Animal Biotelemetry

METHODOLOGY

Open Access

Counting sea lions and elephants from aerial photography using deep learning with density maps

Chirag Padubidri[1,2*], Andreas Kamilaris[1,2], Savvas Karatsiolis[2] and Jacob Kamminga[1]

Abstract

Background: The ability to automatically count animals is important to design appropriate environmental policies and to monitor their populations in relation to biodiversity and maintain balance among species. Out of all living mammals on Earth, 60% are livestock, 36% humans, and only 4% are animals that live in the wild. In a relatively short period, development of human civilization caused a loss of 83% of wildlife and 50% of plants. The rate of species extinction is accelerating. Traditional wildlife surveys provide rough population estimates. However, emerging technologies, such as aerial photography, allow to perform large-scale surveys in a short period of time with high accuracy. In this paper, we propose the use of computer vision, through deep learning (DL) architecture, together with aerial photography and density maps, to count the population of Steller sea lions and African elephants with high precision.

Results: We have trained two deep learning models, a basic UNet without any feature extractor (Model-1) and another with the *EfficientNet-B5* feature extractor (Model-2). We measured the model's prediction accuracy, using Root Mean Square Error (RMSE) for the predicted and actual animal count. The results showed an RMSE of 1.88 and 0.60 to count Steller sea lions and African elephants, respectively, regardless of complex background, different illumination conditions, heavy overlapping and occlusion of the animals.

Conclusions: Our proposed solution performed very well in the counting prediction problem, with relatively low training parameters and minimum annotation. The approach adopted, combining DL and density maps, provided better results than state-of-art deep learning models used for counting, indicating that the proposed method has the potential to be used more widely in large-scale wildlife surveying projects and initiatives.

Keywords: Animal counting, Steller sea-lions, Elephant, Deep learning, Aerial photography

Background

Of all living mammals on Earth, 60% are livestock, 36% humans, and just 4% are wild animals [1]. In a relatively short period, the development of human civilization caused a loss of 83% of wildlife and 50% of vegetation [2]. Moreover, the current rate of global decline in wildlife population is unprecedented in human history, and the rate of extinction of species is accelerating [3, 4]. Wildlife surveys provide species' population estimates and are conducted for reasons such as species management and control, ecological and biological studies, as well as long term trend monitoring and behavioral understanding. This information may be essential for the survival of species and maintaining ecological balance. For example, biologists use population trends to investigate the effect of environmental factors, such as the impact of human activity on a species' population in some area/region.

*Correspondence: c.padubidri@cyens.org.cy
[2] CYENS Center of Excellence, Nicosia, Cyprus
Full list of author information is available at the end of the article

The Steller (or northern) sea lion (SSL; Eumetopias jubatus), the largest member of the family Otariidae, experienced a widespread population decline. The causes for the decline are likely multi-factorial and include a combination of anthropogenic factors, like commercial fishing, commercial hunting, and natural factors like environmental changes, pollution, disease, and natural fluctuation [5]. Environmental change and commercial fishing are believed to be the most probable links to the decline [6]. The Alaskan sea lion population has been reduced to a small fraction of earlier levels resulting in the species being listed as threatened under the U.S. Endangered Species Act(ESA) [7] in November 1990; the western stock was changed to endangered in l997 [6]. A similar trend can be seen in other wildlife. The survey conducted in 18 African countries with historical data showed a decrease in the population of African bush elephant (Loxodonta africana) by an estimated 144,000 between 2007 and 2014, and the population was shrinking by 8% per year continent-wide, primarily due to poaching [8]. Many conservation efforts have been performed to restore the wildlife population. The National Marine Fisheries Service (NMFS) is one such organization working towards marine animal conservation in the USA. NMFS conducts annual aerial surveys to estimate the Steller sea lion population [9]. This survey information can be used to adapt and formulate local and global policies to protect and conserve wildlife.

The use of satellite imagery and aerial photography allows biologists to survey remote species across vast areas. However, manual counting methods based on human labeling are laborious, expensive, limited, logistically challenging, etc. For example, it takes a team of biologists up to 4 months to count Steller sea lions from thousands of photographs collected each year by the NMFS [10]. Automated counting methods based on computer vision techniques may accelerate analyses of the animal census process via wildlife surveys and free up critical resources, allowing organizations to focus on the actual conservation of the animals.

In this paper, the use of deep learning (DL) to automatically count the animals from aerial photography is considered in combination with the modern and promising technique of density maps [11, 12]. DL is a powerful computer vision technique demonstrating excellent performance for environmental monitoring [13–15]. DL extends machine learning (ML) by adding more "depth" (complexity) into the model, transforming the data using various functions that allow data representation in a hierarchical way, through several abstraction levels. Compared to traditional techniques, such as Support Vector Machines and Random Forests, DL has demonstrated enhanced performance in classification and counting computer vision-related problems [15]. A preliminary effort in this direction can be found in [16], where real and synthetic data were used together to count the number of houses from aerial photographs acquired by unmanned aerial vehicles (UAVs).

Related work

DL has been demonstrated as a promising technological solution to the problem of counting entities in images [15]. DL models can be grouped into two categories based on their function: discriminative and generative [17]. Discriminative models are used for predictions/classifications, whereas generative models are used for synthesis/generation of data similar to the input data set. The use of generative data to train DL models is promising, with early attempts in agriculture indicating positive outcomes [16]. Discriminative models, focused on predictions of the precise number of targets in an image, i.e., *counting*, are employed herein. Three annotation methods are explored in this work, as described below.

