KICS The Korean Institute of Communications and Information Sciences

Available online at www.sciencedirect.com

ScienceDirect

ICT Express 8 (2022) 640–645

www.elsevier.com/locate/icte

# Neural network model approach for automated benthic animal identification

Ravail Singh[a,*], Varun Mumbarekar[b]

[a] Indian Institute of Integrative medicine, Canal Road, Jammu, India
[b] Electronics Department Vishwakarma Institute of Technology, 411037, Pune, India

Received 28 August 2020; received in revised form 22 November 2020; accepted 10 March 2021
Available online 23 March 2021

**Abstract**

The most tedious and hectic job is to identify the tiny benthic animals by spending thousands of hour under the microscope, since all the fauna need to be counted, sorted, picked and permanently mounted on glass slides for taxonomic identification. All faunal identifications need a lot of preprocessing and it consumes a lot of time to identify a single specimen. Therefore, to reduce the complexity of many such procedures, combined with the desire to identify larger datasets, we came up with new software based on artificial intelligence which can automatically identify the benthic fauna through the microscopic images. In this paper, we propose a machine learning method for automatic visual identification through the images of the benthic fauna. To this end, we propose a neural network model, where we demonstrate that the proposed approach differentiates the fauna based on images. However, it works well with vast amounts of image data and significant computational resources.


*Keywords:* Machine learning; Model; Automated identification; Benthos

## 1. Introduction

The identification of individuals is extremely important for the basic understanding about the components of biodiversity about their conservation and sustainable use. The most classical way of identification is based on the morphological features of specimen [1–3]. However, morphological and molecular taxonomic classification has many drawbacks since it is easier to spot difference between two bigger organisms like camel and cow but it is very difficult to distinguish the microscopic insects or worms [4]. Another drawback of classical taxonomy is the amount of effort needed to determine one individual that varies significantly between taxa. There are many morphological characters which can remarkably contribute in the difference which includes shape of the different parts of the body, size and color of the animals. However, it gets more difficult in case of small and fragile specimens of benthic worms, like megabenthos, macrobenthos, meiobenthos and microbenthos [5]. To further identify one need to examine all the morphological features like mouth shape, cuticle, setea, tail shape) and moreover needs to understand the whole taxonomic terminology to provide precision in the observation of specific characters. Therefore this entire procedure consumes lot of time and manpower. The complexity of many such procedure, combined with the desire to identify larger datasets, there is a growing need for alternative methods like machine learning [6,7]. To minimize this lengthy procedure there has been long pending demand to develop automated image-based systems for identification. Recently, machine learning has emerged as a potential solution for automatic identification through images [8]. Machine learning and computer vision is like an algorithm or a manual procedure, to extract relevant features for identification from the images of the organisms.

In this paper, we focused on harnessing computer vision to automatically identify the meio and macro benthic fauna. Here, we prepared large images database of macro and meio benthic organism collected from the marine regions of west coast of India. We trained and tested a machine learning convolution neural network (CNN) model that can use images database to classify the benthic fauna and provide the score of accuracy. The images may even represent different views, angles, body

* Corresponding author.
*E-mail address:* ravail.singh@iiim.res.in (R. Singh).
Peer review under responsibility of The Korean Institute of Communications and Information Sciences (KICS).

https://doi.org/10.1016/j.icte.2021.03.003

**Table 1**
Recall and Precision metrics for the trained model.

| Group name | Num of training images | Num of test images | Recall | Precision | FP rate | FN rate |
|---|---|---|---|---|---|---|
| Amphipoda | 150 | 30 | 0.9833 | 0.9725 | 0.005 | 0.003 |
| Bivalvia | 150 | 30 | 1.00 | 1.00 | 0 | 0 |
| Isopoda | 150 | 30 | 0.9722 | 0.9831 | 0.003 | 0.005 |
| Nematoda | 150 | 30 | 1.00 | 0.9729 | 0.005 | 0 |
| Polychaeta | 150 | 30 | 0.9944 | 1.00 | 0 | 0.001 |
| Nemertea | 150 | 30 | 0.9778 | 1.00 | 0 | 0.004 |

parts images, or life stages—the CNN automatically finds the relevant set of features for the task at hand.

The following are the aims of this paper:

To build the automatic identification program which aids in instant identification of benthic organisms, targeting group level and then at higher taxa level identification.

To extract the ecological information, diversity, abundance, presence of young, adult and sex ratio of animals which are important for ecologists to have proper scientific interpretation?

