Combination of image and location information for snake species identification using object detection and EfficientNets

FHDO Biomedical Computer Science Group (BCSG)

Louise Bloch[1,2][0000−0001−7540−4980], Adrian Boketta[1][0000−0002−4182−2479], Christopher Keibel[1][0000−0003−4598−5504], Eric Mense[1][0000−0003−2748−7958], Alex Michailutschenko[1], Obioma Pelka[1,3][0000−0001−5156−4429], Johannes Rückert[1][0000−0002−5038−5899], Leon Willemeit[1], and Christoph M. Friedrich[1,2][0000−0001−7906−0038]

[1] Department of Computer Science, University of Applied Sciences and Arts Dortmund (FHDO), Emil-Figge-Str. 42, 44227 Dortmund, Germany
{louise.bloch, obioma.pelka, johannes.rueckert, christoph.friedrich}@fh-dortmund.de, {adrian.boketta001, keibel, eric.mense001, alex.michailutschenko004, leon.willemeit002}@stud.fh-dortmund.de

[2] Institute for Medical Informatics, Biometry and Epidemiology (IMIBE), University Hospital Essen, Essen, Germany

[3] Department of Diagnostic and Interventional Radiology and Neuroradiology, University Hospital Essen, Essen, Germany

Abstract. Snake species identification based on images is important to quickly treat patients suffering from snake bites using the correct antivenom. The SnakeCLEF 2020 challenge, which is part of the LifeCLEF research platform, is focused on this task and provides snake images and associated location information. This paper describes the participation of the FHDO Biomedical Computer Science Group (BCSG) in this challenge. The implemented machine learning workflow uses Mask Region-based Convolutional Neural Network (Mask R-CNN) for object detection, various image pre-processing steps, EfficientNets for classification as well as different methods to fuse image and location information. The best model submitted before the challenge deadline achieved a macro-averaging F_1-score of 0.404. After the expiration of this deadline, the results could be improved up to a macro-averaging F_1-score of 0.594.

Keywords: snake species identification · object detection · EfficientNets · image classification · metadata inclusion

1 Introduction

This paper explains the participation of University of Applied Sciences and Arts Dortmund (FHDO) Biomedical Computer Science Group (BCSG) at the Conference and Labs of the Evaluation Forum (CLEF) 2020[4] SnakeCLEF challenge[5] for snake species identification [20]. This challenge is part of the LifeCLEF 2020 research platform which focuses on the automated identification of species [14] and consists of four challenges. The implemented approach in this paper is inspired by an article [9] about the winning entry of round 2 of the AICrowd Snake Species Identification Challenge[6].

The identification of snake species is important as there are approximately between 81,410 and 137,880 victims of snakebites dying every year [29]. These deaths result from inaccurate knowledge about the species and consequently about the antivenom needed [5].

The high diversity of snake species [27] and their partially similar appearances lead to confusion [5] and make this choice more complicated. It is also mentioned, that an increasing amount of people who were bitten by a snake bring pictures of the snake, for example, taken with a smartphone, or the killed snake itself to the physician [5].

Therefore, the target of the SnakeCLEF challenge is the improved and robust identification of snake species based on photographs [20].

In this article, the experiments and results of FHDO BCSG are presented. For this reason, Section 2 describes previous work in this field of research. Afterwards, the general machine learning workflow is illustrated in Section 4, followed by a description of the achieved results in Section 5. Finally, the results are summarized in Section 6.

2 Related Work

Automated identification of snake species using machine learning is rarely studied, resulting from small datasets of annotated images.

James et al. [13] described a semiautomatic approach, where taxonomical features have been extracted from images to discriminate six different species. The dataset contained 1,299 images and the least frequent class included 88 images. Using different feature selection approaches, it has been concluded that the bottom-view taxonomical features are less important for the species identification than the front- and side-view features.

As manual extraction of features describing the appearance of a snake is tedious, recent articles used automated feature extraction, for example, texture features [4] or deep learning [2,3,9,18].

[4] `https://clef2020.clef-initiative.eu/`, [last accessed: 2020-07-17]
[5] `https://www.imageclef.org/SnakeCLEF2020`, [last accessed: 2020-07-17]
[6] `https://www.aicrowd.com/challenges/snake-species-identification-challenge`, [last accessed: 2020-07-17]

Texture features were used in Amir et al. [4] to distinguish between 22 different species. Their dataset contained 349 images and the least frequent snake species included three images. Using classical machine learning methods, the approach achieved a classification accuracy of 87 %.

Patel et al. [18] used deep learning methods to develop an application for smartphones which distinguishes images of nine different snake species, occurring on the Galápagos Islands in Ecuador. To this end, object detection, as well as classification algorithms, have been used. The training dataset for their implementation has been a bundle of three data sources, two internet searches of the platforms Google and Flickr were combined with an image dataset provided by the Ecuadorian institution Tropical Herping[7]. In total, 250 images were collected and the least frequent class contained seven images. Different model architectures have been tested for object detection and image classification. The model which was based on Faster Region-based Convolutional Neural Network (Faster R-CNN) [23] ResNet [11] achieved the best classification accuracy of 75 %. The authors state that a larger amount of training samples would be important for further investigations in this field.

Abdurrazaq et al. [2] used three different Convolutional Neural Network (CNN) architectures to distinguish five different snake species. They used a dataset containing 415 images. For the least frequent snake species, 72 images were available. The best results were achieved using a medium-sized classification network.

Abeysinghe et al. [3] used a deep Siamese network [6] to classify a relatively small dataset containing 200 images of 84 species based on World Health Organization (WHO) venomous snake database[8]. The approach described in their article concentrated on single-shot learning as the dataset included 3 to 16 images per species. The achieved results of the automated classification model performed worse than human classification accuracy. Pairwise classification results exceed class prediction accuracy.

As already mentioned, Gokula Krishnan [9] described the results of round 2 of the AICrowd Snake Species Identification Challenge. The solution which achieved the best results has used object detection as a pre-processing step to focus on the image parts containing the snake. On this basis, EfficientNets were applied afterwards for image classification. In round 2 the dataset included 187,720 images assigned to 85 classes.