Counting via detection

In this method, a visual object detector is used to localize individual object instances in the image. Given the localization, counting becomes trivial. In this case, objects are annotated by a bounding box. Several methods [18–20] use detection based object counting. For instance, in [18], a sliding window detection and classification algorithm was proposed to count Steller sea lions. However, counting via detection yields poor results when there is high occlusion among objects in the image, whereas the annotation of densely crowded images is computationally expensive.

Counting via image-level regression

This method is based on image-level label regression. The images are directly fed to the Neural Network to learn a non-linear function to predict the count, making no need for additional labeling of the data. Hence, this method employs the least computationally expensive annotation technique. In [21], an accuracy of 91% is reported for a regression model used to count tomatoes. The model learns feature directly from the input image using regression and predicts the number of tomatoes in the image. However, this method cannot perform object localization, hence cannot be used in cases, where the knowledge of the spatial distribution of objects is important.

Counting via Density Maps

In this method, a point corresponding to each object's location in the image is marked and a density heat map is used for training the model. The spatial properties of the density map frame the scene information better than the previous techniques, mitigating the problem of occlusion. The learning-to-count model of [22] introduces a counting approach, which works by learning a linear mapping from local image features to object density maps. By properly training the DL model, one can estimate the object count by simply integrating over regions of the density map produced by the model. The same strategy is employed in [11, 12].

In [11], a modified Counting Convolutional Neural Network (Counting CNN) model, which is a combination of CCNN and ResNeXt models, is proposed for estimating pig density in livestock farms. The CNN model does not depend on foreground segmentation as it takes only appearance information into consideration. A similar proposal is the DisCountNet model, a two-stage network (DiscNet and CountNet) that uses theories from both detection and density-map networks [12]. Initially, the DiscNet model performs a coarse detection of the patches of the images from a larger input image containing dense objects. The CountNet model then operates on the dense regions of the sparse matrix to generate a density map, which provides fine locations and count predictions on densities of objects. This method requires prior information about the data set to appropriately set operational parameters.

Our contribution

In this work, different data sets are used to count animals using a DL approach based on density maps: a) the Steller sea lions data set [10] and b) the Aerial Elephant data set [23]. This paper is not the first targeting wild animal counting. The novelty of this work lies in the approach of counting animals via density maps, which has important differences in implementation than the use of this technique in related work [11, 12] (see "Methods" and "Proposed approach" sections). Moreover, our data sets have unique challenges due to the high overlapping and occlusion observed. For example, female sea lions frequently cluster together with their pups during their everyday life (see Fig. 2b). The same goes to female African elephants with their calves. By employing the UNet model, an encoder–decoder architecture for animal counting from aerial images [24], we used semantic segmentation for density estimation without pixel-level annotation, improving the counting and localization performance with minimum annotation, demonstrating better results than the state-of-art models and research works.

Methods

Data sets

Steller sea lion data set

The Steller sea lion data set from the Kaggle competition [10] consists of a training and test folder. The training images were manually annotated with a dot by human annotators. For the purpose of this work, we considered only the annotated training images. The training image folder had 948 aerial images comprising different categories and a number of SSLs. The SSL categories have been defined based on age and sex: (a) adult males, (b) subadult males, (c) adult females, (d) juveniles, and (e) pups. Each image was provided with two versions: the original and a dot-annotated image, with dots approximately at the center of each SSL. We assumed that the dot-annotation provided is without any error. The average image resolution was approximately 5000 × 3000 with 3 channels, each image roughly occupying 5 MB.

The Aerial Elephant Data Set

The Aerial Elephant Data Set [23] is a collection of aerial images of African bush elephants, aiming to promote research on animal detection under real-world conditions. The images were gathered using Canon 6D consumer-oriented digital single-lens reflex cameras, which were mounted in a SkyReach BushCat light sport aircraft by means of a purpose-built frame [25]. To maximize the width of the image strip underneath the plane, the frame was built to accommodate three cameras. One camera was oriented such that the lens was pointing straight down centered between the other two cameras tilted to the left and right. The data set consists of 2,074 images containing a total of 15,581 African bush elephants in their natural habitats, imaged with a consistent methodology over a range of background types, resolutions, and times-of-day. These images were acquired over the course of 8 separate campaigns in different physical environments. The images have a resolution of $\approx 5000 \times 3000$ with 3 channels. The resolution of the images varied between 2.4 and 13 cm/pixel. The data set represents both dry- and wet-season backgrounds in a variety of landscapes, captured over the full day from sunrise to sunset.

Proposed approach

A semantic segmentation end-to-end model inspired by the UNet DL model [24] was used. The model was designed to produce density maps, containing precise locations of the animals from input images. The principal disadvantage of semantic segmentation algorithms is the tedious requirement of pixel-level annotation during training. We used dot annotation, i.e., placing dots at the center of each animal, which largely reduces the annotation overhead. Significant improvements in counting and localization performance are gained with only minimum annotation required.

The input image is fed to the proposed architecture, where the *Encoder* part of the architecture generates a high dimensional feature vector1. Then, the *Decoder* semantically projects the features learned by the encoder onto the pixel space, generating a corresponding density distribution for the given image. This density map is used to get the number of objects by simply integrating the density distribution over the region (see "Counting from density map" section) (Fig. 1).