## 2. Database creation

For the database creation, we collected numerous images of various benthic groups using a digital microscope (Fig. 1). Images were extracted from our own samples collected from coastal regions of west coast of India. In the beginning we also used the images from other webpages like nemys.ugen.de, macrobenthos of the North Sea-Polychaeta, Marine portal. The digital microscope used for this activity was Olympus Binocular Microscope (BX-63) connected to a personal desktop. In this work, we used multiple images of specimen from different angles. We collected almost 150 images from each set of groups/family/genus/species. The result of total images that included number of training and test images, their recall and precision metrics is given in Table 1.

In this study, we collected 2 different datasets, one for 'species classifier' which was step 1 of the experiment. The other dataset was specifically collected for 'species detection' (or localization). The classifier will predict the species with its confidence level in a given image. Whereas, the detector will locate and draw a bounding box around the species and correctly label it. The detector is the second step of the experiment which will be specifically used to detect multiple species in a single image and label (i.e direct observation).

For training the detector, the second set of image data was accumulated. The dataset had sediment containing multiple species and its corresponding location. For the detector, we manually noted the position of each species in each image so that the positional data along with the associated image can be fed during the training. There is no need to collect the positional information for training a classifier, as the objective of the classifier is just to predict which species is present in the given image and not to locate the position of the species. But providing positional data is compulsory for training the detector model, as the output of the model will provide two parameters viz. position of the detected species and prediction

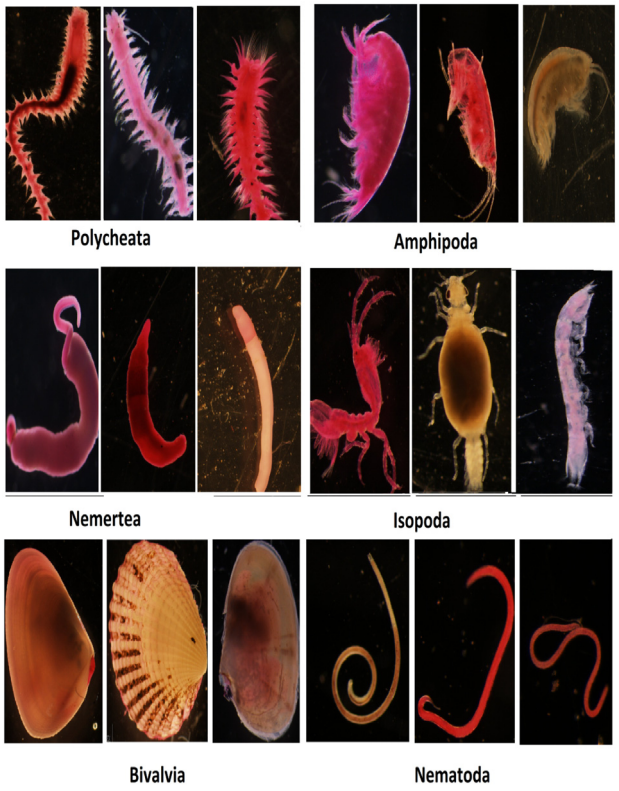


**Fig. 1.** Sample images of each of the organisms for classifier and detector.

of the species name at that location. While training, the detector model will try to learn the patterns and features of the provided location and species name and eventually will detect and locate correctly once trained.

Image characteristics: Most of the image dataset for the classifier was captured using the digital microscope. All the images collected have a uniform background (black) with high contrast. The other images fetched from webpages mentioned previously also have high contrast with uniform background. The second dataset, for the detector model, contains the multiple target species placed on the sediment. The dataset was improved by capturing the images from different angles for each organism.

Pre-processing: The image dataset contained all three color channels (red, green, and blue). Most of the images were rotated to make the dataset robust. Since the input to the VGG16 model is of shape $224 \times 224$, all the images were resized to this size. As all the images had different aspect ratios, resizing

the images would cause distortion or introduce bias while training and might hamper the real accuracy. For preserving the aspect ratio, we calculated the average dimension (average height and width) of the images. And then resized such that one dimension of all the images would be the average size. So the other dimension was either cropped or interpolated with calculated values of pixels in-order to resize it to the desired dimension.

Training and testing set: The image dataset was divided into 2 subsets for training and testing. The ratio of testing images from the total images was 1:10. So nine parts of the image dataset were used for the training procedure and the remaining one part of the dataset was used to validate the accuracy. The segregation of the test and training images was done such that the proportion of the categories was the same in the test and training sets.

