Animal Species Recognition Using Deep Learning

Chapter · March 2020

DOI: 10.1007/978-3-030-44041-1_47

Mai Ibraheam
University of Victoria
4 PUBLICATIONS **13** CITATIONS
SEE PROFILE

Fayez Gebali
University of Victoria
329 PUBLICATIONS **3,476** CITATIONS
SEE PROFILE

Kin Fun Li
University of Victoria
223 PUBLICATIONS **1,901** CITATIONS
SEE PROFILE

Animal Species Recognition Using Deep Learning

Mai Ibraheam[1], Fayez Gebali[2], Kin Fun Li[3], Leonard Sielecki[4]
[1]Department of ECE, University of Victoria, Victoria, BC, Canada
maieelgendy@gmail.com
[2]Department of ECE, University of Victoria, Victoria, BC, Canada
fayez@uvic.ca
[3]Department of ECE, University of Victoria, Victoria, BC, Canada
kinli@uvic.ca
[4]British Columbia Ministry of Transportation and Infrastructure, Victoria, BC, Canada
Leonard.Sielecki@gov.bc.ca

Abstract. Wildlife-human and wildlife-vehicle encounters often result in injuries and sometimes fatalities. Thereby, this research aims to mitigate the negative impacts of these encounters in a way that makes the environment safer for both humans and animals. The proposed detection system is activated when an object approaches its field of vision, by the use of deep learning techniques, automated object recognition is achieved. For training, we use a labeled dataset from the British Columbia Ministry of Transportation and Infrastructure's (BCMOTI) wildlife program, and the Snapshot Wisconsin dataset as well. By using Convolutional Neural Network (CNN) architectures, we can train a system capable of filtering images from these datasets and identifying its objects automatically. Our system achieved 99.8% accuracy in indicating an object being animal or human, and 97.6% accuracy in identifying animal species.

1 Introduction

An estimated one to two million wildlife-vehicle collisions (WVC) occur in the United States each year [1]. These collisions represent a significant danger to human safety and wildlife survival. An estimated 200 people die from WVCs in the United States annually. The total annual cost associated with WVCs in the United States is calculated to be over $8 Billion USD [1]. There are several solutions to mitigate wildlife-vehicle collisions and wildlife-human encounters. These solutions range from the use of simple wildlife warning signs and wildlife exclusion fencing, to more effective but costly methods such as machine vision. Various modern technologies including wireless sensor network [2], ultrasonic acoustic waves [3], video cameras, passive infra-red (PIR) motion detector cameras [4], and Doppler radar sensors [5] that have been developed to assist in the detection of ground-based moving objects. Motion-detecting PIR cameras are popular tools for wildlife detection, due to their: (1) ease of use; (2) reliability; (3) ability to detect moving objects over a wide area; and (4) high image quality at low cost [6], [7].

In North America, PIR cameras need to operate under the adverse conditions of smoke, dust, wind, humidity, freezing temperatures, snow and heavy rain. Also, the white light flash or the noises made by some cameras may disturb animals. For these reasons, an appropriate camera, Reconyx Hyperfire™ PC900, has been selected for our research. This camera has a wide range of features suitable for our applications including: (1) weather resistant case to protect the camera during rough weather conditions; (2) invisible infrared flashes that will not disturb animals; (3) capable of capturing thousands of high definition images in both day and night as shown in (Fig. 2); (4) sensitivity settings that allow the sensitivity of the sensor to be adjusted; and (5) capturing information of time, date, temperature and moon phase integrated in image data [8]. These specifications make PC900 camera a powerful tool for our research.

Processing of images in a real-time fashion manually is impossible and inaccurate. Therefore, we need a model to recognize and classify objects using techniques such as machine learning, especially to deal with in low light conditions and blurry images.

In this research, we aim to: (i) use convolution neural network (CNN) to automatically extract significant features and to classify objects into different animal species or human using the two datasets from BCMOTI and Snapshot Wisconsin [16], and (ii) implement and evaluate this trained model under real-life conditions and improve classification accuracy.

2 Related Work

There have been many attempts to automatically identify animals in camera-trap images; however, most relied on manually designed features to detect animals [9], [10], while others used small datasets (few thousand images only for this application) [11], [12]. Yu et al. [11] manually cropped and selected images, which only contained the entire animal body. This conditioning allowed them to obtain 82% accuracy by using linear support vector machine (SVM) to classify 18 animal species. They used their own dataset which consists of over 7,000 camera-trap images from two different field sites. Several recent works used deep learning to classify camera-trap images.