Data preparation

Both data sets were split into training and testing images with a ratio of 80 : 20, respectively. For the SSLs' data set, the first 800 images were used for training, and the remaining 148 were used as test images to evaluate the model's performance. For the elephants' data set, 1649 images were used for training, while 452 images were used for testing.

We observed that the large image resolution of the SSLs' data set gave better details about the animals than the elephants one (Fig. 2a). In both data sets, the number of animals in each image varied significantly and were grouped closely together, leaving large portions of the image with background only. To address this issue and accelerate the training process, some image pre-processing operations were performed (Fig. 2). Images were cropped to remove sections containing only background information (Fig. 2b). A sliding window-based cropping with 10% overlap was employed to produce images of size 256 × 256 (Fig. 2c). Images that did not contain any animal were manually discarded from the training set (Fig. 2c, crops 3 and 4).

The number of SSLs per image in the pre-processed images ranged from $1 - -80$ with mean $\mu = 4$ and standard deviation $\sigma = 6$, whereas the number of elephants per image varied between $1 and 12$ with mean $\mu = 0.67$ and standard deviation $\sigma = 1.12$.

Implementation

We tried to avoid some pitfalls observed in related work, where information loss occurred due to downsampling and reduction of spatial resolution in higher

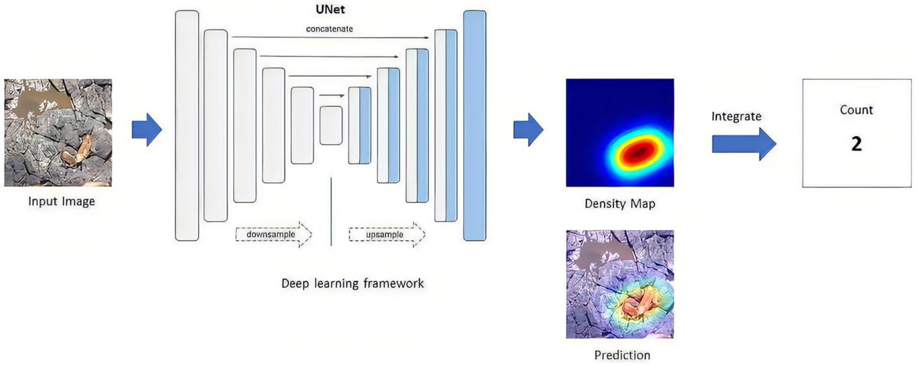

Fig. 1 Block Diagram: The proposed method for SSL counting. Given an aerial image, we feed it into our DL architecture to produce a density map, from which we are able to estimate the number of SSLs present in the image

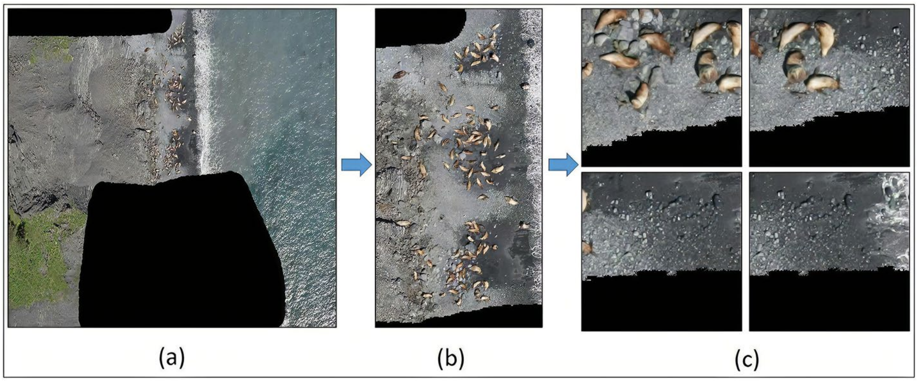

Fig. 2 Data Pre-processing Workflow **a** The original image with dimension (*3328X4992*). **b** Background cropped image. **c** Sliding window-based cropped images of size (*256X256*)

layers of the DL models employed. Specifically, the feed-forward regression networks in [11, 26] compress and encode images into smaller representation vectors, while the combination of CCNN and ResNeXt models in [11] takes an input image of size 72 × 72 and produces an output density map of size 18 × 18. To overcome this potential information loss, the *UNet* model was employed as the learning model [24]. UNet is a CNN architecture originally proposed for biomedical image segmentation, based on an encoder–decoder-type network. The name of the architecture is derived from its distinctive 'U' shape. The down-sampling/encoder block encodes the input images into feature representations at multiple level, capturing the context information from the image. The up-sampling/decoder block decodes the feature maps learned from the encoder. The symmetric up-sampling path enables precise localization of the objects in the image.

In the proposed work, the down-sampling (contracting) path repeatedly applies a block comprised of two 3 × 3 convolutions, followed by batch-normalization, a Rectified Linear Unit (ReLU) activation and a 2 × 2 max-pooling layer of stride 2. The number of feature map channels in the contracting path is doubled at each down-sampling block. Similarly, the up-sampling (expansive) path replaces the max-pooling layers with up-sampling layers that apply nearest-neighbor interpolation. Analogous to the contracting path, the number of feature map channels is halved at each up-sampling block. The feature maps of the up-sampling path are concatenated with the feature maps of the contracting path. Finally, the output layer results by applying a 1 × 1 convolution.

