Detection and Recognition of Badgers Using Deep Learning

Okafor, Emmanuel; Berendsen, Gerard; Schomaker, Lambertus; Wiering, Marco

Published in:
International Conference on Artificial Neural Networks

DOI:
10.1007/978-3-030-01424-7_54

Publication date:
2018

Detection and Recognition of Badgers Using Deep Learning

Emmanuel Okafor[1(✉)], Gerard Berendsen[2], Lambert Schomaker[1], and Marco Wiering[1]

[1] Bernoulli Institute for Mathematics, Computer Science, and Artificial Intelligence, University of Groningen, Groningen, The Netherlands
{e.okafor, l.r.b.schomaker, m.a.wiering}@rug.nl
[2] Twente Quality Centre (TQC), Enschede Area, The Netherlands

Abstract. This paper describes the use of two different deep-learning algorithms for object detection to recognize different badgers. We use recordings of four different badgers under varying background illuminations. In total four different object detection algorithms based on deep neural networks are compared: The single shot multi-box detector (SSD) with the Inception-V2 or MobileNet as a backbone, and the faster region-based convolutional neural network (Faster R-CNN) combined with Inception-V2 or residual networks. Furthermore, two different activation functions are compared to compute probabilities that some badger is in the detected region: the softmax and sigmoid functions. The results of all eight models show that SSD obtains higher recognition accuracies (97.8%–98.6%) than Faster R-CNN (84.8%–91.7%). However, the training time of Faster R-CNN is much shorter than that of SSD. The use of different output activation functions seems not to matter much.

Keywords: Image recognition · Object detection · Deep learning
Badger classification

1 Introduction

Badgers are short-legged omnivores and wild animals, and their existence is in danger in some parts of the world. To control this threat, some countries in Europe: United Kingdom, France, Republic of Ireland, Northern Ireland, and the Netherlands formed the Eurobadger collaboration with the objective to protect the existence of badgers. To assist this protection, there is a need to deploy computer vision systems that can aid in detecting and recognizing these animals, whose habitat is often a network of underground tunnels (setts). This paper describes the use of several deep neural network approaches to detect and classify different badgers.

Previous research [10] suggests that the human eye is an efficient and reliable method for animal detection. However, the effectiveness of the human eye reduces due to tiredness and a human is not able to focus on an animal for 24 h a day. Therefore, it is more efficient to apply computer-vision techniques for detecting and recognizing animals. Early research in [1] detects animal faces using Haar-like features and the Adaboost classifier, while tracking the animals was done using the Kanade-Lucas-Tomasi

V. Kůrková et al. (Eds.): ICANN 2018, LNCS 11141, pp. 554–563, 2018.
https://doi.org/10.1007/978-3-030-01424-7_54

method. Researchers have investigated different approaches to detect animals or humans: detection of humans in motion using background subtraction (BG) [2], using frame differences with the W4 algorithm [20], using background frame differences based on Gaussian functions [12], and the combination of BG and three-frame differencing [13].

Since the emergence of deep neural networks in the computer vision community, they have gained a lot of attention and successes for solving different learning tasks such as classification of objects, plants, and animals [11, 21, 6], classifying wild-animals [16], and recognizing cows with unmanned aerial vehicles (UAVs) using data-augmented images [18, 17]. Concerning wildlife monitoring and conservation, the authors in [3] investigated an automated detection and classification method of animals or non-animals using thermal images. Their method is based on the discrete cosine transform for feature extraction and k-nearest neighbors for classification. The research in [5] approaches wildlife monitoring using UAVs that use thermal image acquisition and a video processing pipeline to provide automatic detection, classification, and tracking of wildlife in a forest or open area. Recent research in [7] unites some scientists with the objective of monitoring wildlife. Their study showed that convolutional neural networks outperform a more classical technique based on the bag of visual words with a support vector machine in their wildlife detection challenge.

To the best of our knowledge, no research has been done concerning the detection and recognition of different badgers. The challenge is that some of the examined badgers have very similar color appearances, and therefore accurately discriminating the various badgers could be a difficult problem for computer vision algorithms.

