EfficientNets and Vision Transformers for Snake Species Identification Using Image and Location Information

FHDO Biomedical Computer Science Group (BCSG)

Louise Bloch[1,2], Christoph M. Friedrich[1,2]

[1]Department of Computer Science, University of Applied Sciences and Arts Dortmund (FHDO), Emil-Figge-Straße 42, 44227 Dortmund, Germany

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

Abstract

The automatic classification of snake species based on non-standardized photographs is important to improve the care of patients suffering from snake bites. The SnakeCLEF 2021 challenge, provides a large database containing images and their recording location of 772 snake species to overcome this problem. This paper describes the participation of the FHDO Biomedical Computer Science Group (BCSG) in this challenge. In the experiments, deep learning-based EfficientNets and Vision Transformer (ViT) models were trained. In a subsequent step, the prior probabilities of the location information were multiplied with the model predictions. An ensemble of both deep learning models achieved the best results, which was a macro averaging F_1-score across countries of 82.88 % for the independent test set.

Keywords

Snake species identification, EfficientNets, Vision Transformer, Image classification, Metadata inclusion

1. Introduction

This paper presents the participation of University of Applied Sciences and Arts Dortmund (FHDO) Biomedical Computer Science Group (BCSG) at the Conference of Labs of the Evaluation Forum (CLEF) 2021[1] SnakeCLEF challenge[2] for snake species identification [1]. This challenge is part of the LifeCLEF 2021 [2] research platform that focuses on the automated identification of species [3]. The LifeCLEF platform consists of four data-driven challenges.

The mortality of snakebites is between 81,000 and 138,000 people per year [4]. Annually, 400,000 victims of snakebites suffering from incurable physical and psychological disabilities [4]. It was expected that the mortality of snakebites can be reduced by identifying the snake species and thus recently administer the right antivenom [5]. Additionally, snake species identification

CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania

louise.bloch@fh-dortmund.de (L. Bloch); christoph.friedrich@fh-dortmund.de (C. M. Friedrich)
https://www.fh-dortmund.de/personen/Christoph-Friedrich/index.php (C. M. Friedrich)
0000-0001-7540-4980 (L. Bloch); 0000-0001-7906-0038 (C. M. Friedrich)

CEUR Workshop Proceedings (CEUR-WS.org)

[1] Conference of Labs of the Evaluation Forum (CLEF) 2021: http://clef2021.clef-initiative.eu/, [Last accessed: 2021-07-01]

[2] SnakeCLEF 2021: https://www.imageclef.org/SnakeCLEF2021, [Last accessed: 2021-07-01]

can also reduce the number of snakes that were killed out of peoples fear and thus might improve the protection of harmful snakes [6].

The aim of the SnakeCLEF challenge is, to deploy data-driven analysis to improve the identification of snake species based on non-standardized photographs. This paper summarizes the experiments and results of FHDO BCSG for the SnakeCLEF 2021 challenge. The presented approach expands the FHDO BCSG submissions [7] for SnakeCLEF 2020.

The article is structured as follows: Section 2 describes related work in this field of research. Afterwards, Section 3 gives a summary of the dataset and Section 4 describes the Machine Learning (ML) workflow and the methods used to implement it. Section 5 shows the results achieved using this workflow. Finally, the results are summarized and concluded in Section 6 which also mentions limitations and gives an outlook about future work.

2. Related Work

The identification of snake species has been previously investigated using ML. However, the manual extraction of features to describe snake species is tedious. Taxonomic features have been previously extracted in a semiautomatic approach [8] to identify six species in 1,299 images. Additionally, snake species identification is often performed using field-based investigations, which contain unstructured and non-standardized photographs often of poor image quality. Thus, most ML models used textural features or deep learning approaches to automatically extract features from snake images.

A textural approach [9] extracted Color and Edge Directivity Descriptor (CEDD) [10] features to differentiate between 22 Malaysian snake species. The dataset was recorded at the Perlis Snake Park, Malaysia and contained 349 images. The rarest species in this dataset included only three images. Five classical ML models were applied for the final classification. The best classification accuracy of 89.22 % was achieved by the nearest neighbour classifier.

Recently published studies [11, 12, 13, 14, 15] often used deep learning-based approaches to distinguish between snake species. Some of the studies were designed as an Object Detection (OD) task.

For example, the mean Average Precision (mAP) of different deep learning-based OD methods were compared to each other [11] to distinguish 1,027 images of eleven Australian snake species. The dataset was extracted from ImageNet [16] and was augmented by a Google Image search[3] and the least frequent class contained 60 images. The best mAP was achieved for a Faster Region-Based Convolutional Neural Network (Faster R-CNN) [17] with a ResNet [18] backbone.

Another similar approach [12] used Faster R-CNN with different detection layers. In this approach, 250 images of nine species, all occurring on the Galápagos Islands, Ecuador were distinguished from each other. The dataset was collected using three data sources, two internet searches performed on the platforms Google and Flickr and an image dataset provided by the Ecuadorian Institution of Tropical Herping[4]. The ResNet backbone achieved the best accuracy of 75 %.

[3] Google Image Search: https://images.google.com/imghp?hl=de&gl=de&gws_rd=ssl, [Last accessed: 2021-07-01]

[4] Tropical Herping: https://www.tropicalherping.com/, [Last accessed: 2021-07-01]

Further studies performed deep learning-based classification tasks. For example, one approach [13] compared the accuracies of three different classification networks namely VGG16 [19], DenseNet161 [20] and MobileNetV2 [21]. The dataset contained 3,050 images containing 28 species. The GrabCut [22] algorithm was applied as a preprocessing step to extract the snakes from the image background. After 50 training epochs, an accuracy of 72 % was reached for the test dataset and the DenseNet161 architecture.

Another approach [14] trained a deep Siamese network [23] for one-shot learning [24] to classify between 84 snake species based on the World Health Organization (WHO) venomous snake database[5]. The dataset contained 200 images and three to 16 images per class.

Although there are more than 3,700 snake species worldwide [25], and more than 600 of them were medically relevant [25], most recently ML approaches were trained on a small number of snake species.

The SnakeCLEF challenge [1, 25] provided a large dataset containing images of more than 700 snake species to overcome this problem. Different deep learning approaches were successfully submitted in previous rounds of this challenge. The winning approach [26] of SnakeCLEF 2020 [27] used a ResNet architecture pre-trained on ImageNet-21k and reached a macro-averaging F1-score of 62.54 %. The FHDO BCSG [7] combined OD and image classification using a Mask Region-Based Convolutional Neural Network (Mask R-CNN) [28] instance detection framework and an EfficientNet-B4 [29] classification model. This method reached a macro-averaging F1-score of 40.35 %. In post-competition submissions, the score could be optimized to 59.4 %.

This research expands the ML workflow developed from FHDO BCSG [7] in SnakeCLEF 2020. In particular, the workflow was extended to use Vision Transformer (ViT) [30] models and an ensemble of ViTs and EfficientNets.

3. Dataset

The training dataset of the SnakeCLEF 2021 and AICrowd Snake Species Identification Challenge round 5 consists of 386,006 photographs of 772 snake species. Those photographs were taken in 188 countries. The photographs originated from three different data sources, two online biodiversity platforms, namely iNaturalist[6] (n=277,025 images; 71.77 %), and Herpmapper[7] (n=58,351 images; 15.12 %) and another source containing noisy data downloaded from Flickr[8] (n=50,630 images; 13.12 %). The entire dataset was split into a training set containing 347,405 photographs (90.00 %), and a validation set containing 38,601 subjects (10.00 %). The class distribution of the united training and validation set is visualized in Figure 1. It can be seen that the dataset was highly imbalanced. Both subsets followed similar class distributions and each class was represented in both datasets. The test dataset consists of 23,673 images.