Loss function

In accordance with the performance metrics used in similar previous studies, the Root Mean Square Error (RMSE) loss between the predicted density map ($\hat{D}$) and the true density map (D) is employed in this paper. RMSE is the square root of the average of squared differences between predicted and actual count, defined as

$$L = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(\hat{D} - D)^2} \tag{1}$$

Counting from density map

For training, a set of annotated images was used, where all animals had been marked with dots placed approximately at their center of mass. The ground truth density map D_x, for an image x, is defined as a sum of Gaussian functions centered at the 2D coordinates p of each dot:

$$D_x = \sum_{p \in A_x} \mathcal{N}(p, \textstyle\sum) \tag{2}$$

where A_x is the set of 2D-point animal annotations for the image x and $\mathcal{N}(p, \sum)$ represents an isotropic 2D Gaussian function with a mean p and a covariance matrix $\sum$. Covariance is modeled as $\sum = \sigma^2 I$, with I being the identity matrix and σ is a parameter linked to the spread of the distribution. For our application, the sigma value was selected based on the average pixel width at the center of the animal as derived from the input data sets (see "Datasets" section). A $\sigma = 25$ was selected for SSL and $\sigma = 10$ for elephant.

The density map represents the distribution. Given the density map D_x, the total count N_x can be obtained by summing up the pixel values of D_x as shown below:

$$N_x = \sum_{i,j} D_{x_{i,j}} \tag{3}$$

In summary, the model predicts the distribution density map, while the total number of objects of interest (i.e., animals' counting) is obtained via the integration of the density map over the image space.

Performance metrics

Mean Absolute Error (MAE) and RMSE were the metrics employed for evaluating the model's performance, in respect to the accuracy of the model's animal counting. The MAE is the average of the absolute differences between predicted and actual count, i.e., it measures the average magnitude of the errors in a set of predictions. The MAE characterizes the accuracy of the algorithm, while the RMSE represents the degree of dispersion of the error. The mathematical representations of MAE and RMSE are provided below:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|(y_i - \hat{y}_i)| \tag{4}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}. \tag{5}$$

where y_i is the actual animal count in the ith image, $\hat{y}_i$ is the predicted animal count in the ith image and N is the total number of test images.

Results

To assess the performance of the proposed network architecture, we employed two different models and trained: ***Model-1***, i.e., a basic UNet without any feature extractor and ***Model-2***, i.e., a UNet with the

EfficientNet-B5 feature extractor [27]. EfficientNet is a CNN developed by Google, characterized by high accuracy and computational efficiency. Model-2 was initialized by pre-trained weights based on the Imagenet data set [28]. All the parameters were optimized using the Adam optimizer with a learning rate of *0.001*.

Training

An Nvidia GeForce RTX 2060 GPU was used for training, with a batch size of 8. Model-1 (without feature extractor) was trained for 7 h and Model-2 (with a feature extractor) for 17 h. Based on the model's performance on the validation set, the early stopping technique was applied to avoid over-fitting. Model-2, which used pre-trained weights and thus some prior relevant information, converged faster than Model-1 (Fig. 3).

Model evaluation: testing

Table 1 shows the model's performance on the testing images (see "Data preparation" section). Model-2, with 37M parameters, outperformed Model-1 with 15M parameters in terms of the counting prediction, for both the SSL and elephants' data sets. Figure 4 shows the actual vs. predicted number of animals. The diagonal red line represents the case of no errors (i.e., perfect prediction). The closer the points are to the line, the better the prediction.

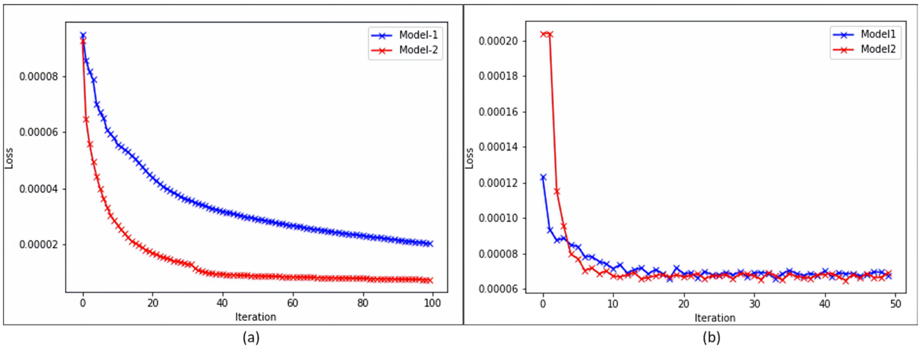

Fig. 3 Training loss curve: training loss function gradient vs. iteration curve for Basic UNet (Model-1) and UNet with EfficientNet-B5 feature extractor architecture (Model-2)

Table 1 Performance comparison of Model-1, Model-2 on test data set

Model	Feature extractor	SSL		Elephant		Parameters
		RMSE	MAE	RMSE	MAE	
Model-2	Eff.Net-B5	1.88	1.09	0.60	0.34	≈37M
Model-1	*No*	5.57	3.54	1.01	0.53	≈14M