3 Dataset

The training dataset used in the actual SnakeCLEF and AICrowd Snake Species Identification Challenge round 4 consists of 245,185 red-green-blue- (RGB-) color-space-images (models trained on the training dataset were referred to as T1) assigned to 783 different snake species. Additionally, a validation dataset is

[7] `https://www.tropicalherping.com/`, [last accessed: 2020-07-17]

[8] `https://apps.who.int/bloodproducts/snakeantivenoms/database/`, [last accessed: 2020-07-17]

available, which includes another 14,029 images (models trained on the training and validation dataset were referred to as T2). The class distribution of the snake species is highly unbalanced as can be seen in the bar plot of the absolute class frequencies depicted in Figure 1.

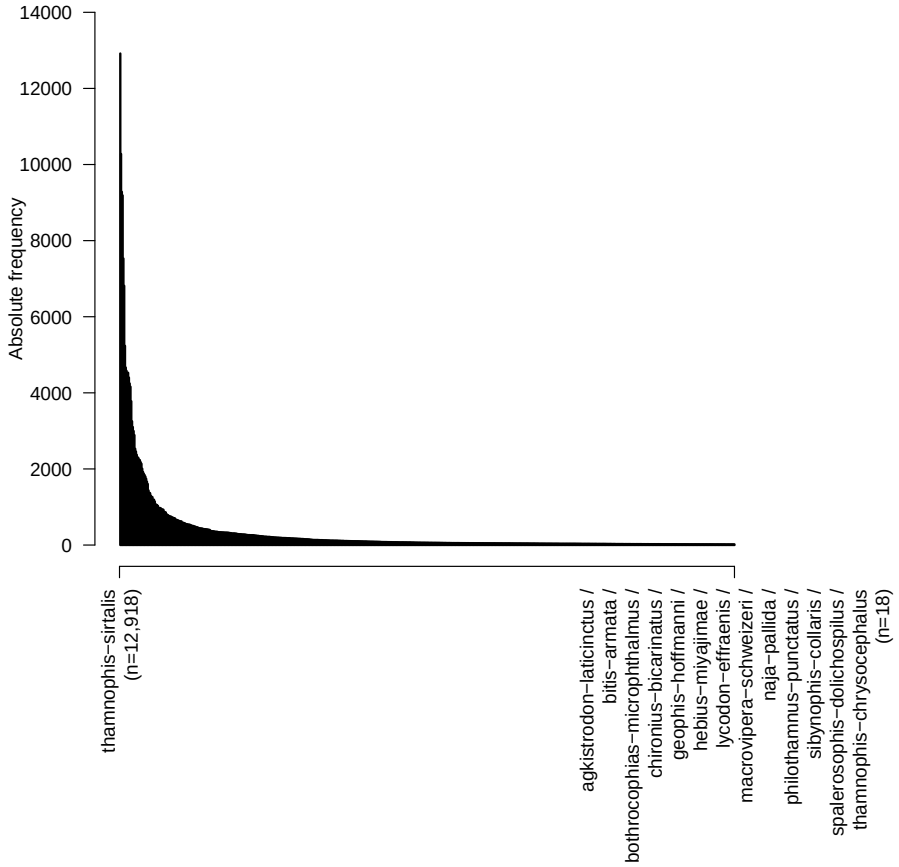

Fig. 1. Distribution of the snake species in training and validation dataset.

3.1 Image Dataset

An analysis of the dataset with AntiDupl[9] revealed 1,713 duplicate images in the training set. Some of these duplicates are associated with different species like "Image not found" images, that are the result of download problems. Other duplicates are correctly associated with several species as they depict distinct

[9] `https://github.com/ermig1979/AntiDupl`, [last accessed: 2020-07-15]

snakes. When the mean squared difference between images in AntiDupl is relaxed to 2 %, another 2,114 duplicates can be found. These are the result of different jpeg compression rates for the same image, resize operations or deletion of copyright information. Another problem that has been found are out-of-class images, that have been injected by the organizers. These images contain no snakes but for example, ice-hockey players, churches, other animals, persons, and mangas. To identify them for exclusion from the training set, a standard ImageNet [8] classifier with 1,000 classes and based on a ResNet50 [11] architecture has been used and a positive list of snake and reptile classes, that are part of the ImageNet Large Scale Visual Recognition Challenge 2012 (ILSRVC2012) [25] dataset has been used. With this classifier, about 4,000 out-of-class images have been identified and the effects of the reduced dataset (abbreviated as D1 hereafter) has been tested and compared to the unfiltered dataset. The results of this comparison are summarized in Table 7.

3.2 Metadata

The images are associated with metadata that provides information about the continent and country of the place where the image has been taken. For some snake depictions, the information is not given and only "UNKNOWN" is provided in the metadata. This information could be used for better classification. It should be noted, that the number of snake species in the dataset does not match the natural occurrence of a snake in a location. For example, the most frequent species with German country information in the dataset is *pantherophis guttatus*, the corn snake which is not endemic in Germany but is the pet snake number one in Germany. Accordingly, the data set takes into account that pet snakes can also attack humans.

4 Methods

This section describes the workflow used to learn a discrimination between the different snake species. The generalized workflow is depicted in Figure 2. The workflow is modular and in the course of the challenge, it was examined how different implementations of the individual modules affect the classification performance on the test dataset. In this section, the components are described more precisely and different implementations of them are demonstrated. The workflow has been implemented using the programming language Python 3.6.9 [28] and was based on Keras 2.2.4-tf [7] with a Tensorflow 2.1.0 [1] backend. For the inference on the AICrowd submission platform, Tensorflow 2.0.0 was used for reasons of compatibility.

Image pre-processing included an optional object detection stage and a mandatory stage, where rectangular images were transferred to a square shape afterwards. Additionally, the images were augmented, optionally branded using locational information and fed into the deep learning training network. Finally, an

optional multiplication of the prediction probabilities and the a priori probability distribution of the snake species occuring at the corresponding location has been implemented.