[5] Venomous snakes distribution and species risk categories: https://apps.who.int/bloodproducts/snakeantivenoms/database/, [Last accessed: 2021-07-01]

[6] iNaturalist: https://www.inaturalist.org/, [Last accessed: 2021-07-01]

[7] Herpmapper: https://www.herpmapper.org/, [Last accessed: 2021-07-01]

[8] Flickr: https://www.flickr.com/, [Last accessed: 2021-07-01]

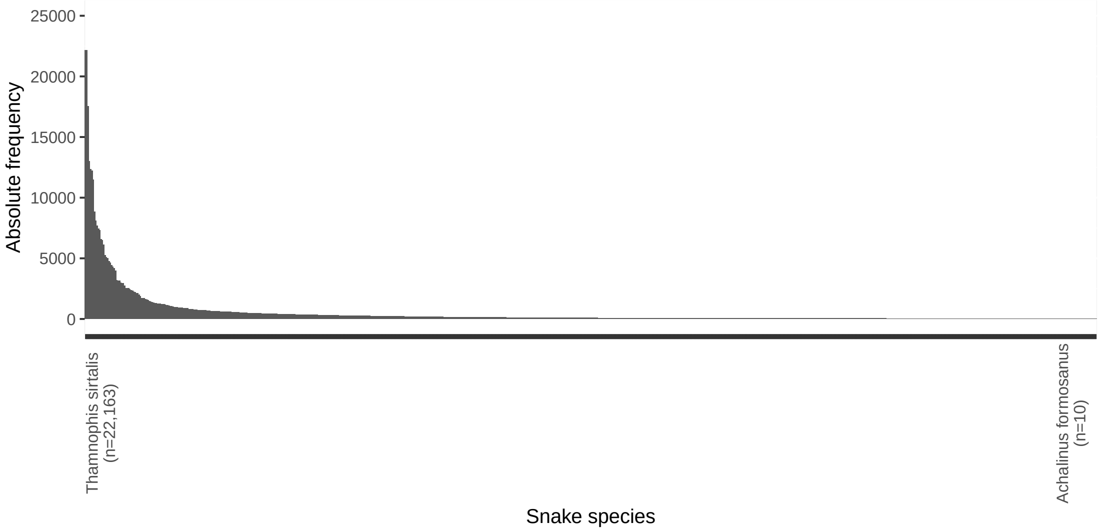

Figure 1: Distribution of the snake species in the unified training and validation set.

3.1. Metadata

In addition to the photographs, metadata that provides information about the continent and country of the image location were available. Most images (n=246,482; 63.85 %) were recorded in the United States of America. For 50,879 (13.18 %) images, no country information was provided and 51,061 images (13.23 %), had no continent information. Those images were marked with the "unknown" flag.

4. Methods

The ML workflow used to learn the differences between snake species is visualized in Figure 2. This ML workflow was implemented in a modular way, thus, during the challenge different combinations of workflow parts and their interactions were investigated. The entire workflow was implemented using the programming language Python v3.6.9 [31].

The preprocessing stage starts with optional filtering of the dataset. Afterwards, optional OD was implemented, which was trained to detect single snakes in the photographs. Due to time constraints, no models using the OD could be submitted during the challenge. The OD stage was followed by an image preprocessing stage which produced images of uniform, quadratic size. Afterwards, data augmentation was used to make the subsequent deep learning models more robust, for example, against rotation, scaling, and noise. EfficientNets and ViTs were trained to distinguish between the snake species. Finally, optional multiplication of the models' prediction probabilities and the a priori probability distribution of the snake species given the location was implemented.

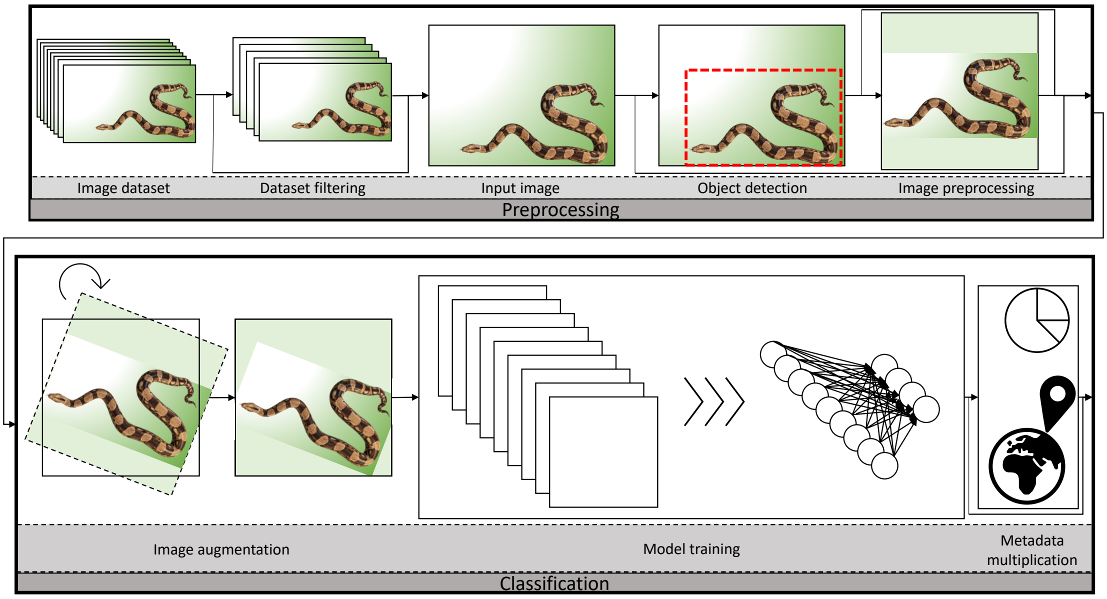

Figure 2: ML workflow used to differentiate between snake species.

4.1. Dataset Filtering

An analysis of the dataset with AntiDupl[9] revealed 29 "Image not found" images, which were the result of download problems.

Another problem that has been found are out-of-class images appearing in the Flickr dataset. 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 classifier with 1,000 classes and based on a ResNet50 [18] architecture has been used. Therefore, a positive list of 46 snake and reptile classes (e.g., *garter_snake*, *sidewinder*, . . .) that are part of the ImageNet Large Scale Visual Recognition Challenge 2012 (ILSRVC2012) [32] dataset has been used. With this classifier, 6,110 out-of-class images (10.47 %) have been identified as out-of-class images in the Flickr dataset. The excluded images were assigned to 384 species. The filtered dataset contained images of all 772 snake species. The least frequent species after the filtering were "bolyeria multocarinata" and "echinanthera melanostigma", both containing one image. The out-of-class images were manually checked and a large proportion of the identified images visualize no snakes. Due to reasons of time limitations during the competition, the out-of-class images were not formally validated. The effects of the reduced dataset have been tested and compared to the unfiltered dataset.

[9] https://github.com/ermig1979/AntiDupl, [Last accessed: 2021-07-01]

4.2. Object Detection

The aim of using OD as a preprocessing step to improve image classification was, to focus the subsequently implemented classification models on the object that should be classified. The idea of using OD for the identification of snake species in photographs before executing the image classification was inspired by the winning team [33] of round 2 of the AICrowd Snake Species Identification Challenge [25]. The OD used in this paper was already implemented and described in the contribution of the FHDO BCSG in the SnakeCLEF 2020 challenge [7]. A Mask R-CNN [28] model was trained to identify snakes in non-standardized photographs. The Mask R-CNN implements instance segmentation, which is a combination of OD and semantic segmentation. Thus, in each image, bounding boxes are identified and classified for each object of interest and each pixel of those bounding boxes was segmented into a range of given classes. Mask R-CNN is an extension of the OD method Faster R-CNN. The Mask R-CNN architecture consists of two phases, in the first phase, identically to Faster R-CNN, a Region Proposal Network (RPN) was implemented to identify candidate bounding boxes followed by a non-maximum suppression [34] to focus on the most promising candidates. The second stage first used a Region Of Interest (ROI) Align Network on the remaining candidate bounding boxes followed by a parallel implementation of Fully Connected Networks (FCN) to identify the object class and the offset of the bounding boxes as well as a Convolutional Neural Network (CNN) for the semantic segmentation task.