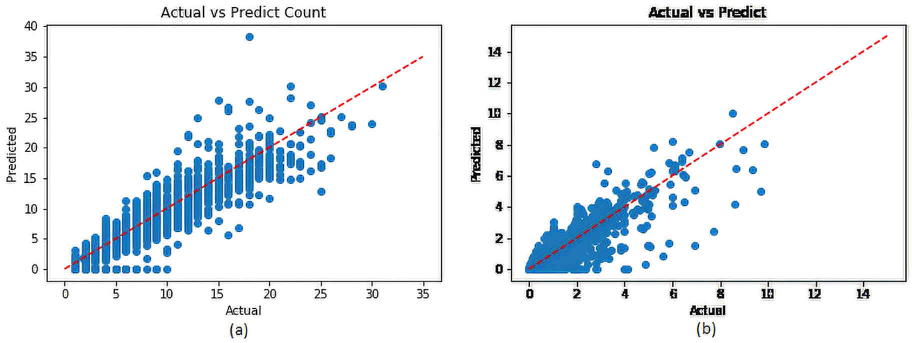

Fig. 4 Visualizing actual vs. predicted count: scatter plot representing ground-truth count and predicted count. The diagonal red line indicate the zero error predictions. **a** SSL predictions, **b** African Elephant predictions

Discussion

We present a method for counting animals in images by calculating the probability of each pixel showing a part of an animal and then integrating over these probabilities to predict the number of animals. This approach is fundamentally different from the prediction of animals with one-shot identification. Our approach assigns a probability of animal occurrence to each pixel and does not solely rely on increased activation detection to infer that an animal is present at a specific image location. While our approach is not advantageous when animals are scattered and nicely captured in the image, it provides significant advantages when animals are very close to each other and have overlapping figures. It is also advantageous in cases when the animals are not completely visible due to occlusion caused by surrounding objects like trees and rocks or other animals and when the size and shape of the animals vary greatly. For example, models that rely on increased output values' detection to count animals suffer from substantial errors when dealing with over-sized or undersized animals. They also tend to miscount the animals when there is an occlusion or when the animals are very close to each other. Assigning a probability to the pixels adds awareness of the content of neighboring pixels. Because probabilities in a local density map are constrained by probability laws, the model is less prone to over-counting and under-counting, because each candidate animal identification must be converted to a distribution reflecting the content of the surrounding pixels. Even in the case of overlapping animal figures, the distributions get accumulated through the integration of probability maps that maintain meaningful scale and representation ability. On the contrary, one-shot models accumulate nearby high activations in an uncontrolled and unscaled manner, resulting in worse results when animals are very close to each other or vary in size. Such models often apply a smoothing operation on their results to avoid such problems, which is a less proper way to maintain a sense of scale than applying local density maps. Furthermore, these operations tend to mitigate the over-counting problem, but reinforce under-counting, because they do not consider local animal features.

The solution presented here made use of the Kaggle competition data set, but was not submitted to that competition. To verify the accuracy of the proposed model, we compared it with two different architectures: (a) the SSL Kaggle competition winning regression model [29] (named *Model-K*) and (b) Count-ception [30], a counting approach based on Inception modules and a fully convolutional network. Both models were trained exactly with the same training images and tested on the same test images. Table 2 shows a comparison of the RMSE and MAE scores for actual and predicted counts obtained by Model-2 (used in this paper) and the scores obtained by Model-K and Count-ception, using the same testing data sets.

Comparison with Model-K

The Model-K architecture is a regression model based on VGG16 without the feature extractor on top. The output layer was flattened and given as input to 2 fully connected (FC) layers with linear output. The regression model was designed to predict classwise (five categories) count. To compare it with the proposed solution, we modify the model by connecting the output layer with a fully connected one output neuron. Model-K was initialized with pre-trained Imagenet weights and then trained using our training data set with a Stochastic Gradient Descent (SGD) optimizer and an MSE loss function. The proposed Model-2 with EfficientNet feature extractor reached an RMSE value of 1.88 and 0.60 for the SLL and elephants' data sets, respectively, performing better than the Model-K with an RMSE of 2.17 and 0.81 for SSL and elephants' data set, respectively (Table 2). The Kaggle-winning Model-K gave a better classwise count (for SSL counting competition) but the proposed Model-2 was able to detect more animals and predict more accurate overall counting.

Comparison with count-ception

The Count-ception network [30] uses Inception modules to build a network for counting objects in an image. The model applies a fully convolutional architecture, and it does not use any pooling layers to retain as much information as possible. After each convolutional layer, batch

Table 2 Performance comparison of Model-2, Model-K, and Count-ception

Model	Feature extractor	SSL		Elephant		Parameters
		RMSE	MAE	RMSE	MAE	
Model-2	Eff.Net-B5	1.88	1.09	0.60	0.34	≈37M
Model-K	VGG	2.17	1.43	0.81	0.43	≈48M
Count-ception	*No*	5.57	3.54	1.59	0.84	≈14M

normalization and leaky ReLU activation are applied for speeding up convergence. The model takes an input image and outputs a prediction map. The predicted count is calculated using the following formula:

$$count = \frac{\sum_{x,y} F(I)}{r^2} \tag{6}$$

where $F(I)$ is the predicted map for the image I and $r = 32$ is the proposed receptive field size. Using our testing data set, Count-ception scored an RMSE value of 5.57 and 1.59 for SSL and elephants' data set, respectively (Table 2). Count-ception had a better counting accuracy for images having less overlap between objects (SSLs and African elephants), but the accuracy largely dropped when the overlap was high, i.e., when the animals were lying close together and/or close to the image boundary. The models proposed in this paper (i.e., Model-1 and Model-2), identified objects lying close to the boundaries more efficiently than Count-ception.