### 2.1. Model

Classifier: For building the classifier we used a pre-trained InceptionV3 model by changing the last layers. But before moving directly to this model, initially, some experiments were performed on a locally built classifier in-order to understand the complexity of the features. Initially, we started with 2 layered convolutional layers followed by a fully connected layer. For this classifier, we achieved not so great accuracy even for training data. Further, we added more convolutional layers followed by maxpooling layers. The results for all these experiments are added in the later section. The addition of new layers improved the training as well as testing accuracy. The testing accuracy was not increasing above 80%. The training was run with a total of 50 epochs and the testing accuracy was hitting 100% while training, it was noticed that after a few epochs, the testing loss was gradually increasing. This is the indication for the overfitting of training data. The model overfits the training samples and hence starts predicting inaccurate output for the new unseen data. To overcome this issue, dropouts were added to the neural network, which reduced the overfitting to a certain extent. The dropouts try to randomly ignore some of the neurons in the neural network during the training process which makes the training noisy and makes certain neurons of a layer more or less responsible for the inputs. In addition to that, we performed image augmentation using Keras library, where-in the augmentation is performed on-the-fly without altering the dataset which is stored in the memory. After implementing the mentioned techniques, the overfitting problem was resolved. Although now, the performance of the model was better, it did not increase the testing accuracy much.

Using TensorFlow, one can easily modify some layers of a pre-trained model and retrain it for a specific classification. We used the InceptionV3 model, which is a 48 layer deep neural network. Instead of using the full architecture, an output of a mixed layer (‘*mixed7*’) was fed to a custom-built fully connected layer followed by softmax function. The weights of the pre-trained model were frozen during the training, as those layers were already trained. Only the custom neural network was trained to learn the problem statement. The implementation was done using Keras with the TensorFlow backend.

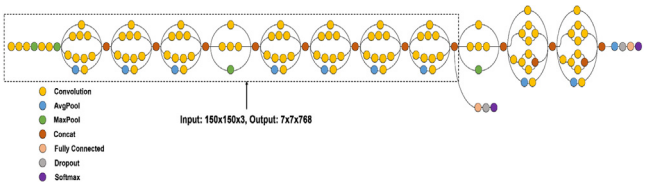


**Fig. 2.** Neural network model architecture.

The input to the model was provided with a 150 × 150 image with all the 3 color channels. The output layer (‘*mixed7*’) which was selected as input to the custom neural network has a shape of 7 × 7 × 768 which is then flattened before feeding it to the fully connected layers [9]. The final layer among them provides several channels which are equal to the number of classes to classify. The softmax layer normalizes the output vector using the basic softmax formula mentioned below. Otherwise, all the convolutional blocks use the ReLU activation function (see Fig. 2).

The InceptionV3 model is used for feature extraction and a new fully connected layer is added to one of the intermediate layer

Detector: For detector, we used mobilenet as the base feature extractor and SSD (Single Shot Multibox Detector) for detecting multiple classes in a single image. The reason behind using the mobilenet as a feature extractor is to achieve faster speed. The mobilenet is lightweight in terms of computation as compared to the other models. It optimizes the computations by replacing normal convolution with ‘depthwise separable convolutions’ [10]. The regular convolution applies convolutional kernel to all the channels and performs a weighted sum of the pixels covered by the kernel across the input channels. So basically, any number of input channels will transform to single-channel output (practically, we apply multiple kernels and thus the output has multiple channels). In mobilenet architecture, only the first layer uses standard convolution. The other layers use depthwise separable convolution. This is the combination of depthwise convolution and pointwise convolution as shown in Fig. 3. Unlike the standard convolution, it does not combine the channels but it performs convolution on each channel individually. As shown in Fig. 3, the input image with 3 channel will output an image with 3 channels with each channel having its own set of weights. This depthwise convolution works like filtering of the input channels. This is followed by pointwise convolution which is like a normal convolution with 1 × 1 kernel. This adds up (weighted sum) all the channels. The main purpose of the pointwise convolution is to create a new feature by combining the output channels of the depthwise convolution. A standard convolution performs filtering and combining in a single execution while the depthwise separable convolution does this in two steps. But the standard convolution has to perform more computations as it needs to tune more weights.