Contributions: This research proposes the use of several object detection algorithms based on deep neural networks for detecting and recognizing badgers from video data. For this, a comparison is made between two neural network-based detectors: SSD [14] and Faster R-CNN [19]. SSD is combined with the Inception-V2 [9] or MobileNet [8] as a backbone and the Faster R-CNN detector is combined with either Inception-V2 or Residual networks [6] with 50 layers (ResNet-50) as feature extractors. Furthermore, we compare the use of two output activation functions: the softmax and sigmoid function. For the experiments, we use several videos recorded with a low-resolution camera. The results show that most of the trained SSD detectors significantly outperform the different variants of the Faster R-CNN detector. All the Faster R-CNN methods are computationally much faster than the SSD techniques for training the system, although for testing SSD is a bit faster.

Paper Outline: Section 2 describes the dataset used and the preprocessing steps. Section 3 explains the detection algorithms and the experimental setup for training the models. Section 4 presents the results. Section 5 concludes the paper and provides directions for future research.

2 Dataset and Preprocessing

The dataset is based on videos of different badgers collected by the foundation of Das and Boom[1]. The dataset contains four individual instances of badgers with a total number of 51 videos. The badger classes (identities) are: *badger_esp, badger_iaco, badger_looi,* and *badger_strik.* The badgers were recorded in 2016 and 2017 at the Badger Rescue Center of Das & Boom in the Netherlands. Additionally, some videos and photos were made at release locations for badger rehabilitation purposes. To identify each badger, they are micro-chipped, so the animal can be tracked during captivity and identified after release. The streaming lengths (Ts) of the videos vary in the range between 15 and 60 s. We extracted approximately a frame per second, for which we developed a script that extracts (Ts $\pm$ 2) video frames. We remark that some frames do not contain the existence of badgers and such frames are not used in our experiments. The details of the used dataset are shown in Table 1. Some example images of the used dataset are shown in Fig. 1.

Table 1. Dataset description

Dataset class	No. of videos	Dataset-Split (frames)		Ts (s)
		Train	Test	
Badger_esp	7	328	28	59
Badger_iaco	28	323	30	15
Badger_looi	9	437	61	59
Badger_strik	7	372	62	60
Total	51	1460	181	

We now describe how we made the ground truth annotations for the detection task. We used one video to create the images for the test set for each of the classes except for *Badger_iaco* where two videos are used in the test set. The remaining videos are used to create the train set. Manual extraction of the bounding box containing the existence of a badger was done using the LabelImg[2] tool. The used tool provides the annotation of a given image, and it is saved in the.xml file format. Each of the annotation files contains 4 coordinates representing the location of the bounding box surrounding the badger, the label and the file path to the images. We employed the Pascal VOC format.

[1] http://www.dasenboom.nl/.

[2] http://www.github.com/tzutalin/labelImg.

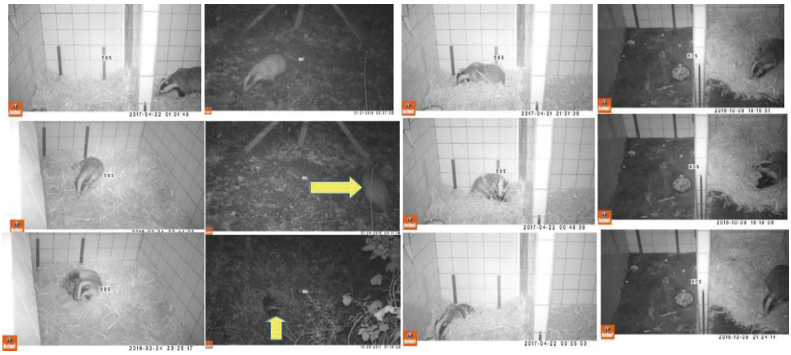

Fig. 1. Example images present in the Badger dataset; where each column represents: *Badger_esp*, *Badger_iaco*, *Badger_looi*, and *Badger_strik* respectively. The yellow arrows in column two indicate the existence of *badger_iaco* under poor illumination conditions (environment). Note that most videos were shot while the badgers were in captivity for a while, although some videos were shot in the wild.

3 Methods

This section describes the used deep neural network detection frameworks. Figure 2 shows the overall network pipeline that consists of data preprocessing as presented in Sect. 2, training the CNN to obtain the different detection models and their corresponding real-time deployment.