Visualization

Figure 5 shows 6 test images together with their actual and predicted density maps, as well as the total count. Images [a–b] are examples, where the difference between the actual and predicted count is small. In these images, Model-2 is not heavily affected by differences in illumination, occlusion, and overlapping. When animals cluster close to each other, the density maps superimpose. The resultant pixel value will be the sum of each pixel in the density maps, which greatly helps to overcome the overlapping issues. The model was able to perform well despite challenging conditions such as the presence of water and complex background information.

Images [c–d] show examples of a noticeable difference between the true and predicted counts. The main source of error in the SSL data set was attributed to juveniles, and pups being inherently difficult to detect because of their small size compared to other SSL types. Pups look like rocks in the background Fig. 6a and tend to be closer to female SSLs (most likely their mothers), which makes their detection difficult Fig. 6b. Similarly, elephant calves were comparably smaller Fig. 6c, and hidden by tree canopy Fig. 6d, were difficult to detect using the proposed model.

Conclusion

We proposed a method for automating animal census from aerial photography using deep learning together with density maps. The deep learning model is trained with minimal effort, using dot annotations placed on the centers of the animals as depicted on the images. The *UNet* semantic segmentation model is proposed, due to its high accuracy in segmentation-related computer vision problems and the relatively low computational cost. Using *EfficientNet* as a feature extractor architecture, lower RMSE values were achieved, based on data sets which included a variety of background complexities, illumination conditions and animal densities as well as high occlusion among the animals. The proposed method would help biologists survey the animals from images at a higher rate and accuracy with less resource, allowing them to focus on the conservation of animals.

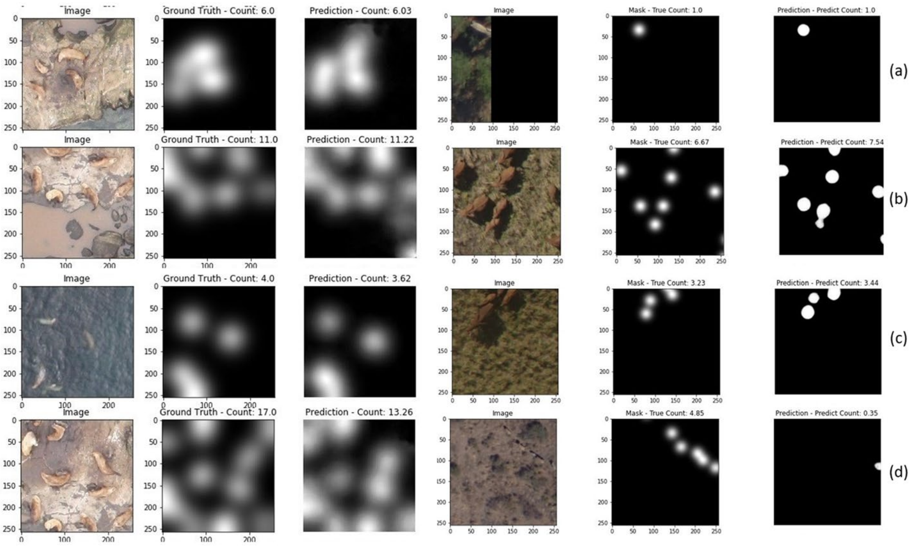
Image
Ground Truth - Count: 6.0
Prediction - Count: 6.03
Image
Mask - True Count: 1.0
Prediction - Predict Count: 1.0
(a)
Image
Ground Truth - Count: 11.0
Prediction - Count: 11.22
Image
Mask - True Count: 6.67
Prediction - Predict Count: 7.54
(b)
Image
Ground Truth - Count: 4.0
Prediction - Count: 3.62
Image
Mask - True Count: 3.23
Prediction - Predict Count: 3.44
(c)
Image
Ground Truth - Count: 17.0
Prediction - Count: 13.26
Image
Mask - True Count: 4.85
Prediction - Predict Count: 0.35
(d)

Fig. 5 Sample outputs: the ground-truth density map and predicted density maps for Model-2 with corresponding animal count for test images. From left to Right: Predicted Density Map; Ground-Truth Density Map; and Input Image