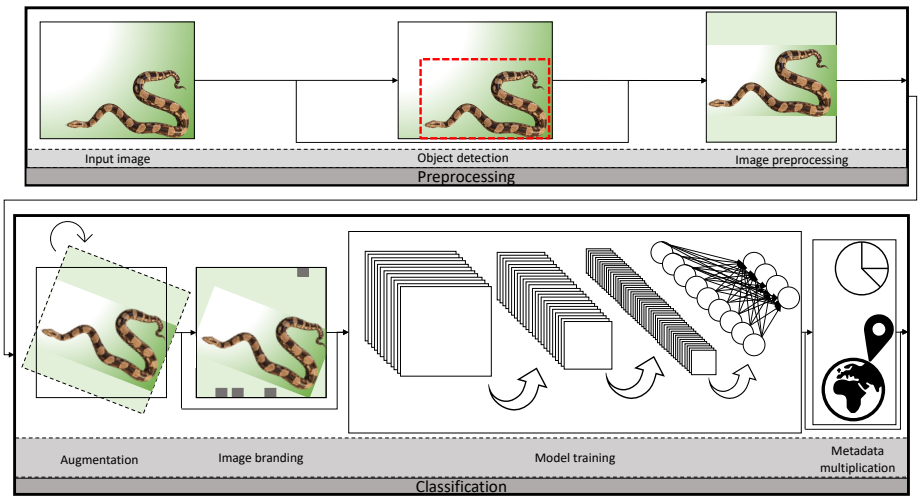

Fig. 2. Generalized workflow for snake species classification.

4.1 Object Detection

The idea of using an object detection stage before executing an image classification was inspired by the winning team [9] of round 2 of the AICrowd Snake Species Identification Challenge. Object detection has been implemented using the Mask R-CNN procedure, first described by He et al. [10]. Mask R-CNN performs instance segmentation as it extracts a bounding box, a class label and a pixel-wise segmentation mask for each object detected in an image. The Mask R-CNN algorithm is organized using two different stages. In the first stage, a backbone CNN extracts a feature map from the original image. In this paper, Resnet-50 has been used as a backbone. Afterwards but also in the first stage, a Region Proposal Network (RPN) is used to identify candidate object regions. So-called anchor boxes are used in this step to predefine bounding boxes. The second stage consists of a Region of Interest (ROI) align network which extracts multiple possible ROI sections. Based on these sections, a fully connected layer network is trained to perform a parallel softmax classification for class identification (snake vs. background in this case) and a regression task to specify bounding boxes. Additionally, a CNN-based mask classifier is employed for pixel-wise segmentation. In this article, the backbone model weights were initialized by the model weights trained on the ImageNet [8] dataset. The training on the snake

dataset has been implemented in two different phases. First all layers except the layers which are included in the backbone were trained using 20 epochs to warm up the model and afterwards 30 epochs were performed to train the entire model. The implementation of the Mask R-CNN used in this article is an adaption[10] of the implementation of Abdulla[11] transferred to use Tensorflow 2.1.0. No data augmentation has been used for object detection. The threshold of minimum detection confidence has been set to 0.3. Stochastic gradient descent (SDG) was used as an optimizer to train the model, momentum was set to 0.9. Further parameters include a weight decay, which was set to 0.0001 and a batch size of 8 was used.

In order to train the object detection model, the annotated snake images available from the winning solution of round 2 of the AICrowd Snake Species Identification Challenge [9] (O1 in Section 5) were used initially. Later, 400 additionally annotations were added to this dataset (O2 in Section 5) to investigate whether the object detection and thus the classification performance can be improved. The object detection results can be found in Table 3. Since Mask R-CNN is used in this approach only for object detection and not for instance segmentation, it may be an adequate solution to use Faster R-CNN instead of the Mask R-CNN. However, the results of the TensorFlow Object Detection application programming interface (API) [12], which represents a guide to choose an adequate object detection architecture shows an increased mean average precision (mAP) of 39.0 for the Microsoft Common Objects in Context (COCO) dataset [17] for Mask R-CNN object detection in comparison to Faster RCNN, which achieves a mAP of 38.7[12]. The use of the Mask R-CNN object detection makes it easier to supplement segmentation data prospectively, which was not used during this challenge due to a lack of time.

4.2 Image Pre-processing

As most of the deep learning classification models expect input images of square shape and predefined dimensions, it has been important to transform the mostly rectangular images, or image parts detected by the object detection into square shape and adjust the image for the expected image dimension of the classification model. There are different possibilities for the extraction of quadratic from rectangular images. The methods used in this paper are described in this section and the implemented combinations of this methods are summarized in Table 1. The results of the experiments achieved using different image pre-processing methods are summarized in Table 4.

Resize The least complex possibility has been to rescale images without consideration of aspect ratio. This resulted in highly distorted images so the texture and the shape of the snake have been disturbed especially for images with

[10] `https://github.com/DiffPro-ML/Mask_RCNN`, [last accessed: 2020-06-30]
[11] `https://github.com/matterport/MaskRCNN`, [last accessed: 2020-06-30]
[12] `https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md` [last accessed: 2020-08-03]

strongly different image dimensions. In this paper, two rescaling procedures, one considering and one retaining the aspect ratio were compared to each other. In the latter case, images had to be padded with further information to transfer them to a square shape.

Scaling Another problem which occurs during pre-processing is the problem of upscaling. Upscaling small images lead to poor image quality. It has been suspected that this could bring difficulties in texture recognition. In this paper, approaches which did and did not use upscaling for image pre-processing were compared to each other. If upscaling has been avoided, approaches were needed to pad pixel information for the remaining image sections.

Fill boundaries As previously mentioned, there were some different cases, where padding was required to get input images with preset image dimensions. One strategy to solve this issue has been to pad the image by a monochrome color. Koitka and Friedrich [16] recommended padding with a color matching to the image instead of using a predefined color (usually black or white). Since black is usually the most frequently occurring color in shady images, this approach used the average color of the original image or rather of the cropped areas as an alternative to pad the image.

In combination with object detection, it has been possible to increase the ROI and thus pad the image using background information instead of monochrome color. In this case, the image section predefined by the object detection workflow has been expanded as long as a quadratic section is found or one of the dimensions of the original image were smaller than the expected dimension of the square. If this happened, the average color of the image has been used to pad protruding boundaries. It has been attempted to include background evenly on all sides to center the snake. Sometimes this was not possible, for example, if the snake was located in a corner of the original image. In this case, the ROI has been moved to include background information of the remaining directions, thus the snake has not been centered in the image.

Table 1. Methods used for image pre-processing.

Abbreviation	Resizing	Scaling	Fill boundaries
I1	No consideration of the aspect ratio	Up-scaling	No padding
I2	Consideration of the aspect ratio	Up-scaling	Monochrome padding
I3	Consideration of the aspect ratio	No up-scaling	Monochrome padding
I4	Consideration of the aspect ratio	No up-scaling	Background padding

4.3 Data Augmentation

Data augmentation has been used to expand the training images and avoid overfitting. In each epoch of the training process, the images were randomly transformed. These transformations included random cropping of approximately 10 % of the image pixels per dimension, a rotation in the range of ±40 °, a width-shift, height-shift, random shearing, zooming each with a factor of 0.2, as well as the possibility of horizontal flipping. If pixel positions were generated during this procedure, for which no image information has been available, those were filled using the value of the nearest available image position. During the challenge, the workflow has been adapted to speed up the image classification procedure. In the later version of the workflow those pixels used black as a monochrome color.