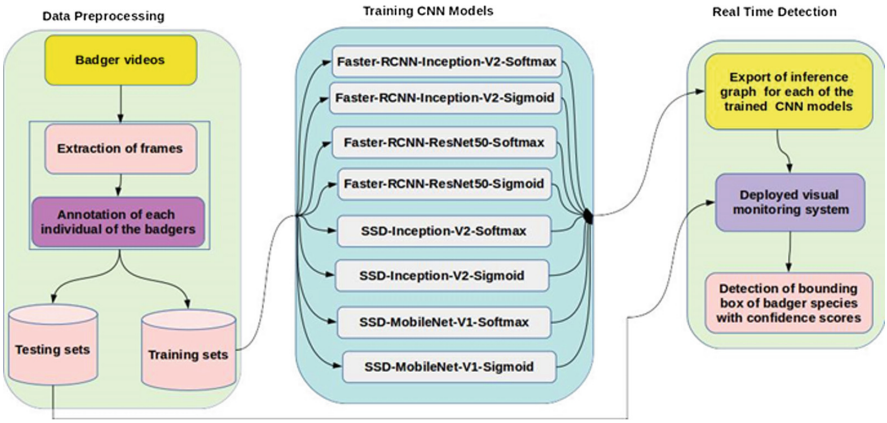

Fig. 2. Overall pipeline for the real-time detection systems; the first box accounts for the data-preprocessing, the second box represents the training of the CNN detection system, and the last box provides the network the inference generator and visual monitoring deployment system in the testing phase.

3.1 SSD with Inception-V2

The Single Shot multi-box-Detector (SSD) [14] is a detection framework that employs feed-forward convolutional neural networks for prediction of object classes and anchor offsets, with no consideration for second phase classification. Instead, it uses non-maximum suppression that allows the final detection of the objects in a single pass. A unique characteristic of this framework is that multi-scale convolutional bounding box outputs are attached to several feature maps at the top of the network layer. At the bottom or base portion of the network, the feature extraction method Inception-V2 [9] increases the breadth and depth of the network with a quite low computational complexity due to the used inception modules. The Inception-V2 extracts feature maps from the input images. The combination of SSD and Inception-V2 is called SSD-Inception-V2 [15]. We examine two forms of classification activation functions; sigmoid and softmax. This results in two variants of this approach.

Network Setup: we have trained the network using pre-trained weights ssd_inception v2_coco_2017_11_17, originally trained by a group of Google researchers. The pre-trained weights contain information from a subset of the Microsoft common object in context (COCO) dataset containing a total of approximately 328 K images with different object classes. We further trained the network using badger images with bounding boxes and class labels as input to the training algorithm. This use of pre-trained weights has the benefit of less training time compared to training random weights from scratch that demands longer computing times. During training of the network, we adopted a similar experimental setup as in [14] because it yields good performances. The network parameters include; the original input image frames contain 427×240 pixels and are resized online to 300×300 pixels, the convolutional box predictor uses a prediction dropout probability 0.8, kernel size 3×3 and a box-code size set to 4. The root mean square propagation (RMSprop) optimization algorithm is used for optimizing the loss functions trained for 40,000 steps using the following parameters; a learning rate of 0.004, decay factor 0.95, and decays at an interval of 16,000 steps. At the non-maximum suppression part of the network a score threshold of 1×10^{-8} is used with an intersection of union (IoU) threshold of 0.6, both the classification and localization weights are set to 1.

3.2 SSD with MobileNet-V1

This method also uses SSD [14] for detection while the MobileNet-V1 [8] as the base network is used as feature extractor. A MobileNet is a neural network-based feature extractor that employs depth-wise separable filters for extracting feature maps from a given image. The depth-wise separable convolution in this network involves the integration of depth-wise convolution and 1×1 point-wise convolution. The merit of this approach is that it reduces computational cost compared to standard convolution [8]. The output from the MobileNet is further processed using SSD. The method is referred to as SSD-MobileNet-V1. Additionally, we consider two forms of classification activation functions: sigmoid and softmax. This results in two variants of this method.

Network Setup: we have trained the network using pre-trained weights ssd_mobilenet_v1_coco_2017_11_17 from the COCO dataset as was explained in the previous subsection, and further trained our custom network using the badger images as input to the SSD-MobileNet-V1 system. The training process uses similar hyperparameters as described in the previous subsection.

