressor's predictions on these points with small values and set them to 0.

The segmentation branch is supervised by the cross-entropy (CE) loss function. We adjust it to minimize the impact of the imbalance in the number of positive and negative samples in the dataset:

$$CE = \frac{1}{M} \sum_{m=1}^{M} -(y_m \log(p_m) + h * (1 - y_m) \log(1 - p_m)) \tag{7}$$

where y_m and p_m is the corresponding value of the location m in the image on the ground-truth segmentation map and the predicted probability map. h is a constant, used for balancing the contribution of positive and negative samples to the loss value and is set as 0.5 in our experiments.

The final loss function is a weighted sum of the above two loss functions:

$$Loss = SL^* + \lambda CE \tag{8}$$

with λ set to 0.1 since the density estimation is the main task of the model.

2.2.3 Model inference

Our model adopts a fully convolutional design, which means it has no strict size constraints on the input image. However, there are four max-pooling layers with kernel size of 2 in the backbone structure, which may result in pixel dropout. To prevent this, the input image has to be enlarged to the smallest size divisible by 16. The output density map D_{out} integrates the predictions from both branches and can be obtained by:

$$D_{out} = D_{pred} \odot \mathbb{1}(S_{pred} \geq 0.5) \tag{9}$$

2.2.4 Experiments

We randomly split our dataset into three parts in a ratio of 3:1:1. The largest part serves as the training set and the remaining parts are used for the purpose of validation and test, respectively. The detailed statistics of these three datasets are shown in the Table 1. Notably, these datasets show drastic change in density distribution and all contain a few samples that are only backgrounds.

Table 1: The statistics of the training, validation and test set. L0, L1, L2, L3 and L4 represent the number of images containing 0, 1-100, 101-500, 501-1000 and 1000+ penguins. Total gives the total number of penguins in the dataset, while Max and Average show the maximum and average number of penguins in one image in the dataset, respectively.

Dataset	Number of images	L0	L1	L2	L3	L4	Total	Max	Average
Training set	446	118	140	137	34	17	87654	2682	196
Validation set	146	35	61	34	14	2	23918	1361	164
Test set	146	39	59	36	6	6	23707	1580	162

In our experiments, we adopt random cropping and random horizontal flipping as data augmentation strategies for training the model. The cropping size is set as 256 $\times$ 256. The parameters of the backbone are initialized with the VGG-19 pre-trained on ImageNet (Deng et al., 2009) and others are randomly initialised from a Gaussian distribution with a standard deviation of 0.01. We train the network for 600 epochs with a batch size of 16 using the Adam optimizer (Kingma and Ba, 2015). We fix the learning rate as 1e-5 and the weight decay as

1e-4. The validation starts after the 100th epoch. The model with the best performance on the validation set is used to report the final result on the test set.

For comparison, we also implement a Faster-RCNN model, the detailed training process is provided in the supplementary materials.

All experiments are carried on a single 16 GB Tesla P100 GPU, with methods implemented with Pytorch. The whole training process takes approximately 3 hours.

3 Results

To evaluate our method, we use the mean absolute error (MAE) and root mean squared error (RMSE) metrics, defined as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|C_i^{pred} - C_i^{gt}| \tag{10}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(C_i^{pred} - C_i^{gt})^2} \tag{11}$$

where N is the total number of the images, C_i^{pred} and C_i^{gt} is the predicted count and the ground-truth count of i-th image, respectively.

We define the model which has the lowest sum of MAE and MSE on the validation data as the best model. This model's performance on the test set is shown in Table 2. To better illustrate our model's performance we provide the results from a Faster-RCNN model for comparison. In addition, separate average performance on images with different count levels, L0 (0), L1 (1-100), L2 (101-500), L3 (501-1000), L4 (1000+), are also calculated. Overall, our model has an outstanding performance on this task, and outperforms the Faster-RCNN model in all aspects. It is also worth mentioning that the count error at the dataset level for our model is +186.6 (+0.8%) while for Faster-RCNN is 4741 (+20.2%).