4.4 Image Classification

EfficientNets As also used by Gokula Krishnan [9], EfficientNets, first described in Tan and Le [26], were used for classification in this approach. The baseline EfficientNet-B0 architecture is generated using an architecture search that parallely optimizes accuracy on a predefined classification task and Floating Point Operations Per Second (FLOPS) [26]. Based on this baseline model, larger models of the same family are created by scaling the depth, height and resolution of the baseline model uniformly. The different models of this family achieve state-of-the-art classification accuracy on ImageNet [8]. Additionally, the architecture is smaller and faster on inference compared to other existing CNNs [26]. EfficientNets were successfully adapted to different machine learning problems using transfer learning [26].

Various models of the EfficientNets family were used in this competition from EfficientNet-B0 up to EfficientNet-B4 networks (B0 - B4 in Section 5). The results of using different models of the EfficientNets family can be found in Table 6. The model weights were initialized by a model pre-trained using noisy student [30]. The EfficientNets were extended by a flatten layer, a dense layer with 1000 neurons and Swish [22] as an activation function and a dense layer with 783 neurons, which corresponds to the number of snake species and softmax activation were added to the previous architecture. The described model was trained for a few epochs on the snake classification task to warm-up the network. In this phase only the newly added layers and the batch normalization layers have been trained. Afterwards all layers were trained for a larger number of epochs (N10+50 denotes a warm-up phase including ten epochs and 50 epochs are used to train the entire model). Different batch sizes were used as further parameters to train the model (32 is encoded as BS32, 64 as BS64 etc., BS64/32 means that a batch size of 64 has been used during the warm-up phase and a batch size of 32 has been used afterwards). The chosen batch size depended on the image size (e.g., an image size of 128×128 is encoded as S128 in Section 5) the classification model and the available graphics processing unit (GPU) memory. The results of models using different image sizes can be found in Table 5. The learning rate (α) was likewise adjusted depending on the batch size (LR1 encodes

a learning rate of 10^{-4} during the warm-up phase and 10^{-5} during fine-tuning and LR2 encodes a learning rate of 10^{-5} during the warm-up phase and 10^{-6} during fine-tuning in Section 5). All submissions described in this paper used the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-7}$) [15] to minimize categorical cross entropy. The implementation of the classification model workflow used an EfficientNets 1.1.0 implementation of Tensorflow Keras 2.2.4 [7].

Since the dataset of the challenge had very unbalanced class frequencies, different class weight functions were used in order to implement an oversampling. Equation 1 describes a linear class weight function (W1 in Section 5) and Equation 2 describes a function where very low frequencies were less oversampled (W2 in Section 5). For both equations, $F(c)$ denotes the frequency of class c. For comparability reasons, one model has been trained without class weights.

$$w_1(c) = \frac{\max F(c)}{F(c)} \tag{1}$$

$$w_2(c) = 1 - \frac{1}{\sqrt{\frac{\max F(c)}{F(c)} + 0.5}} \tag{2}$$

Polyak Averaging Polyak averaging, based on the approach of Polyak [21] and Ruppert [24], is a method to combine the learned weights of different epochs during the model training in order to obtain a final model with more robust weights. In this paper, it has been tested if Polyak averaging leads to improved classification results (P1 denotes the described Polyak averaging in Section 5). Therefore the learned weights of the last five epochs were averaged using an exponential function described in Equation 3, where i has a value of 1 for the last epoch, 2 for the penultimate epoch and 5 for the fifth last epoch.

$$W_{polyak}(i) = \exp\Big(\frac{-i}{2}\Big) \tag{3}$$

4.5 Addition Of Location Information

Optionally, location information was added to some models by multiplying the prediction probabilities of the classification model by the a priori probability of the snake class for the specified location (M1 denotes the multiplication of the locational distribution). The a priori probabilities were estimated by the relative frequency distribution of the snake species at the location in the training and validation dataset. Usually, the country information was used in this step, only if this information was missing, the distribution of the continent has been used instead. For some images, both country and continent information were missing. In those cases, the frequency distribution of the entire dataset has been used. The softmax function was applied after this multiplication, to normalize the results.

Another variant has been implemented based on the previously described procedure. The sole exception has been that the raw prediction probabilities

of images with missing country and continent information were not multiplied (abbreviated as M2 in Section 5). As a second variation of this method, all prediction probabilities were multiplied by a binary variant of the frequency distribution, which thus denotes if a snake was or was not present at a location (M3 in Section 5). The results achieved using the different metadata integration strategies are summarizes in Table 8.

During the experiments of the FHDO BCSG a few alternatives have been investigated. These methods were only tested in small experiments and are not described in this paper for reasons of clarity.

Image Branding As an alternative to the simple multiplication of the location distributions, an approach has been implemented, which directly adds the location information into the classification network. This has been done using a binary image branding technique introduced in Pelka et al. [19], which adds grey (RGB = [102,102,102]) boxes encoding the location information directly to the images. The height of the boxes was set to 8 pixels while the width (b_w) depends on the image dimensions d and is described in Equation 4.

$$b_w = \left\lfloor \frac{d}{8} \right\rfloor - 4 \tag{4}$$

The first box starts directly at the left border of the image and after every box, space was left for 4 pixels.

The continent information has been added as binary boxes at the top border of the image, while the country information has been added at the bottom border of the image. Since a distinction has been made between seven continents as well as the "unknown"-class, every box at the top of the image represents a continent (abbreviated as M4). A similar approach to encode the country information would result in small boxes because 189 countries had to be distinguished. The used image branding approach is illustrated in Figure 3. In this case, a binary encoding of the country index has been chosen, so that eight boxes could be used to represent $2^8 = 256$ different countries. Hereafter, the combined branding of continent and country information is abbreviated as M5.

5 Results

In this section, the classification results for the test dataset of the challenge are described. Table 2 summarizes the most relevant successful submissions of the FHDO BCSG for the SnakeCLEF challenge.