3.3 Faster R-CNN with ResNet-50

The Faster R-CNN algorithm [19] is an improvement of Fast R-CNN [4]. In this system, the working operation of the Faster R-CNN involves two phases. The first phase requires the use of a region proposal network (RPN) which allows concurrent prediction of object anchors and confidence (objectiveness) from some intermediate layers. Note that a feature extraction network can be used for this purpose, in this case, a residual network with a depth of 50 layers (ResNet-50) [6] is used. The second phase requires information from the first phase to make an accurate prediction of the class label and its bounding box refinement. Additionally, we made consideration of the classification activation functions that were earlier discussed in the previous subsections. Hence this results in two variants of this network.

Network Setup: We have trained the network using pre-trained weights faster_rcnn_resnet50_coco_2018_01_28 from the COCO dataset. The training of the network factored in some modified experimental setups as in [19]. The original input image (badger) to the network contains 427 $\times$ 240 pixels and is resized online with an aspect ratio of min-max dimensions [600, 1024] during training. As earlier discussed the network comprises of two phases. The first phase initiates a grid-anchor of size 16 $\times$ 16 pixels with scales [0.25, 0.5, 1.0, 2.0], a nonmaximum-suppression-IoU-threshold set to 0.7, the localization loss weight 2.0, objectiveness weight 1.0 with an initial crop size of 14 $\times$ 14 pixels, kernel size 2 $\times$ 2 with strides set to 2. The second phase computes the prediction score with the IoU-threshold set to 0.6; the SGD optimizer optimizes the loss functions using an initial learning rate 0.0002 and momentum value 0.9. Again, the network was trained for 40,000 steps.

3.4 Faster R-CNN with Inception-V2

The Faster R-CNN detector employs an Inception V2 feature extractor for extracting useful feature maps from an input image. The intermediate layer from the Inception module uses the RPN component of the network for prediction of object anchors and confidences. Similar procedures as explained in [19] were followed.

Network Setup: we have trained the network using pre-trained weights faster_rcnn_inception_v2_coco_2018_01_28 from the COCO dataset. The training of our custom network employs the badger images as input to the Faster-RCNN-InceptionV2 system. The training process uses similar hyperparameters as described in the previous subsection.

All the experiments were carried out using the Tensorflow object detection API framework on a Ge-Force GTX 960 GPU model, and the operating system platform employed is Ubuntu 16.0. We modified the deployment script in the Tensorflow object

detection API, by providing the possibility to evaluate all images in the test directory instead of applying restrictions. Moreover, we also use our own script to compute the performance index metrics of the used methods. The next section discusses the performance and overall training time for each of the methods.

4 Experimental Results

The overall training time for each of the used methods is reported in Table 2. The table shows that the training time of Faster R-CNN is much shorter than the training time of SSD, and the use of Inception-V2 leads to the shortest training times. The frame rates show that most of the methods can analyze 0.8–1.5 images per second using our hardware, and SSD is a bit faster than Faster R-CNN for deployment.

Table 2. Average time evaluation for the different detection systems

Methods (CNN models)	Training time	Time improvement	Testing time	Frame rate (f/s)
Faster_RCNN-Inception_V2_Sigmoid	3 h, 21 m	×3.0	222 s	0.82
Faster_RCNN-Inception_V2_Softmax	3 h, 23 m	×2.9	211 s	0.86
Faster_RCNN-ResNet-50-Sigmoid	5 h, 37 m	×1.4	268 s	0.68
Faster_RCNN-ResNet-50-Softmax	5 h, 44 m	×1.3	267 s	0.68
SSD_Inception_V2_Softmax	10 h, 45 m	×0.24	162 s	1.12
SSD_Inception_V2_Sigmoid	10 h, 46 m	×0.24	163 s	1.11
SSD_MobileNet_V1_Softmax	13 h, 16 m	×0.01	120 s	1.51
SSD_MobileNet_V1_Sigmoid (Baseline)	13 h, 21 m	− − −	122 s	1.48

We have carried out two experimental runs and computed the average precision, recall and accuracy, based on the predicted class label in a detected box. The standard deviations for all the methods are $\leq 1.4\%$, which indicates that the performances of the techniques are consistent. The summary of the average performance indices and the standard deviations for each of the methods is presented in Table 3. From this table, we draw the conclusion that SSD-Inception-V2 for both output functions and the SSDMobileNet-V1-Sigmoid outperforms all the Faster R-CNN variants with $p < 0.05$ significance level.