In this research, the Mask R-CNN was only used as an OD framework to differentiate between snakes and background and did not use the whole instance segmentation pipeline. Due to this reason, it may be an adequate alternative to use Faster R-CNN instead of the Mask R-CNN. The comparison of both models in the TensorFlow OD Application Programming Interface (API) [35] for the Microsoft Common Objects in Context (COCO) dataset [36] shows an increased mAP for the Mask R-CNN model. ResNet-50 was used as the backbone network and was pre-trained on the ImageNet-1k dataset. Based on this model, the OD training process was split into two phases. In the first warm-up phase, the newly added layers were trained for 20 epochs. Afterwards, the weights of the entire model were fine-tuned for another 30 epochs.

An adaption[10] to Tensorflow 2.1.0 of the implementation of the Mask R-CNN model implemented by Abdulla[11] has been used to implement the Mask R-CNN. The OD process included no data augmentation. A threshold of 0.3 was used for the minimum detection confidence. Stochastic Gradient Descent (SDG) was used to optimize the model weights, with a momentum value of 0.9, a weight decay of 10^{-4} and a mini-batch size of 8.

The annotated snake images were available from the winning solution of round 2 [33] of the AICrowd Snake Species Identification Challenge. This dataset contained annotated bounding boxes for 1,426 snake images which was a subset of round 2 [33] of the AICrowd Snake Species Identification Challenge.

The risk of using OD as a preprocessing step for species identification was that important background information was excluded from the images.

[10] DiffProML Mask R-CNN: https://github.com/DiffPro-ML/Mask_RCNN, [Last accessed: 2021-07-01]

[11] Matterport Mask R-CNN: https://github.com/matterport/Mask_RCNN, [Last accessed: 2021-07-01]

4.3. Image Pre-processing

Deep learning models expect the input of squared images. However, the ROIs which were extracted from the OD procedure do not have predefined dimensions. Thus, the ROIs need to be transformed to the image dimensions of the deep learning model. If all ROI dimensions were larger than the image dimensions of the deep learning model, the ROIs were resized. Images that were smaller than the image dimensions of the deep learning model, were not resized. The aspect ratio of the ROIs was not modified during the image preprocessing. Thus, occurring borders were padded using a specified color. In this approach, the image borders, which result from non-squared OD results were padded with the mean color of the truncated image parts, to find a color that matched the image best [37].

4.4. Data Augmentation

Data augmentation has been used to increase the training image dataset by adding slightly modified copies from existing training images to the training dataset. This method helps to reduce overfitting in deep learning models [38]. For the subsequently used classification models, different data augmentation procedures were implemented.

For all EfficientNet models, the data augmentation pipeline includes random cropping of the images from a size of 430×430 pixels to 380×380 pixels, random rotation in the range of $\pm 40°$, a width shift, height shift, random shearing and zooming each with a factor of 0.2 as well as a random horizontal flipping. During the data augmentation procedure, missing image positions were padded with the color of the nearest pixel neighbour. Additionally, the Lanczos interpolation [39] was used.

For the ViT models, the augmentation pipeline included random cropping from an image size of 250×250 to 224×224, and a random horizontal as well as vertical flip each with a probability of 0.5.

4.5. Classification

Two different model types, as well as an ensemble, were trained to detect snake species. EfficientNets were already used in the SnakeCLEF 2020 challenge from the FHDO BCSG [7]. In comparison to those models, ViT models were used to compare their results and to examine if an ensemble of both models can improve the classification results.

4.5.1. EfficientNets

EfficientNets [29] are an optimized CNN based model family. The base architecture of EfficientNets was developed by a CNN architecture search, which was optimized for accuracy and Floating Point Operations Per Second (FLOPS). The main building blocks of the resulting EfficientNets are Mobile Inverted Bottleneck Convolutional (MBConv) layers. The base model is successively scaled up using a uniform balance between model depth, model width and image resolution. The developed model architecture achieved state of the art performances on the ImageNet classification task while being smaller and faster than many of the compared models [29].

In this work, EfficientNet-B4 models were trained to differentiate between snake species. All images were scaled to an image size of 380×380 pixels. The model weights were initialized by a model which was pre-trained using noisy student [40]. This model was extended by adding a flatten layer, a dense layer with 1,000 neurons using a Swish [41] activation function and an output dense layer with 772 neurons using the softmax activation function. The entire model contained 276,495,588 parameters.

A warm-up phase of three epochs was used to train the weights of the newly added layers and the Batch Normalization layers. Afterwards, the weights of all layers were optimized for a larger number of epochs. All EfficientNets were optimized by using an Adam optimizer [42] with the parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-7}$. In both phases, a mini-batch size of 11 was used. The warm-up phase used a learning rate of 10^{-4} and afterwards, the learning rate was decreased to 10^{-6}.

Figure 1 shows, that the training dataset is imbalanced across snake species. During the SnakeCLEF 2020 challenge, the FHDO BCSG tested different oversampling techniques for snake identification using EfficientNets [7]. In this research, the class weight oversampling function which performed best in the SnakeCLEF 2020 and is described in Equation 1 was implemented. The oversampling rate increases less for classes with very small frequencies in comparison to a linear oversampling function. $F(c)$ denoted the frequency of class c.

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

4.5.2. Vision Transformers

ViTs [30] are an alternative to classical CNN deep learning-based image classifiers. Instead of using convolutional operations which focus on local parts of the image, ViTs consider the whole image in parallel. ViT were based on the self-attention principle [43], which was previously applied to Natural Language Processing (NLP). The model design of ViTs follows the Transformers which have been introduced in NLP [43]. In ViTs, the input image is first disassembled into a set number of patches. Additionally, the position of the patch is encoded by a positional encoder. Multiple transformer encoders, which each mainly consists of Multiheaded Self-Attention (MSA) layers and a Multi-Layer Perceptron (MLP) were used to encode the image patches. Another MLP is trained to learn the overall classification from the image encoding.

Two different ViT models were trained in this research. First, a ViT Base model architecture with an image size of 224×224 pixels, a patch size of 16×16 pixels which was pre-trained on the ImageNet21k dataset was used. This model was trained using a mini-batch size of 70, a learning rate of 10^{-5} an Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$) and cross-entropy loss. The ViT Base model contained 86,392,324 parameters.

During the challenge, another ViT model was trained. The architecture of this model was a ViT Large model with an image size of 224×224 pixels and a patch size of 16×16 pixels. This model was also initialized using the weights achieved for the ImageNet21k dataset. During the training of this model, a mini-batch size of 18 was used and a learning rate of 10^{-5}. This model was also trained using an Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$) and cross-entropy loss. The ViT Large model contained 304,092,932 parameters.

The pre-trained ViT models were loaded from the Python library timm v0.4.8 [44] and PyTorch v1.8.0+cu101 [45] was used to train the model.

No class weight oversampling was used for all ViT models in our experiments. Preliminary experimental tests on the training dataset showed decreased classification results using the class weight function used for the EfficientNets with the described ViT models and given hyperparameters. However, due to reasons of time limitations during the challenge, no complex hyperparameter optimization was applied.

4.5.3. Ensemble Model

In addition, a simple ensemble model was added in the experiments. This model ensembles the results of a ViT Large model and an EfficientNet-B4 model, by calculating the mean softmax-scaled probability predictions of both models. It was expected that the ensemble model outperforms the results of the individual models.

4.6. Adding Location Information

The last part of the ML workflow was to optionally add location information to probability predictions of the classification models. For this reason, the prior probabilities were estimated by the relative frequency distribution of the snake species at a location in the training and validation dataset. Two different strategies were applied to combine the location and the image information. In the first strategy, the prior probabilities of the image location were multiplied with the prediction probabilities of the image. The second strategy was similar to the first one, except that the prior probabilities were binarized with a cut-off value of 0. Thus, in the binarized strategy, prior probabilities with non-zero values were transformed to one, whereas prior probabilities with a value of zero were kept unchanged. Both strategies primarily use the country in this step, as it contains more information than the continent. The continent information was only added for images with unknown country information. For images with missing country and continent information, the prior probability of the "unknown" class was used.

5. Results

The following sections describe the classification results of the ML workflow and different ablation studies executed to the workflow modules for the validation and the test dataset.