Chen et al. [12] used a deep convolutional neural network (CNN) to classify 20 animal species in their own dataset of 20,000 images. The authors used an automatic segmentation algorithm (ensemble video object cut) for cropping the animals from the images and used these crops to train and to test their system. The convolution network which was used had 6 layers (3 convolutional layers and 3 max pooling layers) and it gave them a 38.31% accuracy. Gomez [13] used very deep CNNs to identify animal species in multiple versions of the Snapshot Serengeti dataset. This method reached 88.9% accuracy in the evaluation set. Norouzzadeh [14] used CNN and reported accuracies of 93.8% in classifying images that contain only a single animal. The performance matched human accuracy in their experiments.

Most of the published results used the publicly available Snapshot Serengeti dataset [15] which only contains African animals. There are more than one million sets of pictures, with each set containing three photographs. Before the release of the Snapshot Serengeti dataset, there was no publicly available and reliable dataset of animal photographs to work with.

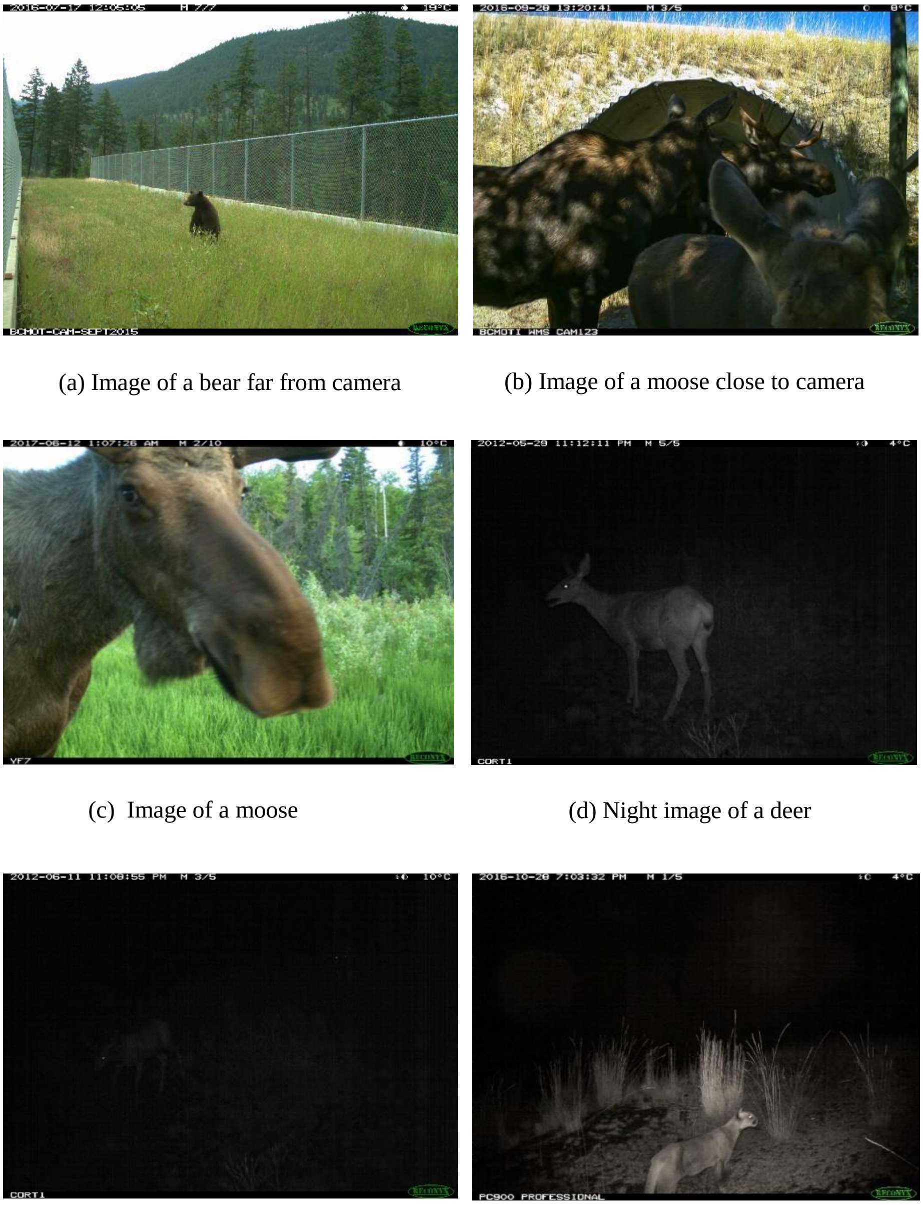
Our research differs from previous similar work as we aim to achieve high accuracy in object detection and animal species identification. In particular, the dataset used has images in a North American forest setting and near highway infrastructure in remote locations, and some images may contain more than one animal per image.