The performance index from the SSD-network variants provides a more precise detection compared to the Faster R-CNN network variants. The lower precision in the

Table 3. Average performances for the different detection and recognition systems

Methods (CNN models)	Performance index		
	Precision	Recall	Accuracy
SSD_Inception_V2_Softmax	0.988 ± 0.012	0.986 ± 0.014	0.986 ± 0.014
SSD_Inception_V2_Sigmoid	0.986 ± 0.003	0.986 ± 0.004	0.986 ± 0.004
SSD_MobileNet_V1_Sigmoid	0.985 ± 0.005	0.983 ± 0.006	0.983 ± 0.006
SSD_MobileNet_V1_Softmax	0.978 ± 0.011	0.978 ± 0.011	0.978 ± 0.011
Faster_RCNN-Inception_V2_Softmax	0.942 ± 0.009	0.917 ± 0.011	0.917 ± 0.011
Faster_RCNN-Inception_V2_Sigmoid	0.945 ± 0.003	0.914 ± 0.008	0.914 ± 0.008
Faster_RCNN-ResNet-50-Sigmoid	0.936 ± 0.000	0.890 ± 0.000	0.890 ± 0.000
Faster_RCNN-ResNet-50-Softmax	0.921 ± 0.003	0.848 ± 0.003	0.848 ± 0.003

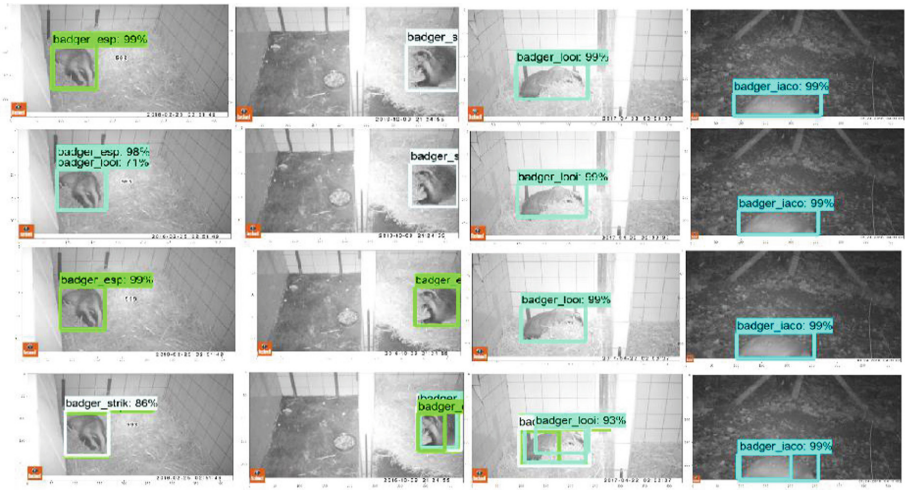

Fig. 3. Testing detection confidence prediction of the badger individual instances using different neural network detection methods: the first row indicates detection using ssd_mobilenet_v1_-softmax, the second row shows the detection using ssd_inception_ v2_sigmoid, the third row shows the detection using faster_rcnn_inception_v2_softmax, and the last row shows the detection using faster_rcnn_resnet50_sigmoid. Note that each of the columns represents the badger individual instances in the order; *Badger_esp, Badger_strik, Badger_looi,* and *Badger_iaco* respectively. (Color figure online)

Faster R-CNN may have arisen due to localization bias problems. Figure 3 shows some examples of the detection scores of badgers within a given image during testing evaluation. From this figure, we observe that the Faster R-CNN methods misclassified this particular example of badger_strik (gray box) as badger_esp (green box) as shown in sub-images within cells (3, 2) and (4, 2). Hence, this explains the lower performance index using the Faster R-CNN methods compared to the SSD network variants. From an application standpoint, it could be profitable to use Inception-V2 as the backbone for the SSD detector since it presents more precise detections of the objects of interest.

Additionally, the results suggest that SSD-based networks are useful in handling localization bias problems.