5.1. Preliminary Experiments on the Validation Dataset

Table 1 summarizes the macro-averaging F_1-scores of the classification models for the original (F_1^{valid}) and the filtered ($F_1^{valid-f}$) version of the validation dataset. The experiments included the use of deep learning-based classifiers (introduced in Sec. 4.5), metadata encoding strategies (introduced in Sec. 4.6), a filtering strategy (introduced in Sec. 4.1) and the used training dataset (t: only training dataset, t+v: unified training and validation dataset). For each model, the results of three versions were provided. The results of the base models are named by an "S" followed

Table 1
Macro-averaging F_1-scores achieved for the official validation dataset (F_1^{valid}) and the validation dataset which was filtered for out-of-class images ($F_1^{valid-f}$). The EfficientNet-B4 classifier was abbreviated as B4, the ViT Base model was abbreviated as ViT-B and the ViT Large model was abbreviated as ViT-L. The ensemble model was an ensemble of model S1 and model S5. All models were trained without the OD and image preprocessing steps. Models, which were exclusively trained on the training dataset are denoted with a "t", models, which were trained on the training and validation dataset are denoted as "t+v". The best results are highlighted in bold.

ID	Classifier	Epochs	Location	Dataset filtering	Training dataset	F_1^{valid}	$F_1^{valid-f}$
S1	B4	123	-	-	t	48.70 %	48.82 %
C_S1	B4	123	country	-	t	60.13 %	60.62 %
B_S1	B4	123	binary	-	t	60.49 %	60.85 %
S2	B4	113	-	-	t	48.25 %	48.82 %
C_S2	B4	113	country	-	t	59.69 %	60.14 %
B_S2	B4	113	binary	-	t	60.15 %	60.59 %
S3	B4	113	-	yes	t	44.96 %	45.72 %
C_S3	B4	113	country	yes	t	57.08 %	58.16 %
B_S3	B4	113	binary	yes	t	57.37 %	58.47 %
S4	B4	25	-	yes	t+v	38.87 %	39.89 %
C_S4	B4	25	country	yes	t+v	51.09 %	52.17 %
B_S4	B4	25	binary	yes	t+v	51.34 %	52.59 %
S5	ViT-L	13	-	-	t	41.86 %	42.54 %
C_S5	ViT-L	13	country	-	t	52.57 %	54.65 %
B_S5	ViT-L	13	binary	-	t	54.73 %	55.71 %
S6	ViT-B	50	-	-	t	27.96 %	28.35 %
C_S6	ViT-B	50	country	-	t	43.71 %	44.36 %
B_S6	ViT-B	50	binary	-	t	41.71 %	42.38 %
S7	Ensemble	-	-	-	t	52.29 %	52.47 %
C_S7	Ensemble	-	country	-	t	58.74 %	59.47 %
B_S7	Ensemble	-	binary	-	t	**63.20 %**	**63.76 %**

by a single number, a "C_" was added as a prefix for all models which used a multiplication of the location information and a "B_" was added as a prefix for the binary country encoding.

Model S4 was trained on the training and validation set and thus overestimated the F_1-score for the validation dataset. The idea of this model was, to improve the results on the test set.

Figure 3 compares the F_1^{valid}-scores of the ablation study executed for different classifiers. The validation results of model S5 and model S6 showed improved performances for the ViT Large model in comparison to the ViT Base model, although, the ViT Large model was only trained for 13 epochs, whereas the ViT Base model was trained for 50 epochs. Model S1, which was an EfficientNet-B4 model trained for 123 epochs achieved better classification results for the validation dataset than the ViT Large model. However, the overall comparison of these models is quite complex due to differences in the training pipelines. The best results for both, the original and the filtered validation dataset, were achieved for the ensemble model S7. Ensemble model B_S7 which used the binary encoding of the location information achieved an F_1^{valid}-score of

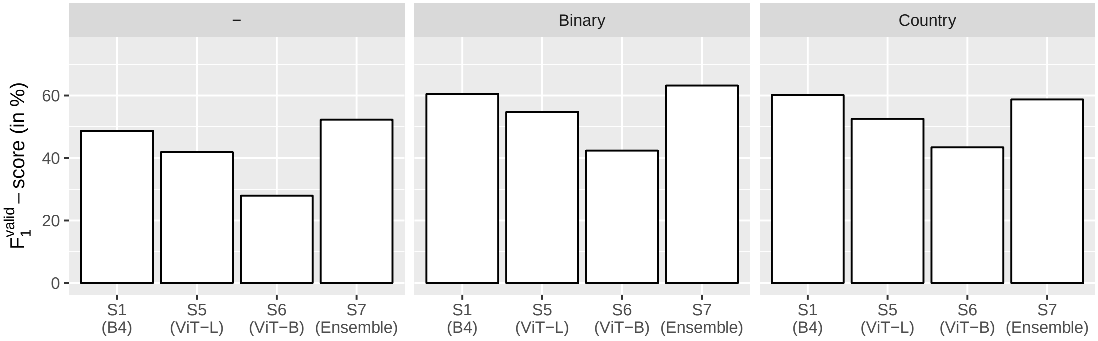

Figure 3: Barplot to compare the F_1^{valid}-scores achieved for different classifiers. The EfficientNet-B4 classifier was abbreviated as B4, the ViT Base model was abbreviated as ViT-B and the ViT Large model was abbreviated as ViT-L. The ensemble model was an ensemble of model S1 and model S5.

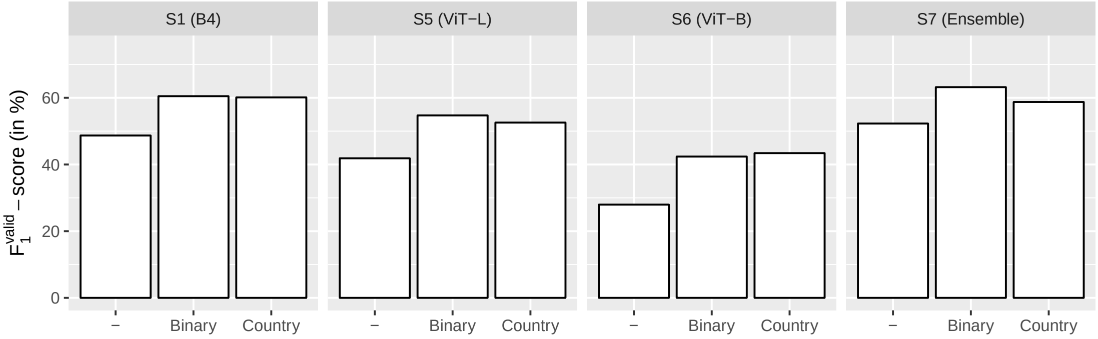

Figure 4: Barplot to compare the F_1^{valid}-scores achieved for the different location information strategies. The EfficientNet-B4 classifier was abbreviated as B4, the ViT Base model was abbreviated as ViT-B and the ViT Large model was abbreviated as ViT-L. The ensemble model was an ensemble of model S1 and model S5.

63.20 % and an $F_1^{valid-f}$-score of 63.76 %. Thus, the ensemble model outperformed the results of the individual models.

Figure 4 compares the F_1^{valid}-scores for the different metadata inclusion strategies. It can be seen, that all models showed increased classification results if the location information were multiplied with the model predictions. The binary encoding of the location information reached an additional increase of the performances for models S1, S2, S3, S4, S5 and S7.