3 Datasets Used in Our Study

Automatic classification of animal species in PIR camera images is a challenging problem due to image conditions. In some instances, the animal covers only a small area of the field of view as shown in (Fig. 1 (a)). In other instances, the animal appears in most of the field of view as shown in (Fig. 1 (b)). Sometimes, only part of the animal is visible in the field of view as shown in (Fig. 1 (c)). Furthermore, different lighting conditions, shadows, and weather can make the feature extraction task even harder (Fig. 1 (d), (e)).

In our research we did not use the Snapshot Serengeti dataset because the wild animal species in Africa are different from the ones in North America. Instead, we used the dataset furnished by BCMOTI. This dataset has 50,000 images for seven different species: bear, moose, elk, deer, cougar, fox, and wolf, which satisfies our research goal. The dataset has to be balanced or uniformly distributed, meaning that every species is equally represented in the dataset. Imbalanced dataset is a problem for neural network techniques because it becomes heavily biased towards the classes with more images. In order to augment this relatively small dataset, we also used the Snapshot Wisconsin [16]. This dataset was collected in North America by using 1037 camera-traps placed in a forest in Wisconsin. It contains 0.5 million capture events for different animal species. In both datasets, a capture event contains three consecutive images. The images were labelled by volunteers. The images in both datasets are of high quality with resolutions ranging between 2048 × 1536 and 512 × 384 pixels.

In the Snapshot Wisconsin dataset, we chose six types of animals (bears, deer, elk, moose, wolf, and fox) since encounters between the larger species of animals and vehicles may lead to more severe crashes on highways. These animals are sometimes involved in tragic direct encounters with humans as well. Furthermore, these animals can be found in North America. BCMOTI and Snapshot Wisconsin differ in many aspects such as dataset size, camera placement, camera configuration, and species coverage, thus allowing one to draw more general conclusions.



(a) Image of a bear far from camera

(b) Image of a moose close to camera

(c) Image of a moose

(d) Night image of a deer

(e) Dark image of two deers

(f) Night image of a cougar

Fig. 1. Examples of BCMOTI dataset

4 Machine Learning for Animal Recognition

The large number of data from camera-traps necessitate the need for automated image processing. In case of animal recognition, we need to train learning algorithm which can process and identify objects (animal species and humans). In recent years, object classification has been dominated by CNNs, as demonstrated in recent ImageNet Large Scale Visual Recognition Challenges (ILSVRC), which have shown promising results [8], [17]. The ILSVRC has recently been the standard benchmark for evaluation of large scale image classification models by using CNN [20]. ILSVRC aims at three main tasks: image classification, single-object localization, and object detection from the ImageNet dataset [21].

Deep learning or deep neural networks are artificial neural networks with several layers of structure. The number of layers in an architecture is referred to as the depth of a network. CNN is a kind of deep neural network. It has a convolution layer which uses filters to convolve an area in the input image to a smaller area (to measure their spatial occurrence), and detects important or specific part within the area. CNN aims to extract the important part of the data, which makes it a perfect model for image classification [11], [18]. As shown in Fig. 2, CNN is essentially a sequence of layers which can be divided into two linked main parts. First, a feature extraction part which extracts local features from images and consists of convolutional layer plus non-linear activation function, usually the Rectifier Linear Unit (ReLU). New feature maps are obtained by pooling layer (sub-sampling), mostly using max pooling, which are then passed as input to the next layer. Finally, the fully-connected layers map the learned features (flat feature maps) to the output classes via a Softmax transformation function. This function converts the output node values to class probabilities, where the fully-connected layers are the output layer with predictions [19].