Some of the estimated density maps are presented in Fig 4. Although the prediction's

Table 2: The evaluation result of our model and the Faster-RCNN on the test set.

Models	Overall		MAE					RMSE				
	MAE	RMSE	L0	L1	L2	L3	L4	L0	L1	L2	L3	L4
Our model	19.9	39.4	7.2	10.8	31.1	65.6	78.8	32.4	16.2	43.0	70.4	111.3
Faster RCNN	54.8	78.9	20.0	51.7	74.4	89.7	158.2	44.0	64.4	95.2	110.6	177.9

resolution is only one-eighth the resolution of the generated ground-true density map, it exhibits similar characteristics at the image level.

4 Discussion

The algorithmic counting of objects in aerial images in ecological studies was previously dominated by detection algorithms. In this section, we will discuss our model's advantages over these traditional detection methods.

Overall, our model has four main advantages over detection methods. First and foremost, our method is able to count extremely small objects. In the case of aerial images, the object of interest in an image is likely to be very small, especially for ecological surveys - in our case, only about 5 $\times$ 5 pixels. Our experiments show even the two-staged detection algorithm Faster-RCNN fails to detect most of the penguins. The reason is as follows: no matter what detection methods, a backbone structure is essential for extracting features. However, the current mainstream deep network structure, often used as the backbone, will downsample the image to a certain extent, for example, the downsampling ratio of VGG series is 16, while 32 for ResNet series (He et al., 2016). With a high downsampling ratio, the representation of a small object on the final feature maps may not be abundant enough for subsequent neural networks to predict the location and classification simultaneously. In contrast, our density estimation model only focuses on the counting of locations on the feature map instead of individuals, which provides better count accuracy.

Secondly, our model only requires point annotation, which means annotators need only to mark the same part of each object with a dot, quite similar to the way human counts. This