Figure 5 compares the $F_1^{valid-f}$-scores achieved using the dataset filtering strategy. This comparison did not show a positive effect for the dataset filtering strategy in these experiments neither for the F_1^{valid}-score nor the $F_1^{valid-f}$-score. One reason for this effect might be the smaller number of different images for rare classes. Another problem in the experiments was that none of the EfficientNet models saturated for the validation set, thus it was hard to do a fair comparison between the models trained on the entire and the filtered dataset. Therefore, the effect of the dataset filtering was also investigated in the post-competition experiments

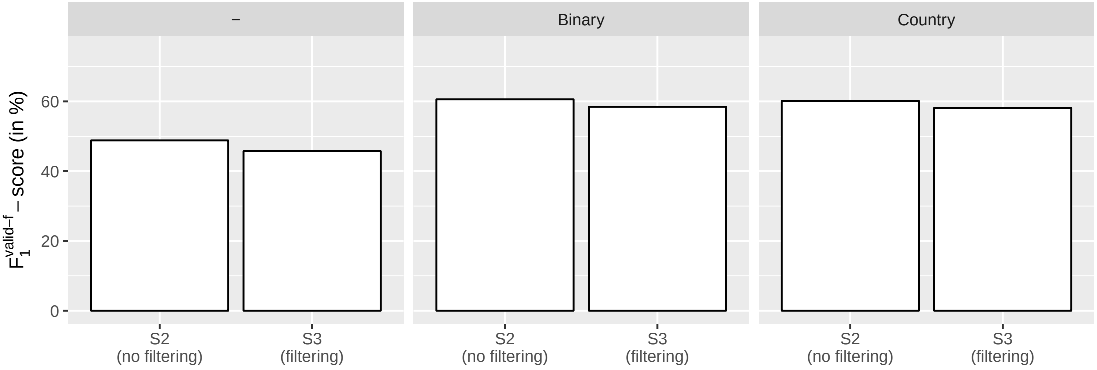

Figure 5: Barplot to compare the $F_1^{valid-f}$-scores achieved for the dataset filtering strategy.

described in Section 5.3.

5.2. Challenge Submissions and Results

Table 2 summarizes the results of the classification models for the official test dataset. Each team was able to submit up to ten submissions to the organizing team. Thus, some models in Table 1 were not validated for the test set. The primary metric for the SnakeCLEF 2021 challenge was the macro averaging F_1-score across countries for the independent test set ($F_{1_C}^{test}$). In addition, this table summarizes the macro-averaging F_1-scores (F_1^{test}) and the accuracy for the test set (ACC^{test}). Similar to Table 1, the experiments included the use of deep learning-based classifiers (introduced in Sec. 4.5), metadata encoding strategies (introduced in Sec. 4.6), a filtering strategy (introduced in Sec. 4.1) and the used training dataset (t: only training dataset, t+v: unified training and validation dataset). Due to time constraints, none of the submissions used the described OD during the challenge. Thus, the effect of the OD was investigated in the post-competition experiments described in Section 5.3. The submission IDs correspond to the IDs of Table 1 with the name of the team "FHDO-BCSG-" as a prefix.

Similar to the validation results, the comparison of model FHDO-BCSG-B_S6 and model FHDO-BCSG-B_S5 showed that the ViT Large model architecture outperformed the ViT Base architecture, although the ViT Base model was trained using a larger number of epochs. The ViT Large model achieved an $F_{1_C}^{test}$-score of 77.39 %. Additionally, model FHDO-BCSG-B_S1, which was the best performing EfficientNet-B4 model, reached better results than model FHDO-BCSG-B_S5, which was the ViT Large model. Model FHDO-BCSG-B_S1 reached an $F_{1_C}^{test}$-score of 78.33 %. Both models were trained with different data augmentation pipelines and learning parameters, which make it hard to do a fair model comparison. The comparison of models FHDO-BCSG-B_S1, FHDO-BCSG-B_S5 and FHDO-BCSG-B_S7 showed that the ensemble model, which was a combination of model S5 and model S1 outperformed the individual models.

The comparison of models FHDO-BCSG-C_S7 and FHDO-BCSG-B_S7, showed that using binary location information achieved slightly better results than using the raw prior probabilities for the ensemble model. The results of Model FHDO-BCSG-C_S7, which multiplied the raw

Table 2
Classification results achieved for the official test dataset, including macro-averaging F_1-scores (F_1^{test}), macro averaging F_1-scores across countries ($F_{1_C}^{test}$) and classification accuracy (ACC^{test}). The EfficientNet-B4 classifier was abbreviated as B4, the ViT Base model was abbreviated as ViT-B and the ViT Large model was abbreviated as ViT-L. All models were trained without the OD and image pre-processing steps. Models, which were exclusively trained on the training dataset are denoted with a "t", models, which were trained on the training and validation dataset are denoted as "t+v". The ensemble model was an ensemble of model S1 and model S5. The best results are highlighted in bold.

Submission ID	Classifier	Epochs	Location	Dataset filtering	Training dataset	F_1^{test}	$F_{1_C}^{test}$	ACC^{test}
FHDO-BCSG-B_S1	B4	123	binary	-	t	75.28 %	78.33 %	89.14 %
FHDO-BCSG-B_S2	B4	113	binary	-	t	74.45 %	76.16 %	89.08 %
FHDO-BCSG-B_S3	B4	113	binary	yes	t	72.82 %	81.04 %	**91.17 %**
FHDO-BCSG-B_S4	B4	25	binary	yes	t+v	75.20 %	76.62 %	89.58 %
FHDO-BCSG-C_S5	ViT-L	13	country	-	t	[a]30.22 %	[b]51.17 %	[c]59.80 %
FHDO-BCSG-B_S5	ViT-L	13	binary	-	t	74.06 %	77.39 %	88.90 %
FHDO-BCSG-B_S6	ViT-B	50	binary	-	t	64.98 %	69.46 %	82.96 %
FHDO-BCSG-S7	Ensemble	-	-	-	t	70.58 %	72.56 %	87.85 %
FHDO-BCSG-C_S7	Ensemble	-	country	-	t	76.27 %	82.04 %	90.10 %
FHDO-BCSG-B_S7	Ensemble	-	binary	-	t	**78.75 %**	**82.88 %**	90.42 %

[a] This result was affected by a mistake during the softmax normalization in the initial submission. Post-competition investigations corrected this score to a value of 72.71 %.

[b] This result was affected by a mistake during the softmax normalization in the initial submission. Post-competition investigations corrected this score to a value of 79.89 %.

[c] This result was affected by a mistake during the softmax normalization in the initial submission. Post-competition investigations corrected this score to a value of 88.74 %.

prior probabilities of the location to the model predictions achieved worse results in all metrics in comparison to model FHDO-BCSG-B_S7, which used a binarized version of the location information. Due to a mistake during the softmax normalization in the initial submission, the differences observed by using both location information methods for the ViT Large model showed stronger differences. The post-competition resubmission of this model with corrected softmax normalization led to similar results of both models.

Model FHDO-BCSG-B_S3, which was trained on the filtered version of the training dataset, achieved slightly worse F_1^{valid} and F_1^{test}-scores but a better $F_{1_C}^{test}$-score in comparison to model FHDO-BCSG-B_S2, which was trained on the unfiltered dataset. Model FHDO-BCSG-B_S3 also reached the overall best ACC^{test}-score of 91.17 %.

In comparison to model FHDO-BCSG-B_S3, model FHDO-BCSG-B_S4 was trained on the unified training and validation dataset. Due to reasons of time limitations during the competition, this model was only trained for 25 instead of 113 epochs. This model showed that the classification results might be improved by training the EfficientNet-B4 model on the training and validation dataset.

The best model submitted by FHDO BCSG was model FHDO-BCSG-B_S7, which was an ensemble of model S1 and model S5 and added binary location to the prediction probabilities. This model reached an F_1^{test}-score of 78.75 %, an $F_{1_C}^{test}$ of 82.88 % and an ACC^{test} of 90.42 %. Model FHDO-BCSG-B_S7 reached fourth place in the SnakeCLEF 2021 challenge.

5.3. Post-Competition Experiments

As previously mentioned, due to reasons of time limitations during the competition, no fair comparison between a model that used OD and one that did not use OD was performed during the competition. The same applied to using the dataset filtering strategy. Therefore, some post-competition experiments were executed leading to the results presented in Table 3 and Table 4. Additionally, the implementation of the EfficientNet models showed some drawbacks. Mainly, the implementation limited the mini-batch size to a small number, which leads to a high number of required epochs before the model saturated. Therefore, the EfficientNet training pipeline was reimplemented after the competition deadline. This reimplementation was similar to the implementation of the ViT training pipeline and thus the pre-trained EfficientNet models were loaded from the Python library timm v0.4.8 [44] and PyTorch v1.8.0+cu101 [45] was used to train the models.