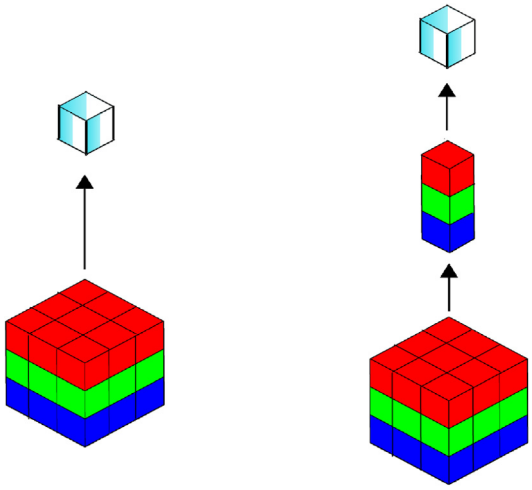

**Fig. 3.** 3 × 3 standard convolution and Depthwise Separable convolution (depthwise convolution followed by pointwise convolution).

The speed for equivalent 3 × 3 depthwise separable convolution is 9 times much faster than standard 3 × 3 convolution. So to extract the features faster we decided to use the mobilenet model and the detector technique as SSD. To use SSD, the final layers of the mobilenet need to be changed. The actual final layers of the mobilenet base network are 7 × 7 pixels followed by the globalpooling layer which transforms into 1 × 1 followed by the classification softmax layer. Instead, for SSD implementation, we not only took the output from the last layer of the base model but also from multiple previous layers and feed this to SSD layers. The mobilenet model converts the image pixels into features that are then used by the SSD to pinpoint the object looking at the features. So here the mobilenet is used as a feature extractor. The input to the SSD is provided from such output layers which will supply high-level features that would make a detector to accurately predict the region of interest. As object detection is much complicated than classification, the SSD detector adds more convolutional layers on top of the base mobilenet layers. The SSD predicts multiple classes present in the image with its confidence score.

## 3. Results and discussion

In recent years, the advancement of the machine learning techniques likes CNNs have contributed immensely in the direction of computer vision [11]. In this study, our aim was to develop the program through which we can do automated taxonomic identification of the benthic organism. In addition the software can measure other important parameters like counting, diversity, abundance, presence of young, adult and sex ratio which are important for ecologists to have proper scientific interpretation. Identification of fauna through machine learning is a very complex procedure since we have to provide all the features of images which make it easier for machine learning algorithms to distinguish the object. This is the first machine learning approach used for the automatic identification of the benthic fauna. Although, in other fields the automatic identification seems to be quite useful although they have used the camera trap images [12]. It is very common that taxonomic image sets use a standard view, where it covers most of the area and that there are few disturbing foreign elements in the images. However, in our program we have also trained the software with sediment particle data which always comes intact with organism images. If images are more heterogeneous, like our dataset one expects more training data to be needed for the same level of identification accuracy. Using roughly 150 images per category, we actually achieved better identification accuracies for the heterogeneous.

This was observed in our datasets as identification of higher groups was particularly challenging in the beginning since the model was getting confused between Nematoda and Nemertea. However, after providing quite number of images the model reported >92% which was quite impressive, especially given the small number of images per category in the training set. Therefore the model needs to achieve a good understanding of the diagnostic features of the higher taxonomic groups. The best model had an error rate of 0.1% while identifying