This table is mainly used to give an overview about the submitted models. In order to get a better insight into the partial results and the effects of the different methods used, partial aspects are considered in individual tables in the further course of this section. It was possible to submit models to the AICrowd Snake Species Identification Challenge after the submission deadline of SnakeCLEF expired. Therefore Table 2 presents a few models which achieve better

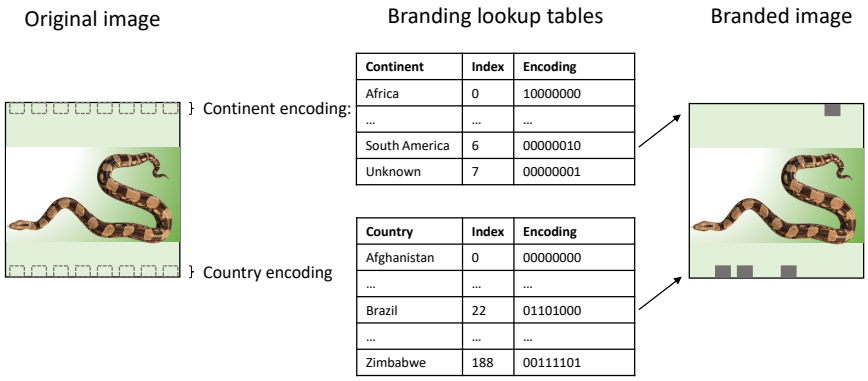

Fig. 3. Appropriated branding approach for country and continent branding.

results than the best submission in the SnakeCLEF challenge. In order to avoid miscommunication, the submissions in Table 2 are listed in chronological order and the deadline of the challenge is highlighted.

The results of the different object detection datasets are summarized in Table 3. This table only presents the parameters, which are necessary for this comparison. It should be noted, that all the other parameters of the compared models are identical, as can be verified in Table 2. This type of presentation is also used in subsequent tables.

The comparison of submissions 68418 and 68450, as reflected in Table 3 shows, that the macro-averaging F_1-score (abbreviated as F_1 hereafter) increased by 0.010 when additional images were annotated, whereas log loss remains stable. Moreover, the number of images where no snakes were identified decreases from 141 to 123 in the joined training and validation dataset.

Additionally, submission 68678 is a model, which was trained using no object detection. This model is not completely comparable to any other models, but submission 68632 differs only in the image pre-processing step. Comparing those two models, shows a slightly better performance of the model which used the object detection. As previously mentioned, this comparison is not entirely fair.

Table 4 summarizes the results of models trained based on different pre-processing methods. As can be seen, pre-processing influenced the classification results achieved for the test dataset. Remarkable was the good performance of the submission 68506 which used image resizing without consideration of the aspect ratio. This model achieved the best F_1 of 0.452. It has been expected, that this image pre-processing would achieve bad results as major distortions were possible, so in some cases humans were not able to recognize snakes in those images.

The second-best result was achieved for submission 67962. In this submission, the ROIs detected during object detection have been expanded and thus were padded using background information. The submission reached an F_1 of 0.403

Table 2. Classification results achieved for the official test dataset, including macro-averaging F_1-score (F_1) and log loss. The best results in each section are highlighted in bold.

ID	Object detection	Image preprocessing		Classification model training						Dataset		Metadata	F_1	Log loss
67675	O1	I2	S128	B0	BS64	W2	LR1	N10+50	-	D1	T2	M1	0.338	6.652
67696	O1	I2	S128	B2	BS64	W2	LR1	N10+50	-	D1	T2	M1	0.392	**6.630**
67700	O1	I2	S128	B0	BS64	W2	LR1	N10+50	-	-	T2	M1	0.352	6.651
67727	O1	I2	S128	B2	BS64	W2	LR1	N10+50	-	-	T2	M1	0.389	6.650
67734	O1	I2	S128	B4	BS64	W2	LR1	N10+50	-	D1	T1	M1	0.403	6.650
67882	O1	I2	S128	B2	BS64	W1	LR1	N10+50	-	-	T2	M1	0.365	6.657
67901	O1	I2	S128	B2	BS64	-	LR1	N10+50	-	-	T2	M1	0.377	6.647
67962	O1	I4	S128	B2	BS64	W2	LR1	N10+50	-	-	T2	M1	0.403	6.650
68023	O1	I2	S128	B4	BS64	W2	LR1	N10+50	P1	D1	T1	M1	**0.404**	6.650
submission deadline														
68418	O1	I4	S196	B2	BS64	W2	LR1	N10+50	-	-	T2	M1	0.475	6.645
68432	O1	I3	S128	B2	BS64	W2	LR1	N10+50	-	-	T2	M1	0.369	6.650
68450	O2	I4	S196	B2	BS64	W2	LR1	N10+50	-	-	T2	M1	0.485	6.645
68506	O1	I1	S128	B2	BS64	W2	LR1	N10+50	-	-	T2	M1	0.452	6.648
68520	O1	I2	S224	B0	BS64	W2	LR1	N10+50	-	-	T1	-	0.322	1.877
68541	O1	I2	S128	B4	BS64	W2	LR1	N10+50	-	-	T2	M1	0.426	6.648
68574	O1	I2	S224	B0	BS64	W2	LR1	N10+50	-	-	T1	M1	0.431	1.659
68575	O1	I2	S224	B0	BS64	W2	LR1	N10+50	-	-	T1	M2	0.447	1.583
68593	O1	I2	S196	B4	BS64	W2	LR1	N10+50	-	D1	T1	M1	0.483	6.645
68632	O1	I3	S196	B4	BS64/32	W2	LR1	N10+50	-	-	T2	M1	0.366	6.646
68655	O1	I2	S224	B0	BS64	W2	LR1	N10+50	-	-	T1	M3	0.445	1.596
68678	-	I1	S196	B4	BS64/32	W2	LR1	N10+50	-	-	T2	M1	0.347	6.647
69365	O2	I4	S380	B4	BS13	W2	LR2	N10+50	-	-	T1	M1	0.460	1.379
69750	O2	I4	S380	B4	BS13	W2	LR2	N10+50	-	-	T1	M5	0.361	1.541
69768	O2	I4	S380	B4	BS13	W2	LR2	N10+50	-	-	T1	M4	0.437	1.363
69849	O2	I4	S380	B4	BS13	W2	LR2	N10+50	-	-	T1	M4+M1	0.459	1.355
69888	O1	I3	S380	B4	BS13	W2	LR2	N10+109	-	-	T1	M2	**0.594**	**1.064**