Following the implementation described in Section 4.5.1, EfficientNet-B4 models were used for all experiments. It should be mentioned that the reimplemented training pipeline differs from the pipeline described in Section 4.5.1. All models were pre-trained on the ImageNet dataset. Additionally, a mini-batch size of 25, a learning rate of 10^{-4} an Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$) and cross-entropy loss were used to train the models. No additional dense layer was introduced before the classification layer. Thus, the models contain 18,932,812 parameters. Similar to the ViT implementation, no class weights were used to train the models. A scheduler was implemented which reduced the learning rate by a factor of 0.1 if the classification accuracies did not improve for five epochs. Early stopping was implemented if the loss function showed no improvement for more than eight epochs. The maximum number of epochs chosen was 100. A mixed-precision strategy was implemented to speed up the training process and increase the mini-batch size.

Similar to the ViT models, the augmentation pipeline included random cropping from an image size of 418×418 to 380×380, and a random horizontal as well as vertical flip each with a probability of 0.5.

The experimental results for the validation dataset are summarized in Table 3. In comparison to the models submitted during the challenge, the results showed that the number of epochs to train the EfficientNet model was reduced from more than 100 to less than 10 epochs using the reimplemented pipeline. Nevertheless, the achieved results outperformed the challenge results.

The effect of two pipeline modules was investigated. First, the effect of the dataset filtering strategy was examined by training two models, model S8 was trained on the entire training dataset and model S9 used the filtered training set. As can be seen in Figure 6, the comparison of both models showed improved results for model S9. Additionally, the overall best F_1^{valid}-score of 70.92 % was achieved for model B_S9, which used the binary encoding of the location information. The best $F_1^{valid-f}$-score was 70.11 % reached by model C_S9. The best model trained on the entire training dataset was model B_S8, which reached an F_1^{valid}-score of 68.55 % and an $F_1^{valid-f}$-score of 68.85 %.

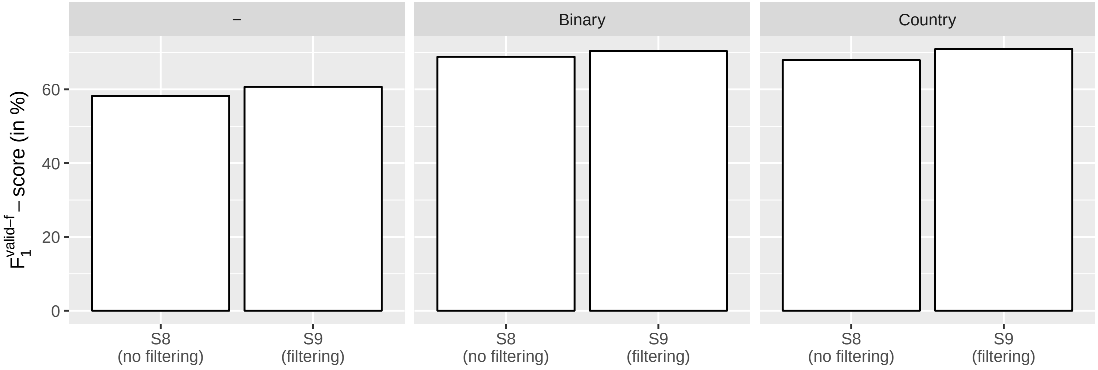

Figure 6: Barplot to compare the post-competition $F_1^{valid-f}$-scores achieved for the dataset filtering strategy.

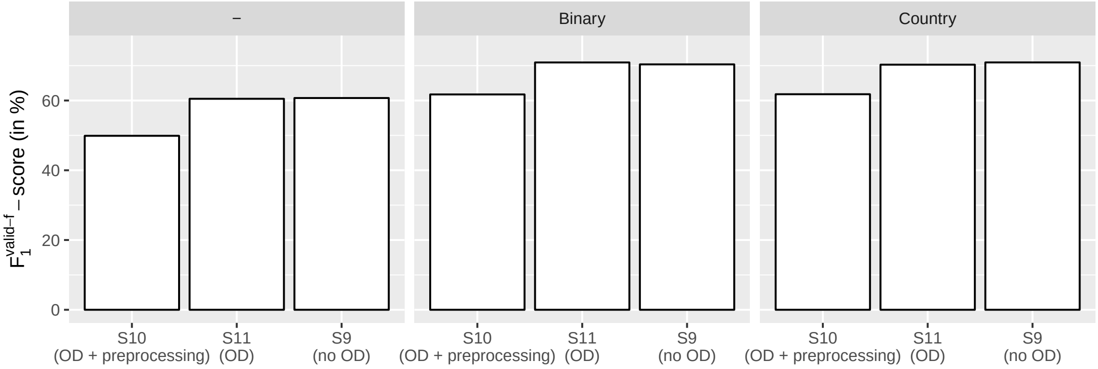

Figure 7: Barplot to compare the post-competition $F_1^{valid-f}$-scores achieved for the OD strategies.

The second experiment investigated the effect of the OD module and was visualized in Figure 7 using barplots. Therefore, the F_1-scores achieved by model S9 were compared to those reached for model S10, trained using the OD module described in Section 4.2. It can be noted that the results of model S10 reached worse results than model S9. Thus, for this model architecture, the OD module harmed the classification results. It was assumed that this reduction of the classification results might lead from the preprocessing pipeline used after the OD. For this reason, model S11 was trained using the OD module but no further image preprocessing. It can be noted that the results for model S11 outperformed the results of model S10, and reached similar results as model S9. However, the OD showed no improvement for the validation dataset in comparison to using the unprocessed images.

All post-competition models with the binary location encoding were evaluated for the test set. The achieved results are summarized in Table 4. Model B_S8 was the EfficientNet-B4 model that was trained on the entire training dataset. This model reached an $F_{1_C}^{test}$-score of 65.70 %. Model B_S9, which was trained for the filtered dataset, outperformed this model by reaching an

Table 3

Post-competition macro-averaging F_1-scores achieved after the challenge deadline for the official validation dataset (F_1^{valid}) and the validation dataset which was filtered for out-of-class images ($F_1^{valid-f}$). The EfficientNet-B4 classifier was abbreviated as B4. All models were trained on the training dataset. The best results are highlighted in bold.

ID	OD	Pre-processing	Classifier	Epochs	Location	Dataset filtering	F_1^{valid}	$F_1^{valid-f}$
S8	-	-	B4	4	-	-	58.12 %	58.24 %
C_S8	-	-	B4	4	country	-	68.20 %	67.91 %
B_S8	-	-	B4	4	binary	-	68.55 %	68.85 %
S9	-	-	B4	6	-	yes	60.85 %	60.71 %
C_S9	-	-	B4	6	country	yes	69.92 %	**70.92 %**
B_S9	-	-	B4	6	binary	yes	**70.11 %**	70.36 %
S10	OD	yes	B4	5	-	yes	48.99 %	49.88 %
C_S10	OD	yes	B4	5	country	yes	60.30 %	61.80 %
B_S10	OD	yes	B4	5	binary	yes	60.61 %	61.74 %
S11	OD	-	B4	5	-	yes	59.70 %	60.49 %
C_S11	OD	-	B4	5	country	yes	69.00 %	70.27 %
B_S11	OD	-	B4	5	binary	yes	69.32 %	70.91 %

Table 4

Post-competition classification results achieved for the official test dataset, including macro-averaging F_1-scores (F_1^{test}), macro averaging F_1-scores across countries ($F_{1_C}^{test}$) and classification accuracy (ACC^{test}). The EfficientNet-B4 classifier was abbreviated as B4. All models were trained on the training dataset. The best results are highlighted in bold.

ID	OD	Pre-processing	Classifier	Epochs	Location	Dataset filtering	F_1^{test}	$F_{1_C}^{test}$	ACC^{test}
B_S8	-	-	B4	4	binary	-	67.97 %	65.70 %	74.57 %
B_S9	-	-	B4	6	binary	yes	69.82 %	78.11 %	82.78 %
B_S10	OD	yes	B4	5	binary	yes	64.66 %	68.65 %	72.65 %
B_S11	OD	-	B4	5	binary	yes	**72.16 %**	**78.44 %**	**83.11 %**

$F_{1_C}^{test}$-score of 78.11 %.