In standard neural networks, each neuron is fully connected to all neurons in the previous layer and the neurons in each layer are completely independent. When applied to high dimensional data such as natural images, the total number of parameters can reach millions, leading to serious problem in memory and computation time during training. By contrast, in CNNs, each neuron is connected only to a small region of the preceding layer, forming local connectivity [22], [23]. In addition, an important property of CNNs is parameter sharing, which reduces the number of parameters and computation complexity. Thus, compared to regular neural networks with similar size of layers, CNNs have much fewer connections and parameters, making them easier to train. The CNNs have three main characteristics: spatial structure, local connectivity and parameter sharing, which allow CNNs to convert an input image into layers of abstraction. The lower layers present detailed features of images such as edges, curves and corners, while the higher layers exhibit more abstract features of the object [24], [25]. Thereby, CNN is suitable for our research.

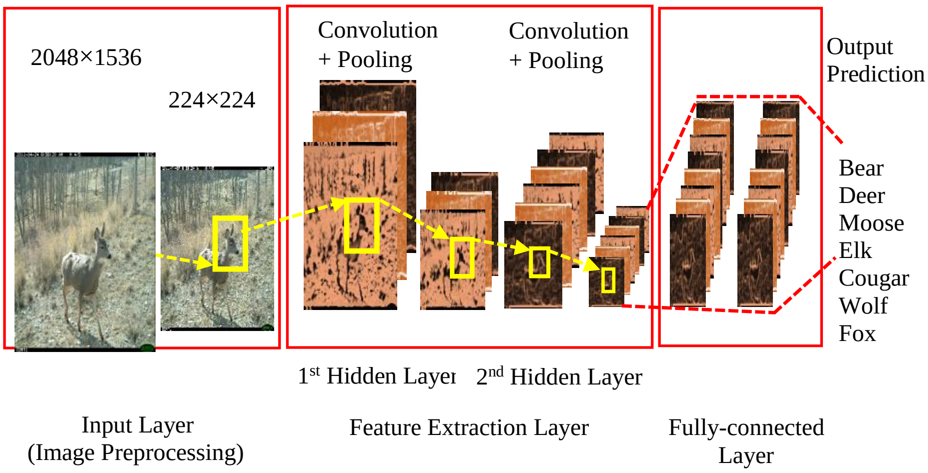


Fig. 2. Illustration of an example CNN architecture in animal recognition

5 Training for Animal Recognition System

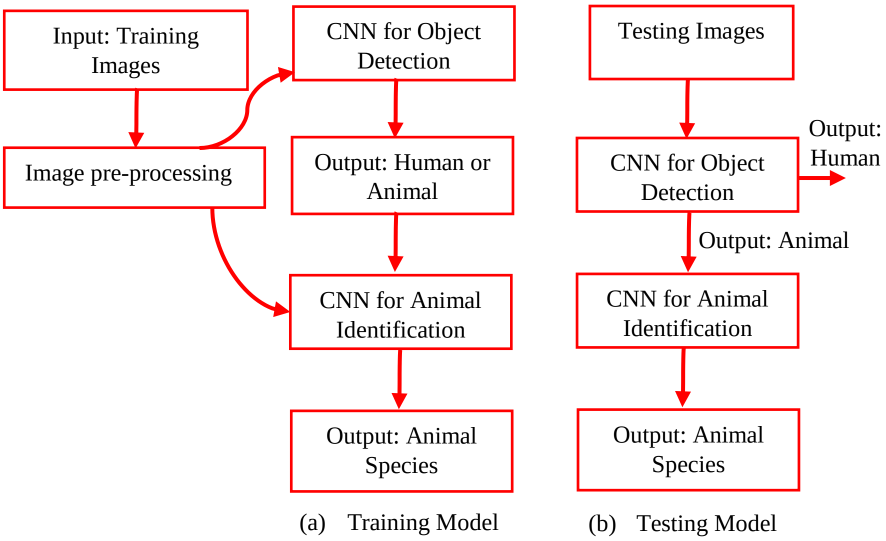
In this section, we present our proposed recognition system as shown in Fig. 3. It consists of two CNN image classification models. The first CNN model is designed to train a binary classifier for object detection: an animal or human; then another CNN model is constructed to train a multi-class classifier to produce the probabilities of the input image being one of the seven possible animal species (animal identification of bear, elk, deer, moose…).