Abbreviations: O1: Object detection dataset from [9], **O2:** Expanded dataset, **I1:** No aspect ratio, up-scaling, no padding, **I2:** Aspect ratio, up-scaling, monochrome padding, **I3:** Aspect ratio, no up-scaling, monochrome padding, **I4:** Aspect ratio, no up-scaling, background padding, **S**x**:** Image size: $x \times x$ pixels, **B**x**:** EfficientNet-Bx, **BS**x**:** Batch size of x for image classification, **BS**x/y**:** Batch size: warm-up-phase: x, fine-tuning: y, **W1:** Linear weights, **W2:** Nonlinear weights, **LR1:** Learning rate warm-up phase: 10^{-4}, fine-tuning: 10^{-5}, **LR2:** Warm-up phase: 10^{-5}, fine-tuning: 10^{-6}, **N**x**+**y**:** Training epochs warm-up phase: x, fine-tuning: y, **P1:** Polyak averaging, **D1:** Reduced dataset, **T1:** Training dataset, **T2:** Training + test dataset, **M1:** Multiplication of metadata, **M2:** Multiplication without unknown cases, **M3:** Binary multiplication, **M4:** Continent branding, **M5:** Continent and country branding

Table 3. Official classification results on the test dataset to compare object detection datasets. The results include F_1 and log loss. The best results in each section are highlighted in bold. Presented results represent ablation studies, thus non-mentioned parameters are fixed in each section.

ID	Object detection	Image pre-processing	F_1	Log loss
68418	Dataset from [9] (O1)	Aspect ratio, no up-scaling, background padding (I4)	0.475	**6.645**
68450	Expanded dataset (O2)	Aspect ratio, no up-scaling, background padding (I4)	**0.485**	**6.645**
68632	Dataset from [9] (O1)	Aspect ratio, no up-scaling, monochrome padding (I3)	**0.366**	**6.646**
68678	No object detection (-)	No aspect ratio, up-scaling, without padding (I1)	0.347	6.647

and thus outperformed the F_1 of submission 68432, which used a monochrome color padding strategy, by 0.034. The comparison between submission 68432 and submission 67727 shows a slightly positive effect of using upscaling, as the F_1 of submission 67727 is 0.020 higher than the F_1 of submission 68432. The previously described comparison is based on small images containing 128×128 pixels, for future investigations, it would be interesting how the pre-processing methods affect larger images.

Table 4. Official classification results on the test dataset to compare pre-processing methods. The results include F_1 and log loss. The best results are highlighted in bold. Presented results represent ablation studies, thus non-mentioned parameters are fixed.

ID	Pre-processing pipeline	F_1	Log loss
68506	No aspect ratio, up-scaling, without padding (I1)	**0.452**	**6.648**
67727	Aspect ratio, up-scaling, monochrome padding (I2)	0.389	6.650
68432	Aspect ratio, no up-scaling, monochrome padding (I3)	0.369	6.650
67962	Aspect ratio, no up-scaling, background padding (I4)	0.403	6.650

Table 5 summarizes the official classification results achieved using different image sizes as model input. The results of the comparison corresponds to other experiments executed during the challenge and shows that models trained on larger image input sizes achieved better classification results. Increasing the image size from 128×128 to 196×196 boosted the F_1 by approximately 0.080. The used image sizes may look striking, because EfficientNet-B0 models are usually trained using images including 224×224 pixels and EfficientNet-B4 models are optimized for an image size of 380×380 pixels. The use of small images in this approach resulted from the fact that some early submissions failed because of

memory issues. The problem has been fixed after the deadline of the SnakeCLEF challenge expired. Some of the later submissions used larger image sizes consistent to the original EfficientNets input sizes and thus achieved better results.

Table 5. Official classification results on the test dataset to compare image input sizes. Both models use the EfficientNet-B4 architecture. The results include F_1 and log loss. The best results are highlighted in bold. Presented results represent ablation studies, thus non-mentioned parameters are fixed.

ID	Image size	F_1	Log loss
67734	128×128 (S128)	0.403	6.650
68593	196×196 (S196)	**0.483**	**6.645**

Next, the influence of different model architectures on the classification results were investigated. In Table 6, a comparison is presented concerning different model architectures. The comparison shows, concurrently to some experiments not listed here for reasons of clarity, increased F_1 for upscaled models. Submission 67727, which was based on an EfficientNet-B2 architecture outperformed submission 67700 by an increase of the macro averaging F_1-score of 0.037. Submission 68541, which represents an EfficientNet-B4 architecture, achieved an F_1 of 0.426 and thus outperformed the results of submissions 67727 and 67700 by 0.037 and 0.074. It should be noted that all of the submissions summarized in Table 6 were trained using an image size of 128×128 pixels which is due to some memory issues already mentioned before.

Some additional experiments were performed comparing different top layer architectures, for lack of time those were not completely comparable to each other, especially because the number of epochs used for training differed for most of the models. For this reason these results are not elaborated in this paper.

Table 6. Official classification results on the test dataset to compare different models of the EfficientNets family. The results include F_1 and log loss. All models used input images containing 128×128 pixels. The best results are highlighted in bold. Presented results represent ablation studies, thus non-mentioned parameters are fixed.

ID	Model architecture	F_1	Log loss
67700	EfficientNet-B0 (B0)	0.352	6.651
67727	EfficientNet-B2 (B2)	0.389	6.650
68541	EfficientNet-B4 (B4)	**0.426**	**6.648**

It has been mentioned in Section 4 that different weight functions can be used to overcome unbalanced class distributions. The results of submissions 67727, 67882 and 67901 show, that the function introduced in Equation 2, which has been used in submission 67727 achieved a macro averaging F_1-score of 0.389 and thus outperformed submission 67882, which used a linear class weight function and achieved an F_1 of 0.365 and submission 67901, which used no class weights and achieved a macro averaging F_1-score of 0.377. One possible reason for the poor results of the linear weighting could be the high differences in class frequencies, which lead to larger weights for rare classes.

The results of the dataset filtering strategies, which are presented in Table 7, have been inconclusive. For the workflow used in submissions 67675 and 67700, the model trained on the reduced dataset performed worse than the model trained on the complete dataset. The opposite behaviour has been observed for the workflow used in submissions 67696 and 67727, which achieved F_1 of 0.392 and 0.389. As the filtering removed images from the training dataset, where no snakes were present and no clear benefit is reached using this filtering, one could assume, that there might be some images in the test dataset where no snakes are present.