The comparison between model B_S9, B_S10 and B_S11 investigated the effect of the OD module. Model B_S10 used both, the OD and image preprocessing pipelines and performed worse than model B_S9. However, model B_S11 that used only the OD module outperformed both models with an $F_{1_C}^{test}$-score of 78.44 %.

6. Conclusion

Overall, snake species identification is a challenging task, dealing with highly imbalanced class distributions, high intra-class variance and small inter-class variance.

The experiments investigated in this research, show that both, EfficientNets and ViTs can be successfully trained to distinguish between snake species. The best results were achieved by combining both model types using an ensemble model. ViT models which used the large architecture achieved better classification results than the ViT Base architecture.

The multiplication of location information with the model predictions can improve the results of both model types.

The training dataset was also filtered for images that did not include snakes. This filtering strategy showed improvements in the post-competition experiments for the test set. The visual validation of the out-of-class images showed some misclassified images that contained snakes. It was planned to improve this filtering strategy in future work to prevent those misclassifications using more recently deep learning models for this step.

Due to time constraints, only one model was trained on the entire training and validation dataset. It was expected that a model training on the entire dataset would increase the results for the test dataset. Additionally, during the competition, most models did not reach a plateau and thus the optimal number of epochs and the optimal classification results were not achieved for the submissions. This fact may also lead to worse performances for the filtering strategy during the competition and could be surmounted in the post-competition experiments. Another drawback that resulted from time constraints was that no models were evaluated for the OD method during the challenge. Post-competition experiments overcome this limitation and the OD showed improved test results when using no image preprocessing after the OD.

As has already been prepared in the post-competition experiments, future work will address the implementation of a more uniform implementation pipeline to enable a fair comparison of EfficientNets and ViT models. Additionally, the impact of different mini-batch sizes and learning rates should be investigated more systematically. To improve the classification results, up-scaled EfficientNet and ViT architectures should also be used and the pipeline will be enhanced using EfficientNetv2 classifiers [46]. Furthermore, the results of using different Transfer-Learning strategies should be compared to each other.

Acknowledgments

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

The authors thank Lukáš Picek, Department of Cybernetics, University of West Bohemia, Czechia, for post-competition evaluation.

References

[1] L. Picek, A. M. Durso, R. Ruiz De Castañeda, I. Bolon, Overview of SnakeCLEF 2021: Automatic Snake Species Identification with Country-Level Focus, in: Working Notes of the 12th International Conference of the CLEF Association (CLEF 2021): 21-24.09.2021; Bucharest, Romania, 2021.

[2] A. Joly, H. Goëau, S. Kahl, L. Picek, T. Lorieul, E. Cole, B. Deneu, M. Servajean, R. Ruiz De Castañeda, I. Bolon, H. Glotin, R. Planqué, W.-P. Vellinga, A. Dorso, P. Bonnet, I. Eggel, H. Müller, Overview of LifeCLEF 2021: A System-oriented Evaluation of Automated

Species Identification and Species Distribution Prediction, in: Proceedings of the 12th International Conference of the CLEF Association (CLEF 2021): 21-24.09.2021; Bucharest, Romania, 2021.

[3] A. Joly, H. Goëau, E. Cole, S. Kahl, L. Picek, H. Glotin, B. Deneu, M. Servajean, T. Lorieul, W.-P. Vellinga, P. Bonnet, A. M. Durso, R. R. de Castañeda, I. Eggel, H. Müller, LifeCLEF 2021 Teaser: Biodiversity Identification and Prediction Challenges, in: D. Hiemstra, M.-F. Moens, J. Mothe, R. Perego, M. Potthast, F. Sebastiani (Eds.), Proceedings of the European Conference on Information Retrieval (ECIR 2021): 28.03-01.04.2021; Online Event, Springer International Publishing, Cham, 2021, pp. 601–607. doi:10.1007/978-3-030-72240-1_70.

[4] J. M. Gutiérrez, J. J. Calvete, A. G. Habib, R. A. Harrison, D. J. Williams, D. A. Warrell, Snakebite Envenoming, Nature Reviews Disease Primers 3 (2017). doi:10.1038/nrdp.2017.63.

[5] H. F. Williams, H. J. Layfield, T. Vallance, K. Patel, A. B. Bicknell, S. A. Trim, S. Vaiyapuri, The Urgent Need to Develop Novel Strategies for the Diagnosis and Treatment of Snakebites, Toxins 11 (2019). doi:10.3390/toxins11060363.

[6] H.-T. Tseng, L.-K. Huang, C.-C. Hsieh, No More Fear of Every Snake: Applying Chatbot-Based Learning System for Snake Knowledge Promotion Improvement: A Regional Snake Knowledge Learning System, in: Proceedings of the IEEE 20th International Conference on Advanced Learning Technologies (ICALT 2020): 06-09.07.2020; Tartu, Estonia, 2020, pp. 72–76. doi:10.1109/ICALT49669.2020.00029.

[7] L. Bloch, A. Boketta, C. Keibel, E. M. an Alex Michailutschenko, O. Pelka, J. Rückert, L. Willemeit, C. M. Friedrich, Combination of Image and Location Information for Snake Species Identification using Object Detection and EfficientNets, in: Working Notes of the 11th Conference and Labs of the Evaluation Forum (CLEF 2020): 22-25.09.2020; Thessaloniki, Greece, 2020. URL: http://ceur-ws.org/Vol-2696/paper_201.pdf.

[8] A. James, D. Kumar, B. Mathews, S. Sugathan, Discriminative Histogram Taxonomy Features for Snake Species Identification, Human-Centric Computing and Information Sciences 4 (2014). doi:10.1186/s13673-014-0003-0.

[9] A. Amir, N. A. H. Zahri, N. Yaakob, R. B. Ahmad, Image Classification for Snake Species Using Machine Learning Techniques, in: S. Phon-Amnuaisuk, T.-W. Au, S. Omar (Eds.), Proceedings of the Computational Intelligence in Information Systems Conference (CIIS 2016): 18-20.11.2016; Brunei, Brunei Darussalam, Springer International Publishing, Cham, 2017, pp. 52–59. doi:10.1007/978-3-319-48517-1_5.

[10] S. A. Chatzichristofis, Y. S. Boutalis, CEDD: Color and Edge Directivity Descriptor: A Compact Descriptor for Image Indexing and Retrieval, in: A. Gasteratos, M. Vincze, J. K. Tsotsos (Eds.), Proceedings of the International Computer Vision Systems (ICVS 2008): 12-15.05.2008; Santorini, Greece, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 312–322. doi:10.1007/978-3-540-79547-6_30.

[11] Z. Yang, R. Sinnott, Snake Detection and Classification using Deep Learning, in: Proceedings of the 54th Hawaii International Conference on System Sciences (HICSS 2021): 05-08.1.2021; Maui, Hawaii, US, 2021. doi:10.24251/hicss.2021.148.

[12] A. Patel, L. Cheung, N. Khatod, I. Matijosaitiene, A. Arteaga, J. W. Gilkey, Revealing the Unknown: Real-Time Recognition of Galápagos Snake Species Using Deep Learning,

Animals 10 (2020) 806. doi:10.3390/ani10050806.

[13] M. Vasmatkar, I. Zare, P. Kumbla, S. Pimpalkar, A. Sharma, Snake Species Identification and Recognition, in: Proceedings of the IEEE Bombay Section Signature Conference (IBSSC 2020): 04-06.12.2020; Mumbai, India, 2020, pp. 1–5. doi:10.1109/IBSSC51096.2020.9332218.

[14] C. Abeysinghe, A. Welivita, I. Perera, Snake Image Classification Using Siamese Networks, in: Proceedings of the 3rd International Conference on Graphics and Signal Processing (ICGSP 2019): 01-03.06.2019; Hong Kong, Hong Kong, Association for Computing Machinery, New York, NY, USA, 2019, p. 8–12. doi:10.1145/3338472.3338476.

[15] I. S. Abdurrazaq, S. Suyanto, D. Q. Utama, Image-Based Classification of Snake Species Using Convolutional Neural Network, in: Proceedings of the International Seminar on Research of Information Technology and Intelligent Systems (ISRITI 2019): 05-06.12.2019; Yogyakarta, Indonesia, Institute of Electrical and Electronics Engineers (IEEE), 2019, pp. 97–102. doi:10.1109/isriti48646.2019.9034633.