By using two CNNs (‘cascade filtering’), we made our system to learn hierarchically (each subsequent layer is a convolution of the previous layer’s values). This means that the last layer in CNN has a higher level features. Thereby, the testing time is reduced due to the compression of the fully-connected layers parameters. The more reduction of non-significant parameters (features), the higher object identification accuracy is obtained. Therefore, the training of two different models is to closely resemble the process in real-time settings. We examined the training model on both the BCMOTI and Snapshot Wisconsin datasets. These two datasets have been split into 70% for training, 15% for validation, and 15% for testing. The models were trained on the training set, and fine-tuned on the validation set to reduce overfitting. When accuracy stopped improving on the validation set, then final results were reported using the test set.

The BCMOTI and Snapshot Wisconsin datasets have high dimensional images, while the input of CNN models require fixed dimension due to the existence of fully-connected layers. Thereby, all training images were normalized to 224×224 pixels.

Our CNN model consists of one input layer, three convolution and max pooling hidden layers, one fully connected layer and one output layer. There is no rules for selecting the number of layers. In each hidden layer, there are three main operations: convolution, max pooling and ReLU non-linear activation function. We kept adding convolution layers till we have reached the optimum, which is the best possible accuracy with minimal errors.

The updating of weight values of all layers is done by the propagation of errors from the output towards the input, and the soft-max function is used at the output layer in order to calculate the probability of animal species.



(a) Training Model (b) Testing Model

Fig. 3. Recognition system

6 Testing

6.1 Confidence Threshold

Since the CNN model for animal identification aims to distinguish seven classes (bear, moose, elk, deer, cougar, fox, and wolf), the output will be seven probabilities for each image. To predict the class of a particular image, we selected the class with the highest probability. This probability can be viewed as a confidence measure in the model's prediction [26]. The higher the predicted probability is, the more confident is the prediction. This characteristic enables us to use confidence thresholding on the final output probabilities, of which the model's decision is ignored if its highest probability is less than a certain threshold. The overall accuracy of the model is thus increased, by ignoring low confidence predictions.

The accuracy before confidence thresholding differs from one animal species to another, depending on the number of images we have in our dataset for each one, and the quality (brightness, clarity) of the images. The effect of confidence thresholding on prediction accuracy for the training and testing sets is shown in Fig. 4. If we set the threshold to 99%, the accuracies are between 93.2% and 97.6% (the accuracy has increased by almost 4%). Hence, the confidence threshold has a large effect on animal species accuracies as shown in Fig. 5. With respect to the dependency on the number of images, we have 100.000 images for deer from several angles, so the deer identification accuracy is almost 100%. On the other hand, we have 24.012 images for moose, thus we get 86.1% identification accuracy.

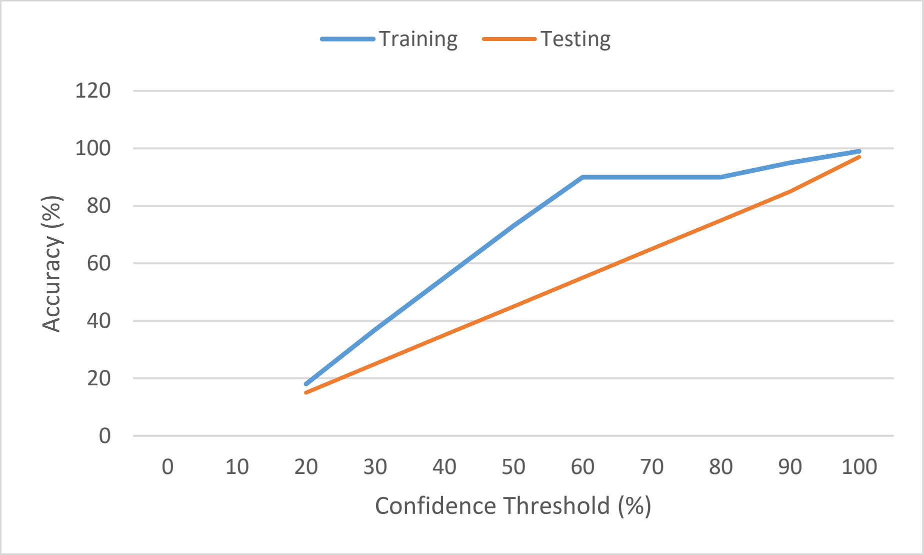


Fig. 4. Effect of applying different confidence thresholds on model's accuracy (testing three values of threshold at 20%, 60%, and 80%)

6.2 Accuracy

The accuracy was reported to evaluate our system as the percentage of correct prediction for images compared to the actual label. Accuracies of animal species varied according to the amount of training images (0.3 million) and the significant extracted features. Our Model achieved a classification accuracy of 99.8% and 97.6% on average which is reasonably good results for object detection (human vs animal) and animal identification respectively.