5 Conclusion

Real-time detection using deep learning can be used for many localization and identification tasks. In this paper, several deep neural networks were used to detect and classify different badgers using a novel animal dataset. We compared the single shot multi-box detector (SSD) combined with Inception-V2 or MobileNet, to faster-region-based convolutional neural network (Faster R-CNN) combined with Inception-V2 or residual networks (ResNet). We used the pre-trained networks and further trained them on our dataset. The four detectors were combined with either a softmax or sigmoid function for computing the output probability scores, hence resulting in eight different models.

The results showed that SSD with the Inception-V2 as a backbone obtains the highest mean accuracy performance (98.6%). Furthermore, we noticed that during testing, SSD has a higher frame rate than Faster R-CNN, although its training time is longer. Our analyses suggest that the examined SSD methods tackle the problem of localization bias much better than Faster R-CNN during prediction of the bounding boxes. Finally, we noticed that the use of the sigmoid or softmax output activation functions led to comparable results.

Future work will be directed at the scalability in the number of classes and environments, using a much larger dataset. We also suggest that the best found model, SSD-Inception-V2-Softmax, could be improved and deployed into UAVs or thermal acquisition cameras, as this can help to detect badgers in environments where they are endangered.

References

1. Burghardt, T., Calic, J.: Real-time face detection and tracking of animals. In: 8th Seminar on Neural Network Applications in Electrical Engineering, NEUREL 2006, pp. 27–32. IEEE (2006)
2. Chen, P.: Moving object detection based on background extraction. In: International Symposium on Computer Network and Multimedia Technology, CNMT 2009, pp. 1–4. IEEE (2009)
3. Christiansen, P., Kragh, M., Steen, K.A., Karstoft, H., Jørgensen, R.N.: Platform for evaluating sensors and human detection in autonomous mowing operations. Precis. Agric. **18**(3), 350–365 (2017)
4. Girshick, R.: Fast R-CNN. arXiv preprint arXiv:1504.08083 (2015)
5. Gonzalez, L.F., Montes, G.A., Puig, E., Johnson, S., Mengersen, K., Gaston, K.J.: Unmanned aerial vehicles (UAVs) and artificial intelligence revolutionizing wildlife monitoring and conservation. Sensors **16**(1), 97 (2016)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

7. He, Z., Kays, R., Zhang, Z., Ning, G., Huang, C., Han, T.X., Millspaugh, J., Forrester, T., McShea, W.: Visual informatics tools for supporting large-scale collaborative wildlife monitoring with citizen scientists. IEEE Circ. Syst. Mag. **16**(1), 73–86 (2016)
8. Howard, A.G., et al.: Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
9. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
10. Koik, B.T., Ibrahim, H.: A literature survey on animal detection methods in digital images. Int. J. Futur. Comput. Commun. **1**(1), 24 (2012)
11. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
12. Liu, H., Hou, X.: Moving detection research of background frame difference based on Gaussian model. In: 2012 International Conference on Computer Science & Service System (CSSS), pp. 258–261. IEEE (2012)
13. Liu, H., Dai, J., Wang, R., Zheng, H., Zheng, B.: Combining background subtraction and three-frame difference to detect moving object from underwater video. In: OCEANS 2016-Shanghai, pp. 1–5. IEEE (2016)
14. Liu, Wei, et al.: SSD: single shot multibox detector. In: Leibe, Bastian, Matas, Jiri, Sebe, Nicu, Welling, Max (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
15. Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T., Omata, H.: Road damage detection using deep neural networks with images captured through a smartphone. arXiv preprint arXiv:1801.09454 (2018)
16. Okafor, E., et al.: Comparative study between deep learning and bag of visual words for wild-animal recognition. In: 2016 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–8. IEEE (2016)
17. Okafor, E., Schomaker, L., Wiering, M.A.: An analysis of rotation matrix and colour constancy data augmentation in classifying images of animals. J. Inf. Telecommun., 1–27 (2018)
18. Okafor, E., Smit, R., Schomaker, L., Wiering, M.: Operational data augmentation in classifying single aerial images of animals. In: 2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA), pp. 354–360. IEEE (2017)
19. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
20. Sengar, S.S., Mukhopadhyay, S.: Moving object detection based on frame difference and W4. Signal, Image Video Process. **11**(7), 1357–1364 (2017)
21. Szegedy, C., et al.: Going deeper with convolutions. In: CVPR (2015)