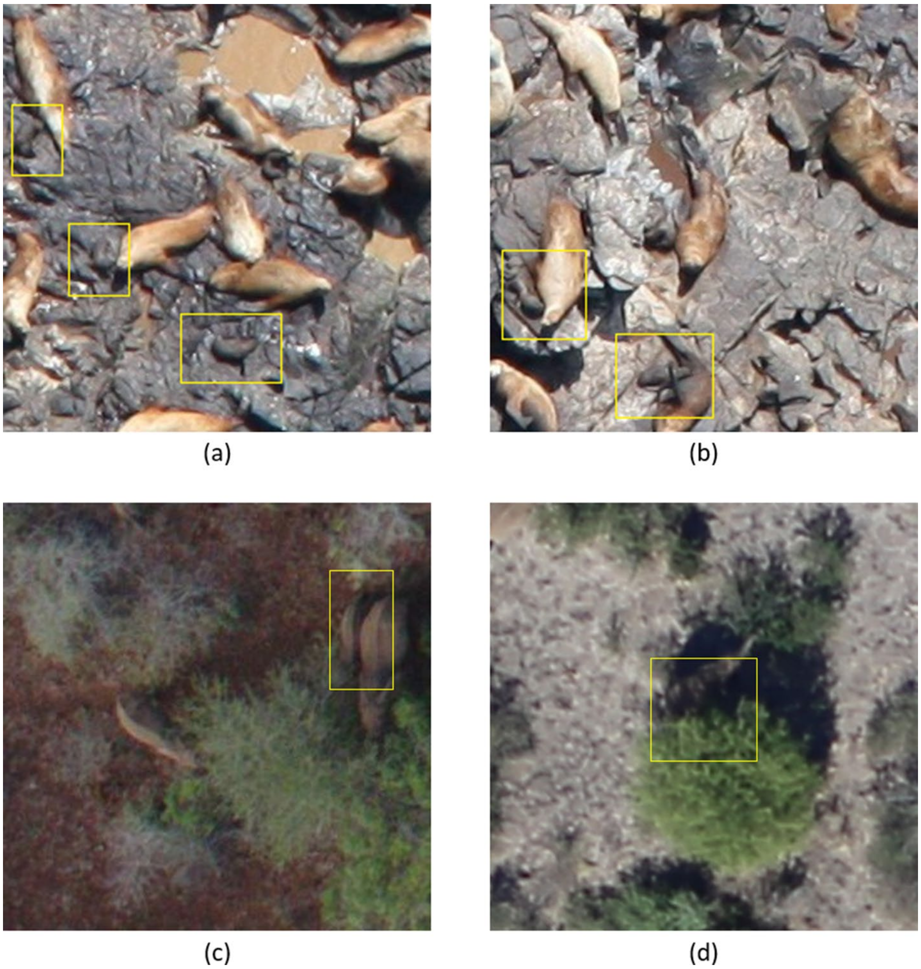

Fig. 6 Sample Images: **a** Pups looks similar to rocks. **b** Pups placed very close to female SSL, making it hard to detect. **c** Elephant calves **d** Elephant occlusion with tree canopy

The proposed method performed very well in the counting prediction problem, with a relatively low number of training parameters and minimal annotation. The principal sources of error were the generally high occlusion observed and the fact that small-sized animals were sometimes almost identical with background elements of the image. A comparison with other state-of-art models indicated that the method proposed in this paper performed better at the animal counting task. The evaluation of the model using two different data sets indicates that the proposed solution could be extended for counting other species' populations at large scale and at a fast pace, using remote sensing techniques such as aerial photography. The use of satellite/aerial images allows surveying remote species across vast areas. The proposed method will primarily benefit the biologging community with a faster and easier animal survey.

Future work

The authors will focus their future efforts on improving the detection accuracy of Model-2. Various optimization techniques will be considered and the generation of synthetic data will be investigated, especially to generate images of animals that are occluded and animals that look almost identical to their background. Finally, non-isometric Gaussian functions will be used to improve the alignment between the animals' positions and their corresponding density maps.

Abbreviations

NOAA: National Oceanic and Atmospheric Administration; CCNN: Count Convolutional Neural Network; SGD: Stochastic Gradient Descent; ReLU: Rectified Linear Unit.

Acknowledgements

Andreas Kamilaris and Savvas Karatsiolis have received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No. 739578 complemented by the Government of the Republic of Cyprus through the Directorate General for European Programmes, Coordination and Development.

Authors' contributions

Chirag Padubidri carried out the experiment under the guidance of Andreas Kamilaris and Savvas Karatsiolis. Chirag Padubidri wrote the manuscript with support from Andreas Kamilaris, Savvas Karatsiolis and Jacob Kamminga. Andreas Kamilaris supervised the project. All authors read and approved the final manuscript.

Availability of data and materials

The Steller sea lion data set and the African elephant data set used for this article are available in Steller Sea lion Data Set and African Elephant Data Set, respectively. The proposed implementation is available at Animal Counting using Density map

Declarations

Ethics approval and consent to participate

All the images of Steller sea lions were collected by the Alaska Fisheries Science Center, NOAA Fisheries under authorization of U.S. Marine Mammal Permit 18528 and IACUC A/NW 2016-3

Consent for publication

Not applicable.

Competing interests

All authors declare that they have no competing interests.

Author details

[1]Pervasive Systems, University of Twente, Enschede, Netherlands. [2]CYENS Center of Excellence, Nicosia, Cyprus.