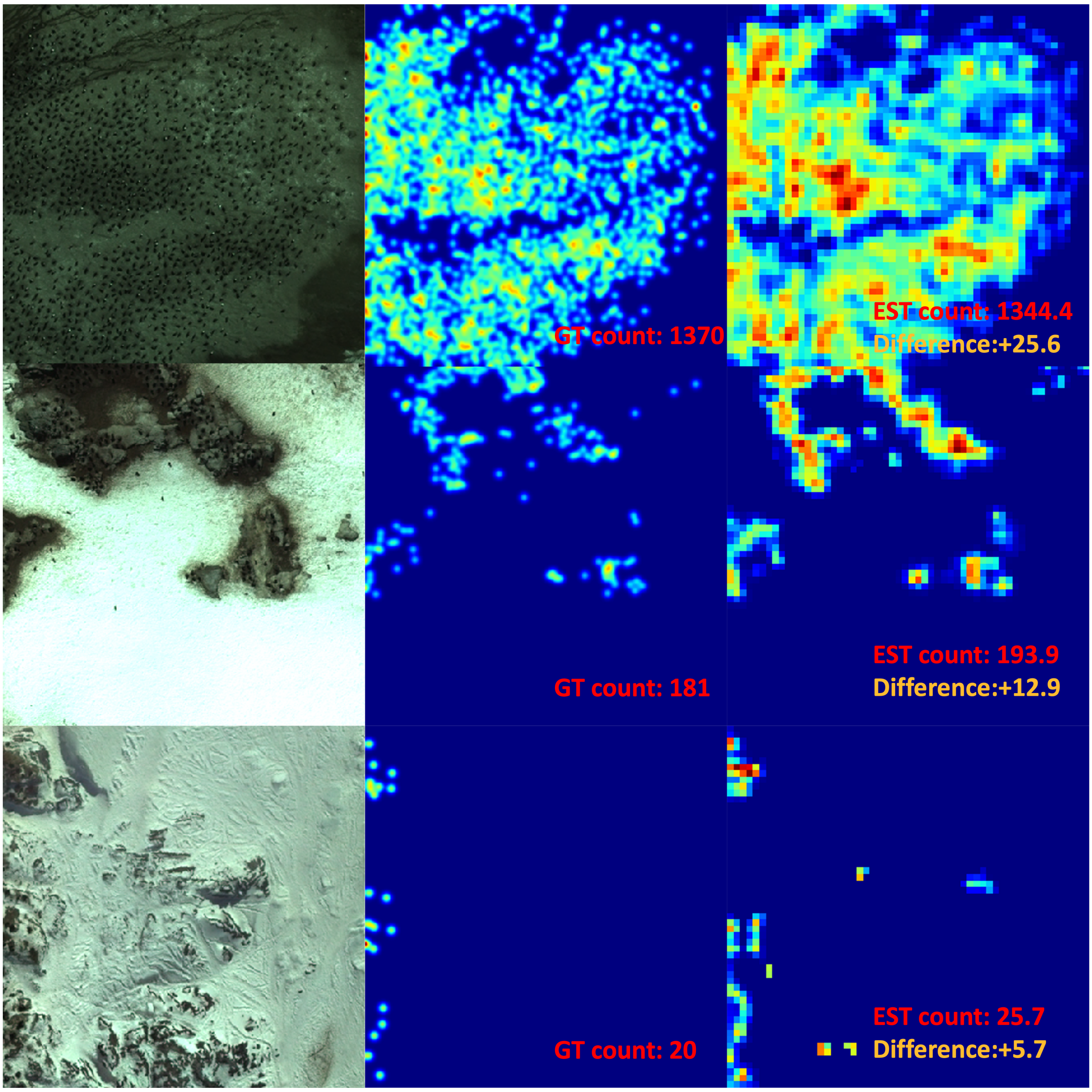


Figure 4: Some visualization results of the estimated density maps. The three images in each row, from left to right is the input, the Gaussian-smoothed ground-truth density map and the prediction. The corresponding count is given in the lower right corner of the density map. The difference between the ground-truth and the estimated counts is highlighted.

reduces markedly the labelling effort, compared to bounding box annotations required by the detection methods, where annotation consists of drawing a rectangle around the object, closely matching the object's edges, which is laborious.

Thirdly, the density map estimation method can better handle objects located at the edge of the image. It's often the case that the images taken by UAV are of large size, and considering GPU memory constraints, researchers have to crop them into digestible pieces for the deep learning networks. It's inevitable that some objects are also split into pieces, scattering them over several image patches. Such a situation results in a complex detection result whether objects are undetected due to incomplete feature representations, or are repeatedly detected across multiple image patches. However, this will not pose a problem to the density estimation model, where the count of an object is not necessarily integer, thanks to the Gaussian smoothing. Hence, there will not produce redundant counts when summing up two non-overlapping neighbouring image patches.

Last but not least, our model can utilise negative samples (images with zero count) during training phase, which makes it more robust than the detection model when dealing with background. For drone footage, there will be many images that are completely background i.e. no objects. However, detection algorithms can not use them since they require every training sample to contain at least one object of interest. This is a fundamental short-coming of the detection algorithms. Meanwhile, our model can fully use these images to improve its ability to differentiate the foreground and background. This also explains the large difference in performance of these two models on images of count level, L0.

In this work, we propose a CNN-based density map estimation model to count extremely small penguins in aerial images, especially those acquired by UAV systems. Compared to the traditional two-staged detection method, Faster RCNN, our model shows a significant improvement in counting accuracy when faced wtih small objects. Although the precise location of each object is not given, the model still indicates areas where objects may exist. Another potential advantage of our model is through reducing the labour in image annotation. In some studies, the object counting needs to be as precise as possible, necessitating a human counter despite the labour. In this case our model can aid the process by excluding regions that do not need detailed consideration. Overall, we hope our research can help researchers who use drones in ecological surveys.