Table 7. Official classification results on the test dataset to compare different dataset filtering strategies. The results include F_1 and log loss. The best results in each section are highlighted in bold. Presented results represent ablation studies, thus non-mentioned parameters are fixed in each section.

ID	Strategy for dataset filtering	F_1	Log loss
67675	Duplicates and plausibility filtering (D1)	0.338	6.652
67700	No filtering (-)	**0.352**	**6.651**
67696	Duplicates and plausibility filtering (D1)	**0.392**	**6.630**
67727	No filtering (-)	0.389	6.650

As can be noted in Table 2, most of the earlier submissions, achieved high log losses of about 6.6, while others had log losses of about 1 with a confusing dependency to the achieved macro averaging F_1-scores. This problem appeared because of softened prediction results if softmax normalization is used after the multiplication of the location frequencies and has been fixed using maximum-normalization instead (e.g., submissions 68520, 68574, 68575 and 68655).

The results of adding metadata to improve image classification are described in Table 8. It can be seen, that adding metadata to a model by multiplying the model results using the a priori probability of the snake class for the given location lead to an increased F_1. Submission 68574, which was trained using the same workflow as submission 68520, except adding metadata, outperforms this model by an increase of F_1 of 0.109. As can be seen looking at submission 68575 those results can be further improved by a value of 0.016, if the multiplication

is only used for available country and continent information. Submission 68655 exhibited a similar result of 0.445 using binary information about the availability of a species in a country or continent.

The results of the submissions 69365, 69750, 69768 and 69849 show that models which used the image branding presented in Section 4, achieved no benefit in comparison to multiplying the raw predictions by the location information. It can be noted, that submission 69750, which combined country and continent branding achieved a poor F_1 of 0.361, whereas the model, which used only continent branding (submission 69768), achieved a better F_1 of 0.437. One possible reason for this might be the use of the complex positional encoding of the country information which is hard to learn for a CNN, which focuses more on local differences. Because of the limited time, it was not possible to investigate this problem more thoroughly, thus additional investigations should be under examination in future work. The combination of continent branding and multiplication of the a priori probability distribution in submission 69849 achieved similar results than the model, which used no branding, but the multiplication.

Table 8. Official classification results on the test dataset to compare strategies to fuse image and location data. The results include F_1 and log loss. The best results in each section are highlighted in bold. Presented results represent ablation studies, thus non-mentioned parameters are fixed in each section.

ID	Strategy to integrate metadata	F_1	Log loss
68520	No metadata (-)	0.322	1.877
68574	Multiplication (M1)	0.431	1.659
68575	Multiplication without unknown (M2)	**0.447**	**1.583**
68655	Binary multiplication (M3)	0.445	1.596
69365	Multiplication (M1)	**0.460**	1.379
69750	Branding continent and country (M5)	0.361	1.541
69768	Branding continent (M4)	0.437	1.363
69849	Branding continent and multiplication (M4+M1)	0.459	**1.355**

The best model submitted before the SnakeCLEF deadline expired was submission 68023 which was based on an EfficientNet-B4 model architecture and achieved an F_1 of 0.404. Due to the previously mentioned memory issues, this model used an unusual small image size of 128×128 pixels. The newly added layers of the described model were trained with a warm-up phase of ten epochs and another 50 epochs were used to fine-tune the entire model. During the training process a batch size of 64, a learning rate of 0.0001 and Adam optimizer have been used. Location information was added using the described multiplication procedure. Polyak averaging with exponential weights has been used to combine the results of the last five training epochs. The Polyak averaging achieved an

improvement of F_1 of 0.001 in comparison to submission 67734 which used no Polyak averaging.

The best submission after the SnakeCLEF deadline expired was submission 69888 which achieved a macro-averaging F_1-score of 0.594 and a log loss of 1.064. The main differences in comparison to the best model before the deadline expired were, that the model was trained using the predefined image dimensions of an EfficientNet-B4, which are 380×380 pixels. Due to the increased image size, a smaller batch size of 13 and a decreased learning rate has been used. Furthermore, 109 instead of 50 training epochs have been applied, and the location distribution was multiplied only for known countries and continents. The model included no Polyak averaging.

6 Conclusions

In conclusion, it can be stated that snake species identification is a challenging task, primarily because of the high diversity of snake species, high intra-class variance, and low inter-class variance.

The main improvements in snake species classification presented in this paper are based on increasing image size, combining location and image information as well as upscaled model architecture. The results presented in this article show improved classification results using an object detection strategy previously to the image classification. However, a plausibility filtering of the training dataset showed no clear improvement. Some differences were detected in dependence of the pre-processing steps. Nevertheless, no clear insights could be achieved about which steps are particularly promising for good classification results. The implementation and application of the different pre-processing steps turned out to be relatively time-consuming. Besides, it has been previously mentioned, that there were some memory issues which lead to a focus on small image sizes as well as less upscaled model architectures in the early course of the challenge. Thus the time needed to optimize the classification parameters more precisely and to try out different optimizers was reduced. It is expected, that the results may be further improved by adjusting those parameters.

Due to the use of AICrowd as a submission platform, it has been possible to test a large number of different models. This enables to get direct feedback about the performance on the test dataset, and thus gives a good estimate about which methods gets the most promising results. In addition it facilitates the comparison between teams before the deadline expires. In this article, it has been mentioned before, that there were some memory issues which were related to the architecture of the test environment. In some cases debugging has been complicated because the logs were not accessible. These concerns were compensated by the very prompt and useful help from the organizers.

7 Acknowledgment

The work of Louise Bloch and Obioma Pelka was partially funded by a PhD grant from University of Applied Sciences and Arts Dortmund, Germany.

The authors want to thank Raphael Brüngel for the constructive proofreading of the manuscript.