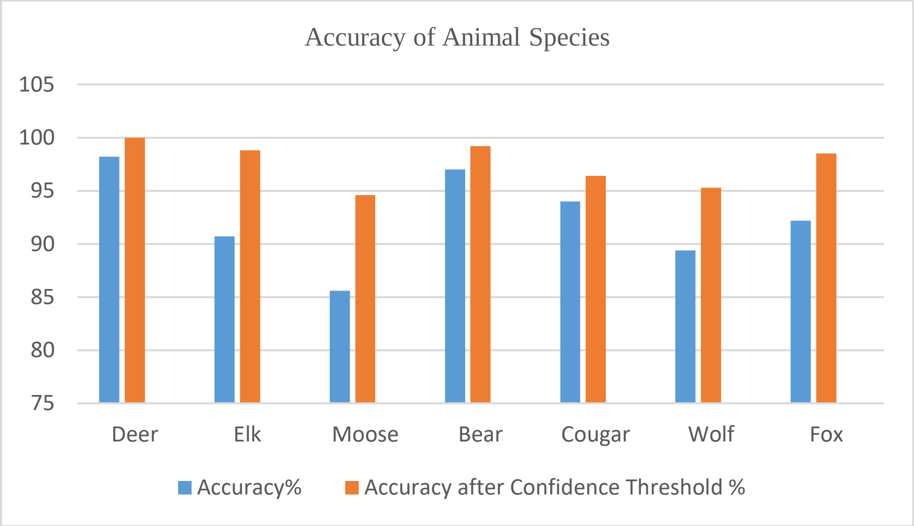


Fig. 5. Accuracies of animal species before and after confidence threshold

7 Discussion and Future Work

Our experimental results have shown that high accuracy of more than 96% can be achieved in detecting objects and identifying animal species when confidence thresholding is employed. The animal identification results have shown a good performance in identifying seven species of animals. While this performance may not yet be sufficient to build a fully automated recognition, there are many ways to improve the system accuracy and to reduce computation time in our future work by (i) training with more images (greater than the 0.3 million images used here); (ii) using an accurate segmentation algorithm to deal with only the regions of interest (small part of the image); and (iii) enhancing the quality (brightness, quality) of the data. Furthermore, it is very important to reduce the time (which was 9 sec. in this research) to detect animal in the field, so we can use this system in real-time applications.

8 Conclusion

In this paper, we first briefly explained our motivation for this project with the ultimate goal of reducing the number of wildlife-human encounters and wildlife-vehicle collisions. We reviewed previous research done in the same area. Then, we illustrated that object detection and wild animals' identification can be done using deep convolutional neural networks. The system has shown to be robust, stable and suitable for dealing with images captured from the forest using the BCMOTI and Snapshot Wisconsin datasets. We proposed and demonstrated the feasibility of a deep learning approach towards building automated animal recognition system. Our models achieved 99.8% accuracy in detecting objects (human or animal), and more than 96% accuracy in identifying seven animal species. We are working on alternative ways to improve the system's performance.