5 Data Availability

The dataset will be archived and in the UK Polar Data Centre and available to public at the time of publication. The code is available here: https://github.com/cha15yq/Counting-penguins-in-aerial-images

6 Acknowledgements

WWF (UK) supported PNT under grant GB095701, which also provided funds to develop the training datasets. The authors thank Adrian Fox and Nathan Fenney at BAS for help with aerial image acquisition and image processing.

7 contribution

All authors contributed critically to the drafts and gave final approval for publication. Yifei Qian and Carl R.Donovan conceived and designed statistical and computational methodology; Grant R.W. Humphries, Philip N. Trathan and Andrew Lowther provided data and biological expertise; Yifei Qian and Carl R.Donovan led the writing of the manuscript.

8 conflict

The authors declare no competing interests.

References

Butler, R.G. and D. Muller-Schwarze (1977). "Penguin census by aerial photographic analysis at Cape Crozier, Ross Island". In: *Antarctic Journal of the USA* 12, pp. 25–27.

Deng, Jia et al. (2009). "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

Dunstan, Andrew et al. (June 2020). "Use of unmanned aerial vehicles (UAVs) for mark-resight nesting population estimation of adult female green sea turtles at Raine Island". In: *PLOS ONE* 15, pp. 1–18. DOI: 10.1371/journal.pone.0228524.

Fraser, William R. et al. (1999). "Using Kite-Based Aerial Photography for Conducting Adélie Penguin Censuses in Antarctica". In: *Waterbirds: The International Journal of Waterbird Biology* 22.3, pp. 435–440. ISSN: 15244695, 19385390.

Girshick, Ross et al. (2014). "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Han, Jun and Claudio Moraga (1995). "The Influence of the Sigmoid Function Parameters on the Speed of Backpropagation Learning". In: *Proceedings of the International Workshop on Artificial Neural Networks: From Natural to Artificial Neural Computation*. IWANN '96. Berlin, Heidelberg: Springer-Verlag, pp. 195–201. ISBN: 3540594973.

He, Kaiming et al. (2016). "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

Hodgson, Jarrod et al. (Mar. 2016). "Precision wildlife monitoring using unmanned aerial vehicles". In: *Scientific reports* 6, p. 22574. DOI: 10.1038/srep22574.

Hodgson, Jarrod C. et al. (2018). "Drones count wildlife more accurately and precisely than humans". In: *Methods in Ecology and Evolution* 9.5, pp. 1160–1167. DOI: https://doi.org/10.1111/2041-210X.12974. eprint: https://besjournals.onlinelibrary.wiley.com/doi/pdf/10.1111/2041-210X.12974.

Hoekendijk, Jeroen et al. (Dec. 2021). "Counting using deep learning regression gives value to ecological surveys". In: *Scientific Reports* 11, p. 23209. DOI: 10.1038/s41598-021-02387-9.

Hong, Suk-Ju et al. (2019). "Application of Deep-Learning Methods to Bird Detection Using Unmanned Aerial Vehicle Imagery". In: *Sensors* 19.7. ISSN: 1424-8220.

Kellenberger, Benjamin, Michele Volpi, and Devis Tuia (2017). "Fast animal detection in UAV images using convolutional neural networks". In: *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 866–869. DOI: 10.1109/IGARSS.2017.8127090.

Kingma, Diederik P. and Jimmy Ba (2015). "Adam: A Method for Stochastic Optimization". In: *ICLR (Poster)*.

Lee, Won Young, Mijin Park, and Chang-Uk Hyun (Sept. 2019). "Detection of two Arctic birds in Greenland and an endangered bird in Korea using RGB and thermal cameras with an unmanned aerial vehicle (UAV)". In: *PLOS ONE* 14, pp. 1–16. DOI: 10.1371/journal.pone.0222088.

Lempitsky, Victor and Andrew Zisserman (2010). "Learning To Count Objects in Images". In: *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*. NIPS'10. Vancouver, British Columbia, Canada: Curran Associates Inc., pp. 1324–1332.