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X.: Tensorflow: A system for large-scale machine learning. In: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16). pp. 265–283 (2016), https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf
2. Abdurrazaq, I.S., Suyanto, S., Utama, D.Q.: Image-Based Classification of Snake Species Using Convolutional Neural Network. In: 2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI). pp. 97–102. Institute of Electrical and Electronics Engineers (IEEE) (2019). https://doi.org/10.1109/isriti48646.2019.9034633
3. Abeysinghe, C., Welivita, A., Perera, I.: Snake Image Classification Using Siamese Networks. In: Proceedings of the 2019 3rd International Conference on Graphics and Signal Processing (ICGSP '19). pp. 8–12. Association for Computing Machinery (ACM), New York, NY, USA (2019). https://doi.org/10.1145/3338472.3338476
4. Amir, A., Zahri, N.A.H., Yaakob, N., Ahmad, R.B.: Image Classification for Snake Species Using Machine Learning Techniques. In: Phon-Amnuaisuk, S., Au, T.W., Omar, S. (eds.) Computational Intelligence in Information Systems. pp. 52–59. Springer International Publishing, Cham (2017). https://doi.org/10.1007%2F978-3-319-48517-1_5
5. Bolon, I., Durso, A.M., Botero Mesa, S., Ray, N., Alcoba, G., Chappuis, F., Ruiz de Castañeda, R.: Identifying the snake: First scoping review on practices of communities and healthcare providers confronted with snakebite across the world. PLOS ONE **15**(3), e0229989 (03 2020). https://doi.org/10.1371/journal.pone.0229989
6. Bromley, J., Bentz, J.W., Bottou, L., Guyon, I., Lecun, Y., Moore, C., Säckinger, E., Shah, R.: Signature Verification using a "Siamese" Time Delay Neural Network. International Journal of Pattern Recognition and Artificial Intelligence **07**(04), 669–688 (08 1993). https://doi.org/10.1142/s0218001493000339
7. Chollet, F.: Keras (2015), https://keras.io,[last accessed: 2020-07-14]
8. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255. Institute of Electrical and Electronics Engineers (IEEE) (2009). https://doi.org/10.1109/cvpr.2009.5206848
9. Gokula Krishnan: Diving into Deep Learning — Part 3 — A Deep learning practitioner's attempt to build state of the art snake-species image classifier (2019), https://medium.com/@Stormblessed/diving-into-deep-learning-part-3-a-deep-learning-practitioners-attempt-to-build-state-of-the-2460292bcfb, [last accessed: 2020-06-10]

10. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 2980–2988. Institute of Electrical and Electronics Engineers (IEEE) (2017). https://doi.org/10.1109/iccv.2017.322
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778. Institute of Electrical and Electronics Engineers (IEEE) (2016). https://doi.org/10.1109/cvpr.2016.90
12. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., Murphy, K.: Speed/accuracy trade-offs for modern convolutional object detectors. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3296–3297 (2017)
13. James, A., Kumar, D., Mathews, B., Sugathan, S.: Discriminative histogram taxonomy features for snake species identification. Human-centric Computing and Information Sciences **4**(1) (02 2014). https://doi.org/10.1186/s13673-014-0003-0
14. Joly, A., Goëau, H., Kahl, S., Deneu, B., Servajean, M., Cole, E., Picek, L., Ruiz De Castañeda, R., Bolon, I., Lorieul, T., Botella, C., Glotin, H., Champ, J., Vellinga, W.P., Stöter, F.R., Dorso, A., Bonnet, P., Eggel, I., Müller, H.: Overview of LifeCLEF 2020: a System-oriented Evaluation of Automated Species Identification and Species Distribution Prediction. In: Proceedings of CLEF 2020, CLEF: Conference and Labs of the Evaluation Forum, Sep. 2020, Thessaloniki, Greece. (2020)
15. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: 3rd International Conference for Learning Representations (2014), `https://arxiv.org/abs/1412.6980`
16. Koitka, S., Friedrich, C.M.: Optimized Convolutional Neural Network Ensembles for Medical Subfigure Classification. In: Jones, G.J., Lawless, S., Gonzalo, J., Kelly, L., Goeuriot, L., Mandl, T., Cappellato, L., Ferro, N. (eds.) Experimental IR Meets Multilinguality, Multimodality, and Interaction: Proceedings of the 8th International Conference of the CLEF Association, CLEF 2017. pp. 57–68. Springer International Publishing, Cham (09 2017). https://doi.org/10.1007/978-3-319-65813-1_5
17. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. pp. 740–755. Springer International Publishing, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
18. Patel, A., Cheung, L., Khatod, N., Matijosaitiene, I., Arteaga, A., Gilkey, J.W.: Revealing the Unknown: Real-Time Recognition of Galápagos Snake Species Using Deep Learning. Animals **10**(5), 806 (2020). https://doi.org/10.3390/ani10050806
19. Pelka, O., Nensa, F., Friedrich, C.M.: Variations on Branding with Text Occurrence for Optimized Body Parts Classification. In: 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). pp. 890–894. Institute of Electrical and Electronics Engineers (IEEE) (2019). https://doi.org/10.1109/EMBC.2019.8857478
20. Picek, L., Ruiz De Castañeda, R., Durso, A.M., Bolon, I., Sharada, P.M.: Overview of the SnakeCLEF 2020: Automatic Snake Species Identification Challenge. In: CLEF task overview 2020, CLEF: Conference and Labs of the Evaluation Forum, Sep. 2020, Thessaloniki, Greece. (2020)
21. Polyak, B.: New method of stochastic approximation type. Automatic Remote Control **51**, 937—-946 (1990)

22. Ramachandran, P., Zoph, B., Le, Q.V.: Searching for activation functions. Computing Research Repository (CoRR) **abs/1710.05941** (2017), http://arxiv.org/abs/1710.05941
23. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(6), 1137–1149 (2017). https://doi.org/10.1109/tpami.2016.2577031
24. Ruppert, D.: Efficient Estimations from a Slowly Convergent Robbins-Monro Process. Tech. rep., School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY (02 1988)
25. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision **115**, 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
26. Tan, M., Le, Q.: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. vol. 97, pp. 6105–6114. Long Beach, California, USA (06 2019), http://proceedings.mlr.press/v97/tan19a.html
27. Uetz, P., Hallermann, J., Hosek, J.: The Reptile Database 2019, http://reptile-database.reptarium.cz, [last accessed: 2020-06-10]
28. Van Rossum, G., Drake, F.L.: Python 3 Reference Manual. CreateSpace, Scotts Valley, CA, 1 edn. (2009)
29. World Health Organization (WHO): Snakebite envenoming - Key Facts 2019 (2019), https://www.who.int/news-room/fact-sheets/detail/snakebite-envenoming, [last accessed: 2020-06-10]
30. Xie, Q., Luong, M.T., Hovy, E., Le, Q.V.: Self-training with Noisy Student improves ImageNet classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10687–10698 (06 2020)