[16] J. Deng, W. Dong, R. Socher, L. Li, K. Li, L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009): 20-25.06.2009; Miami Beach, Florida, US, Institute of Electrical and Electronics Engineers (IEEE), 2009, pp. 248–255. doi:10.1109/cvpr.2009.5206848.

[17] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (2017) 1137–1149. doi:10.1109/tpami.2016.2577031.

[18] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in: Proceesings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016): 27-30.07.2016; Las Vegas, Nevada, US, Institute of Electrical and Electronics Engineers (IEEE), 2016, pp. 770–778. doi:10.1109/cvpr.2016.90.

[19] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, in: Y. Bengio, Y. LeCun (Eds.), Conference Track Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015): 07-09.05.2015; San Diego, California, US, 2015. URL: http://arxiv.org/abs/1409.1556.

[20] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely Connected Convolutional Networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017): 21-26.07.2017, Honolulu, Hawaii, 2017, pp. 2261–2269. doi:10.1109/CVPR.2017.243.

[21] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, MobileNetV2: Inverted Residuals and Linear Bottlenecks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018): 18-22.06.2018, Salt Lake City, Utah, US, 2018, pp. 4510–4520. doi:10.1109/CVPR.2018.00474.

[22] S. Hua, P. Shi, GrabCut Color Image Segmentation Based on Region of Interest, in: Proceedings of the 7th International Congress on Image and Signal Processing (ICISP 2014): 30.06-02.07.2014; Cherburg, France, 2014, pp. 392–396. doi:10.1109/CISP.2014.7003812.

[23] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, R. Shah, Signature Verification Using a "Siamese" Time Delay Neural Network, in: J. Cowan, G. Tesauro, J. Alspector (Eds.), Advances in Neural Information Processing Systems (NIPS 1993): Denver, Colorado, US, volume 6, Morgan-Kaufmann, 1994. URL: https://proceedings.neurips.cc/paper/1993/file/

288cc0ff022877bd3df94bc9360b9c5d-Paper.pdf.

[24] G. Koch, R. Zemel, R. Salakhutdinov, Siamese Neural Networks for One-Shot Image Recognition, in: Proceedings of the Deep Learning workshop of the International Conference on Machine Learning (ICML 2015): 06-11.06.2015; Lille, France, volume 2, 2015. URL: https://www.cs.cmu.edu/~rsalakhu/papers/oneshot1.pdf.

[25] A. M. Durso, G. K. Moorthy, S. P. Mohanty, I. Bolon, M. Salathé, R. Ruiz de Castañeda, Supervised Learning Computer Vision Benchmark for Snake Species Identification from Photographs: Implications for Herpetology and Global Health, Frontiers in Artificial Intelligence 4 (2021) 17. doi:10.3389/frai.2021.582110.

[26] M. G. Krishnan, Impact of Pretrained Networks for Snake Species Classification, in: Working Notes of the 11th Conference and Labs of the Evaluation Forum (CLEF 2020): 22-25.09.2020; Thessaloniki, Greece, 2020. URL: http://ceur-ws.org/Vol-2696/paper_194.pdf.

[27] L. Picek, R. Ruiz De Castañeda, A. M. Durso, P. M. Sharada, Overview of the SnakeCLEF 2020: Automatic Snake Species Identification Challenge, in: Proceedings of the 11th Conference and Labs of the Evaluation Forum (CLEF 2020): 22-25.09.2020; Thessaloniki, Greece, 2020. URL: http://ceur-ws.org/Vol-2696/paper_258.pdf.

[28] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV 2017): 22-29.10.2017; Venice, Italy, Institute of Electrical and Electronics Engineers (IEEE), 2017, pp. 2980–2988. doi:10.1109/iccv.2017.322.

[29] M. Tan, Q. Le, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning (ICML 2019): 10-15.06.2019; Long Beach, California, US, volume 97, 2019, pp. 6105–6114. URL: http://proceedings.mlr.press/v97/tan19a.html.

[30] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, in: Proceedings of the 9th International Conference on Learning Representations (ICLR 2021): 03-07.05.2021; Online Event, 2021. URL: https://openreview.net/forum?id=YicbFdNTTy.

[31] G. Van Rossum, F. L. Drake Jr, Python Tutorial, Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995.

[32] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, International Journal of Computer Vision 115 (2015) 211–252. doi:10.1007/s11263-015-0816-y.

[33] Moorthy 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. URL: https://medium.com/@Stormblessed/diving-into-deep-learning-part-3-a-deep-learning-practitioners-attempt-to-build-state-of-the-2460292bcfb, [last accessed: 2021-05-24].

[34] R. Girshick, F. Iandola, T. Darrell, J. Malik, Deformable Part Models are Convolutional Neural Networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015): 07-12.06.2015; Boston, Massachusetts, US, 2015, pp. 437–446. doi:10.1109/CVPR.2015.7298641.

[35] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, K. Murphy, Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017): 21-26.07.2017; Honolulu, Hawaii, US, 2017, pp. 3296–3297. URL: https://openaccess.thecvf.com/content_cvpr_2017/papers/Huang_SpeedAccuracy_Trade-Offs_for_CVPR_2017_paper.pdf.

[36] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: Common Objects in Context, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Proceedings of the 13th European Conference on Computer Vision (ECCV 2014): 06-12.09.2014; Zurich, Switzerland, Springer International Publishing, Cham, 2014, pp. 740–755. doi:10.1007/978-3-319-10602-1_48.

[37] S. Koitka, C. M. Friedrich, Optimized Convolutional Neural Network Ensembles for Medical Subfigure Classification, in: G. J. Jones, S. Lawless, J. Gonzalo, L. Kelly, L. Goeuriot, T. Mandl, L. Cappellato, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction: Proceedings of the 8th International Conference of the CLEF Association (CLEF 2017): 11-14.09.2017; Dublin, Ireland, Springer International Publishing, Cham, 2017, pp. 57–68. doi:10.1007/978-3-319-65813-1_5.

[38] C. Shorten, T. M. Khoshgoftaar, A Survey on Image Data Augmentation for Deep Learning, Journal of Big Data 6 (2019). doi:10.1186/s40537-019-0197-0.

[39] C. E. Duchon, Lanczos Filtering in One and Two Dimensions, Journal of Applied Meteorology 18 (1979) 1016–1022. doi:10.1175/1520-0450(1979)018<1016:lfioat>2.0.co;2.

[40] Q. Xie, M.-T. Luong, E. Hovy, Q. V. Le, Self-Training with Noisy Student Improves ImageNet Classification, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020): 14-19.06.2020; Online Event, 2020, pp. 10687–10698. doi:10.1109/CVPR42600.2020.01070.

[41] P. Ramachandran, B. Zoph, Q. V. Le, Searching for Activation Functions, in: Proceedings of the Workshop Track of 6th International Conference on Learning Representations (ICLR 2018): 30.4-03.5.2018; Vancouver, Canada, 2018. URL: https://openreview.net/forum?id=SkBYYyZRZ.

[42] D. P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, in: Proceedings of the 3rd International Conference for Learning Representations (ICLR 2014): 14-16.04.2014; Banff, Canada, 2014. URL: https://arxiv.org/abs/1412.6980.

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, I. Polosukhin, Attention is All you Need, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems (NIPS 2017): 04-09.12.2017; Long Beach, California, US, volume 30, Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[44] R. Wightman, PyTorch Image Models, 2019. doi:10.5281/zenodo.4414861.

[45] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An Imperative Style, High-Performance Deep Learning Library, in: H. Wallach, H. Larochelle, A. Beygelzimer,

F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems (Neurips 2019): 08-14.12.2019; Vancouver, Canada, Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[46] M. Tan, Q. V. Le, EfficientNetV2: Smaller Models and Faster Training, in: Proceedings of the 38th International Conference on Machine Learning (ICML 2021): 18-24.07.2021; Online Event, 2021. URL: https://arxiv.org/abs/2104.00298.