Lin, Hui et al. (Aug. 2021). "Direct Measure Matching for Crowd Counting". In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. Ed. by Zhi-Hua Zhou. International Joint Conferences on Artificial Intelligence Organization, pp. 837–844.

Liu, Wei et al. (2016). “SSD: Single Shot MultiBox Detector”. In: *Lecture Notes in Computer Science*, pp. 21–37. ISSN: 1611-3349. DOI: 10.1007/978-3-319-46448-0_2.

Ma, Zhiheng, Xing Wei, Xiaopeng Hong, and Yihong Gong (2019). “Bayesian Loss for Crowd Count Estimation With Point Supervision”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6141–6150. DOI: 10.1109/ICCV.2019.00624.

Ma, Zhiheng, Xing Wei, Xiaopeng Hong, Hui Lin, et al. (2021). “Learning to Count via Unbalanced Optimal Transport”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.3, pp. 2319–2327.

Redmon, Joseph et al. (2016). “You Only Look Once: Unified, Real-Time Object Detection”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788. DOI: 10.1109/CVPR.2016.91.

Ren, Shaoqing et al. (2016). *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. arXiv: 1506.01497 [cs.CV].

Rong, Liangzi and Chunping Li (Jan. 2021). “Coarse- and Fine-Grained Attention Network With Background-Aware Loss for Crowd Density Map Estimation”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3675–3684.

Ruder, Sebastian (2017). *An overview of gradient descent optimization algorithms*. arXiv: 1609.04747 [cs.LG].

Simonyan, Karen and Andrew Zisserman (2015). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. arXiv: 1409.1556 [cs.CV].

Torney, Colin J., Andrew P. Dobson, et al. (May 2016). “Assessing Rotation-Invariant Feature Classification for Automated Wildebeest Population Counts”. In: *PLOS ONE* 11.5, pp. 1–10. DOI: 10.1371/journal.pone.0156342.

Torney, Colin J., David J. Lloyd-Jones, et al. (2019). “A comparison of deep learning and citizen science techniques for counting wildlife in aerial survey images”. In: *Methods in Ecology and Evolution* 10.6, pp. 779–787. DOI: https://doi.org/10.1111/2041-210X.13165. eprint: https://besjournals.onlinelibrary.wiley.com/doi/pdf/10.1111/2041-210X.13165.

Trathan, P. N., N. Ratcliffe, and E. A. Masden (2012). “Ecological drivers of change at South Georgia: the krill surplus, or climate variability”. In: *Ecography* 35.11, pp. 983–993. DOI: https://doi.org/10.1111/j.1600-0587.2012.07330.x.

Trathan, Philip N. (2004). “Image Analysis of Color Aerial Photography to Estimate Penguin Population Size”. In: *Wildlife Society Bulletin (1973-2006)* 32.2, pp. 332–343. ISSN: 00917648, 19385463.

Vermeulen, Cédric et al. (Feb. 2013). “Unmanned Aerial Survey of Elephants”. In: *PLOS ONE* 8.2, pp. 1–7. DOI: 10.1371/journal.pone.0054700.

Wan, Jia and Antoni Chan (2019). “Adaptive Density Map Generation for Crowd Counting”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1130–1139. DOI: 10.1109/ICCV.2019.00122.

Wang, Boyu et al. (2020). “Distribution Matching for Crowd Counting”. In: *Advances in Neural Information Processing Systems*.

Wang, Zhou et al. (2004). “Image quality assessment: from error visibility to structural similarity”. In: *IEEE Transactions on Image Processing* 13.4, pp. 600–612. DOI: 10.1109/TIP.2003.819861.

Zeiler, M.D. et al. (2013). “On rectified linear units for speech processing”. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3517–3521. DOI: 10.1109/ICASSP.2013.6638312.

Zhang, Yingying et al. (2016). “Single-Image Crowd Counting via Multi-Column Convolutional Neural Network”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 589–597. DOI: 10.1109/CVPR.2016.70.