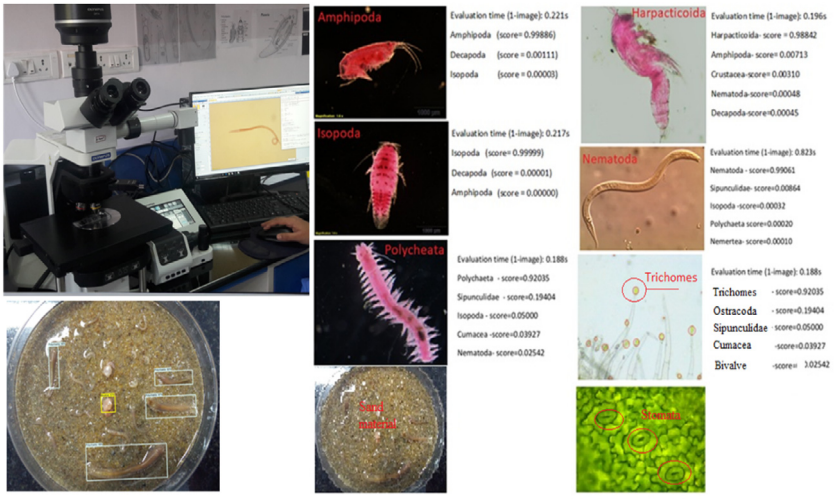


**Fig. 4.** The score level of each group and the execution process of the model.

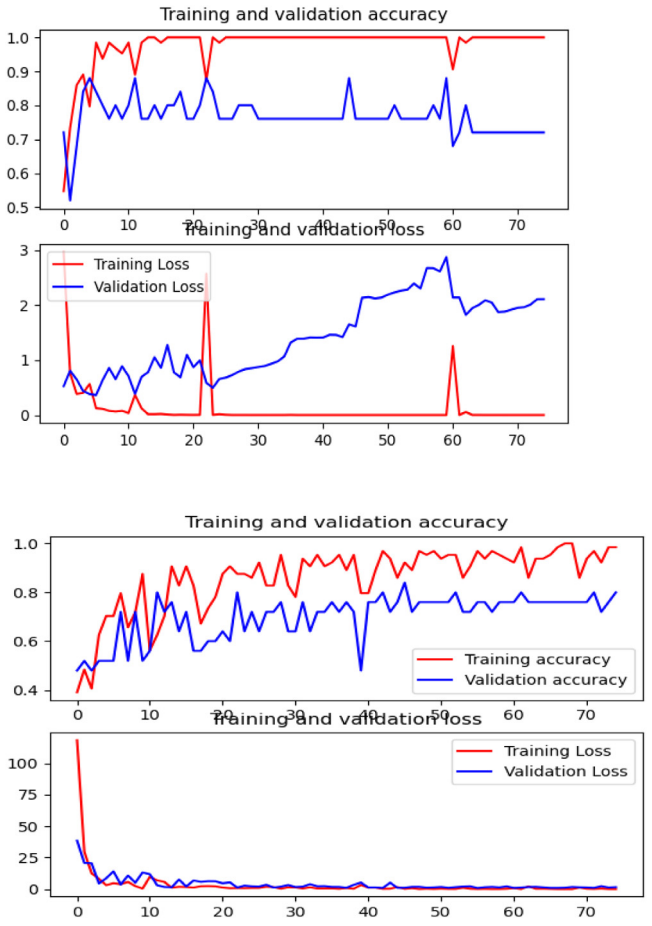


**Fig. 5.** The training loss curve before and after applying dropout and image augmentation.

the Harpacticoida, Isopoda Nematoda, and Amphipoda with 99% identification score (Fig. 4). A notable outlier is the family of Polycheata, which had unexpectedly low identification accuracy (85%) given the sample size (90 images). Another case that stands out is the Bivalve (80% accuracy) and small number of examples (55 images), which probably makes difficult to bring the accuracy.

The overfitting issue seen resolved after dropout was applied and image augmentation was used (Fig. 5)

It was observed that for the InceptionV3 model minimum of 32 samples per class is sufficient enough for having a higher accuracy for classifying between Orders (taxonomy). Although, we have used 150 images per class for robustness and achieves 99.9% accuracy on validation data (unseen data).

The images shown in Figs. 6 and 7 are very much different from training data with different orientations. The clipped image of Amphipoda is seen predicted correctly, as the major features are still visible in the image.

## 4. Conclusion

We proposed the first, to the best of our knowledge, approach that combines a powerful Neural Network architecture with tensor flow. This program can rapidly classify the millions

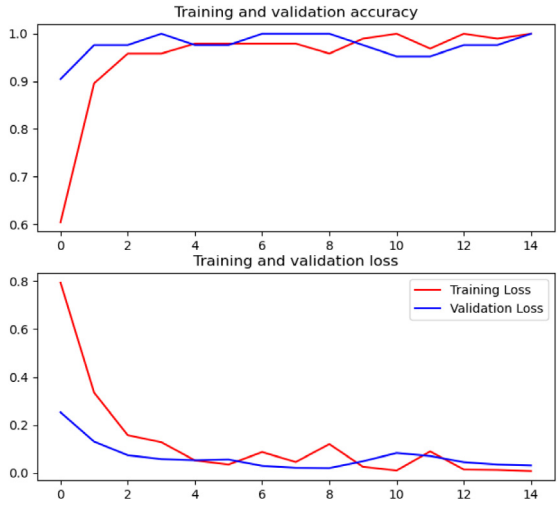


**Fig. 6.** Accuracy and loss plot for training the InceptionV3 model. The accuracy curve reaches a maximum with fewer epochs for the InceptionV3 network.