Received: 27 August 2020 Accepted: 11 June 2021
Published online: 07 August 2021

References

1. Bar-On YM, Phillips R, Milo R. The biomass distribution on Earth. Proc Natl Acad Sci. 2018;115(25):6506–11. https://doi.org/10.1073/PNAS.1711842115.
2. Carrington D. Humans just 0.01% of all life but have destroyed 83% of wild mammals—study. The Guardian. Dated 21 May 2018 (accessed 27 July 2020)
3. Diaz S, Settele J, Brondizio E, Ngo HT, Gueze M, Agard J, Arneth A, Balvanera P, Brauman K, Butchart S, Chan K, Garibaldi L, Ichii K, Liu J, Subramanian SM, Midgley G, Miloslavich P, Molnar Z, Obura D, Pfaff A, Polasky S, Purvis A, Razzaque J, Reyers B, Chowdhury RR, Shin Y-J, Visseren-Hamakers I, Willis K, Zayas C. Summary for policymakers of the global assessment report on biodiversity and ecosystem services. Technical Report May 2019; 2019. https://www.ipbes.net/news/ipbes/ipbes-global-assessment-summary-policymakers-pdf
4. Kamminga J, Ayele E, Meratnia N, Havinga P. Poaching detection technologies—a survey. Sensors (Switzerland). 2018;18(5):1474. https://doi.org/10.3390/s18051474.
5. Atkinson S, Demaster D, Calkins D. Anthropogenic causes of the western steller sea lion eumetopias jubatus population decline and their threat to recovery. Mamm Rev. 2008;38:1–18. https://doi.org/10.1111/j.1365-2907.2008.00128.x.
6. Loughlin TR. The steller sea lion : a declining species. Biosph Conser Nat Wildl Hum 1998;1(2):91–8.https://doi.org/10.20798/biospherecons.1.2_91
7. U.S. Fish and Wildlife Service: Endangered Species Act; 1973. Available online: https://www.fisheries.noaa.gov/topic/laws-policies#endangered-species-act. Accessed 27 July 2020.
8. Chase MJ, Schlossberg S, Griffin CR, Bouché PJC, Djene SW, Elkan PW, Ferreira S, Grossman F, Kohi EM, Landen K, et al. Continent-wide survey reveals massive decline in african savannah elephants. PeerJ. 2016. Doi: https://doi.org/10.7717/peerj.2354.
9. Fisheries N. Steller sea lion survey reports; 2009-19. Available online: https://www.fisheries.noaa.gov/alaska/marine-mammal-protection/steller-sea-lion-survey-reports. Accessed 27 July 2020
10. Center, N.F.A.F.S.: NOAA Fisheries steller sea lion population count. Available online: https://www.kaggle.com/c/noaa-fisheries-steller-sea-lion-population-count/overview. Accessed 27 July 2020
11. Tian M, Hao G, Chen H, Wang Q, Long C, Ma Y. Automated pig counting using deep learning. Comput Electron Agric 2019;163:104840. Doi: https://doi.org/10.1016/j.compag.2019.05.049.
12. Rahnemoonfar M, Dobbs D, Yari M, Starek M. Discountnet: discriminating and counting network for real-time counting and localization of sparse objects in high-resolution uav imagery. Remote Sens. 2019. https://doi.org/10.3390/rs11091128.
13. Schmidhuber J. Deep learning in neural networks: an overview; 2014. CoRR arXiv:1404.7828.
14. Lecun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553):436–44. https://doi.org/10.1038/nature14539arXiv:1312.6184v5.
15. Kamilaris A, Prenafeta-Boldu FX. Deep learning in agriculture: a survey; 2018. CoRR arXiv:1807.11809
16. Kamilaris A., van den Brink C., Karatsiolis S. (2019) Training Deep Learning Models via Synthetic Data: Application in Unmanned Aerial Vehicles. In: Vento M. et al. (eds) Computer Analysis of Images and Patterns. CAIP 2019. Communications in Computer and Information Science, vol 1089. Springer, Cham. https://doi.org/10.1007/978-3-030-29930-9_8
17. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial nets. In: Proceedings of the 27th international conference on neural information processing systems. NIPS'14. MIT Press, Cambridge, MA, USA; 2014, vol. 2, p. 2672–2680.
18. Young-Chul Yoon K-JY. Animal detection in huge air-view images using cnn-based sliding window. In: International workshop on frontiers of computer vision (IWFCV); 2018.
19. Liu J, Gao C, Meng D, Hauptmann A. Decidenet: Counting varying density crowds through attention guided detection and density estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, p. 5197–206. https://doi.org/10.1109/CVPR.2018.00545.
20. Chattopadhyay P, Vedantam R, Selvaraju RR, Batra D, Parikh D. Counting everyday objects in everyday scenes. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR); 2017. p. 4428–4437.
21. Rahnemoonfar M, Sheppard C. Deep count: fruit counting based on deep simulated learning. Sensors (Basel, Switzerland) 2017. https://doi.org/10.3390/s17040905
22. Lempitsky V, Zisserman A. Learning to count objects in images. In: Lafferty JD, Williams CKI, Shawe-Taylor J, Zemel RS, Culotta A, editors. Advances in neural information processing systems 23. Curran Associates Inc; 2010. p. 1324–32. http://papers.nips.cc/paper/4043-learning-to-count-objects-in-images.pdf
23. Naudé JJ, Joubert D. The aerial elephant dataset Zenodo. 2019. https://doi.org/10.5281/zenodo.3234780.
24. Ronneberger O, Fischer P, Brox T. U-net: convolutional networks for biomedical image segmentation; 2015. CoRR arXiv:1505.04597.
25. Naude J, Joubert D. The aerial elephant dataset: A new public benchmark for aerial object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) Workshops; 2019.
26. Oñoro D, López-Sastre R. Towards perspective-free object counting with deep learning, vol. 9911. Berlin: Springer; 2016. https://doi.org/10.1007/978-3-319-46478-7_38.
27. Tan M, Le QV. Efficientnet: Rethinking model scaling for convolutional neural networks; 2019. CoRR arXiv:1905.11946.
28. Yakubovskiy P. Segmentation models. San Francisco: GitHub; 2019.
29. Kaggle: Use Keras to count Sea Lions, Kaggle. 2017. https://www.kaggle.com/c/noaa-fisheries-steller-sea-lion-population-count/discussion/35408
30. Cohen JP, Lo HZ, Bengio Y. Count-ception: counting by fully convolutional redundant counting; 2017. CoRR arXiv:1703.08710

Publisher's Note