References

1. Huijser, M.P., P. McGowen, J. Fuller, A. Hardy, A. Kociolek, A.P. Clevenger, D. Smith and R. Ament. Wildlife-vehicle collision reduction study. Report to congress. U.S. Department of Transportation, Federal Highway Administration, Washington D.C., USA (2008)
2. Meek, P.D., Ballard, G.-A., Fleming, P.J.S.: The pitfalls of camera trapping as a survey tool in Australia. Australian Mammalogy 37, (2015)
3. Peter B. Nagy: Experimental Methods in the Physical Sciences. Volume 35, Science direct, (1999) 161-221
4. R. Kays, S. Tilak, B. Kranstauber, P. A. Jansen, C. Carbone, M. J. Rowcliffe, T. Fountain, J. Eggert,Z. He: Monitoring wild animal communities with arrays of motion sensitive camera traps. arXiv:1009.5718, (2010)
5. Jussi Kuutti, Mikko Paukkunen, Miro Aalto, Pekka Eskelinen, Raimo E. Sepponen: Evaluation of a Doppler radar sensor system for vital signs detection and activity monitoring in a radio-frequency shielded room. Volume 68, Science direct, (2015) 135-142
6. Hamel S., Killengreen S.T., Henden J.A., Eide N.E., Roed-Eriksen L., Ims R.A., Yoccoz N.G., O'Hara R.B.: Towards good practice guidance in using camera-traps in ecology: influence of sampling design on validity of ecological inferences. Methods in Ecology and Evolution 4, (2013)
7. Rovero, F., F. Zimmermann, D. Berzi, P. Meek.: Which camera trap type and how many do I need? A review of camera features and study designs for a range of wildlife research applications. Hystrix 24, (2013)
8. Swanson A. A., Kosmala M., Lintott C. C., Simpson R. R., Smith A., Packer C.: Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna. Volume 2, Scientific data, (2015) 150026
9. Swinnen KRR, Reijniers J, Breno M, Leirs H: A novel method to reduce time investment when processing videos from camera trap studies. PLoS One, (2014)
10. Figueroa K, Camarena-Ibarrola A, Garcia J, Villela HT: Fast automatic detection of wildlife in images from trap cameras. Progress in Pattern Recognition, Image analysis, Computer Vision, and Applications, Springer I nternational Publishing, Cham, Switzerland, (2014) 940-947
11. Xiaoyuan Yu, Wang Jiangping, Roland Kays, Patrick A. Jansen: Automated identification of animal species in camera trap image. EURASIP Journal on Image and Video Processing, (2013) 1-10
12. Chen, G., Han, T.X., He, Z., Kays, R., Forrester, T.: Deep convolutional neural network based species recognition for wild animal monitoring. IEEE International Conference on Image Processing (ICIP), (2014) 858-862
13. Gomez Al., Salazar A., Vargas F.: Towards Automatic Wild Animal Monitoring: Identification of Animal Species in Camera-trap Images using Very Deep Convolutional Neural Networks. arXiv:1603.06169v2, (2016)
14. Norouzzadeh M. S., Nguyen A., Kosmala M., Swanson A., Palmer M. S., Packer C., Cluen J.: Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. Proceedings of the National Academy of Sciences, (2018) 115, E5716–E5725

15. Chen G., Han T. X., He Z., Kays R., Forrester T.: Deep convolutional neural network based species recognition for wild animal monitoring. IEEE International Conference on Image Processing (ICIP), (2014) 858–862
16. https://dnr.wi.gov/topic/research/projects/snapshot/ (Snapshot Wisconsin)
17. Deng J., Dong W., Socher R., Li L.-J., Li K., Fei-Fei L.: ImageNet: A large-scale hierarchical image database. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2009) 248-255
18. LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., Jackel, L. D.: Backpropagation applied to handwritten zip code recognition. Neural Computation, (1989) 541–551
19. Bishop C. M.: Pattern recognition. Machine Learning, Volume 128, (2006) 1-58
20. Krizhevsky A., Sutskever I., and Hinton G. E.: ImageNet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems, (2012) 1097-1105
21. Russakovsky O., Deng J., Su et al. H.: ImageNet large scale visual recognition challenge. International Journal of Computer Vision, Volume 115, No. 3, (2015) 211-252
22. Simonyan K., Zisserman A.: Very deep convolutional networks for large-scale image recognition. Cornell University, (2014)
23. Gehring J., Auli M., Grangier D., Yarats D., Dauphin Y. N.: Convolutional Sequence to Sequence Learning. ArXiv e-prints, (2017)
24. Ciregan D., Meier U., Schmidhuber J.: Multi-column deep neural networks for image classification. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2012)
25. Gomez A., Diez G., Salazar A., Diaz A.: Animal identification in low quality camera-trap images using very deep convolutional neural networks and confidence thresholds. in International Symposium on Visual Computing, (2016)
26. Bank D., Greenfeld D., Hyams G.: Improved Training for Self Training by Confidence Assessments. arXiv:1710.00209v2, (2018)