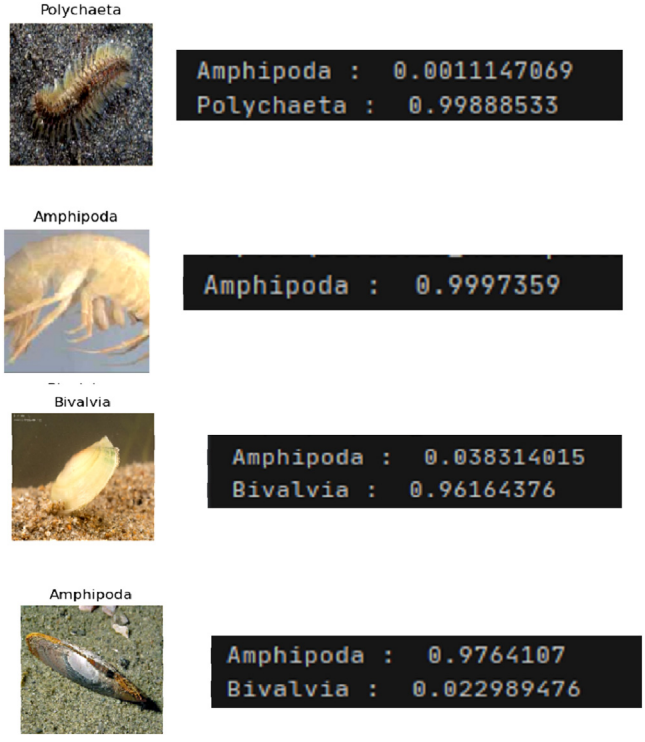


**Fig. 7.** Scores for some interesting unseen images downloaded from internet.

of benthic faunal images and it is a major breakthrough in the ecological and taxonomical field. This model can also be useful for the taxonomic identification of the other species. The advantage is that there is no need to provide morphological features manually for each set of objects, the machine itself will identify the unique feature set of each class. Hence, using this approach there will be less overhead while creating a classifier and one can easily increase the number of classes without major modification. Some of our results do still show entanglement in the variations. At present this software can work till group level but with increasing images size it can easily classify at genus/species level.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgment

The author would like to thank the director Dr. D. Srinivasa Reddy, Director, IIIM Jammu for providing the facility to carry out the work.

## References

[1] Paul Hebert, T. Ryan Gregory, The promise of DNA barcoding for taxonomy, Syst. Biol. 54 (5) (2005) 852–859, Web.

[2] A. Valentini, F. Pompanon, P. Taberlet, DNA barcoding for ecologists, Trends Ecol. Evol. 24 (2009) 110–117.

[3] H.R. Taylor, W.E. Harris, An emergent science on the brink of irrelevance: a review of the past 8 years of DNA barcoding, Mol. Ecol. Resour. 12 (2012) 377–388.

[4] C.M.G. Oliveira, R.A. Monteiro, V.C. Blok, Morphological and molecular diagnostics for plant-parasitic nematodes: working together to get the identification done, Trop. Plant Pathol. 36 (2011) 65–73.

[5] V. Savolainen, et al., Towards writing the encyclopaedia of life: An introduction to DNA barcoding, Philos. Trans. R. Soc. B 360 (1462) (2005) 1805–1811, Web.

[6] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444.

[7] J. Schmidhuber, Deep learning in neural networks: an overview, Neural Netw. 61 (2015) 85–117.

[8] M.A. Tabak, M.S. Norouzzadeh, D.W. Wolfson, S.J. Sweeney, K.C. Vercauteren, N.P. Snow, R.S. Miller, Machine learning to classify animal species in camera trap images: Applications in ecology, Methods Ecol. Evol. 10 (2019) 585–590.

[9] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, 2015, CoRR, abs/1512.00567.

[10] G. Andrew, Z. Howard Menglong, Bo Chen, K. Dmitry, W. Weijun, T. Weyand, M. Andreetto, H. Adam, MobileNets: Efficient convolutional neural networks for mobile vision applications, 2017, CoRR, abs/1704.04861.

[11] P. Mamoshina, A. Vieira, E. Putin, A. Zhavoronkov, Applications of deep learning in biomedicine, Mol. Pharmaceut. 13 (2016) 1445–1454.

[12] M. Valan, K. Makonyi, A. Maki, D. Vondráček, F. Ronquist, Automated taxonomic identification of insectswith expert-level accuracy using effective feature transfer from convolutional networks, Syst. Biol. 68 (2019) 876–895.