A Deep Learning Method for Visual Recognition of Snake Species

Rail Chamidullin[1], Milan Šulc[1], Jiří Matas[1] and Lukáš Picek[2]

[1]Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague
[2]Department of Cybernetics, Faculty of Applied Sciences, University of West Bohemia

Abstract

The paper presents a method for image-based snake species identification. The proposed method is based on deep residual neural networks – ResNeSt, ResNeXt and ResNet – fine-tuned from ImageNet pre-trained checkpoints. We achieve performance improvements by: discarding predictions of species that do not occur in the country of the query; combining predictions from an ensemble of classifiers; and applying mixed precision training, which allows training neural networks with larger batch size. We experimented with loss functions inspired by the considered metrics: soft F1 loss and weighted cross entropy loss. However, the standard cross entropy loss achieved superior results both in accuracy and in F1 measures. The proposed method scored third in the SnakeCLEF 2021 challenge, achieving 91.6% classification accuracy, Country F1 Score of 0.860, and F1 Score of 0.830.

Keywords

Snake Species Identification, Fine-grained Classification, Computer Vision, Convolutional Neural Networks, Deep Learning

1. Introduction

The paper describes a method for automatic image-based snake species identification submitted by the CMP team to the SnakeCLEF 2021 challenge [1] – a part of LifeCLEF 2021 workshop [2]. The problem of identifying snake species from images is difficult because the classification is fine-grained, some species look very similar, and up to hundreds of different snake species live in one country.

Taxonomic knowledge about snakes is crucial in diagnosis and medical response to snakebites. Accurate identification of the snake species is important for the appropriate treatment of snakebite victims since specific antivenoms are effective against specific venomous snakes. Moreover, antivenoms should not be used to treat bites from non-venomous snakes because of side effects such as allergic reactions [3]. Snakebites are a global health problem that kills or disables half a million people a year in developing countries [3].

This paper is structured as follows: Section 2 describes related work focusing on snake species identification. Section 3 introduces the input data and evaluation methodology of the SnakeCLEF 2021 challenge. Section 4 describes the adopted architecture of deep neural network and the

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

chamirai@fel.cvut.cz (R. Chamidullin); sulcmila@fel.cvut.cz (M. Šulc); matas@fel.cvut.cz (J. Matas)
0000-0003-1728-8939 (R. Chamidullin); 0000-0002-6321-0131 (M. Šulc); 0000-0003-0863-4844 (J. Matas); 0000-0002-6041-9722 (L. Picek)

optimization procedure. Section 5 covers all experiments, ranging from preliminary experiments to the final challenge submissions. Finally, the results are summarized in Section 6.

2. Related Work

Before the existence of large-scale image datasets for snake species classification, Abeysinghe et al. [4] proposed a one-shot learning approach for fine-tuning a Convolutional Neural Network (CNN) for the task of snake species identification. The authors used a small dataset of 84 snake species, with most species having no more than 3 training images. The authors utilize a Siamese network [5] that ranks similarity between two inputs: The network is trained by binary cross entropy minimization to estimate the probability of the query image belonging to the same class as the reference image. At test time the query image is compared against all annotated reference images of each class.

In 2020, the first year of the SnakeCLEF challenge [6], introduced a dataset with 287,632 images of 783 snake species taken in 145 countries. Only two teams presented their recognition systems for identifying snake species.

The best scoring team in SnakeCLEF 2020, gokuleloop [7], fine-tuned ResNet-50-V2 [8] from ImageNet-1K and ImageNet-21K [9] pre-trained checkpoints, the latter leading to better results. The author applied the following training techniques:

- Gradient accumulation – a technique that accumulates gradients from small mini-batches allowing larger effective mini-batch size.
- Mixup augmentation [10] – an augmentation technique that combines random image pairs from the training dataset.
- Group normalization [11] – differently from batch normalization, GN divides the channels into groups and computes the mean and variance within each group.

The second team in SnakeCLEF 2020, FHDO_BCSG [12], first detected regions where snakes occur using a Mask R-CNN [13] object detector, and then classified the snake species in the regions using EfficientNet [14]. The authors adjusted the output probabilities of EfficientNet based on the geographic location of the image: The softmax values for each image were multiplied by the species a priori probability for a given geographic location. To clean the training dataset from noisy samples, the authors utilized an ImageNet-1K pre-trained ResNet-50 network and discarded images not classified as snake and reptile classes.

3. Challenge Description

3.1. Dataset

The training dataset provided by SnakeCLEF 2021 covers 772 snake species and contains annotated images from three different sources: iNaturalist, HerpMapper and Flickr. Examples of images are in Figure 1. The majority of images are from iNaturalist and HerpMapper, with 277,025 and 58,351 images, respectively. Their labels are confirmed by human annotators. The Flickr dataset is the smallest, with 50,630 web-scraped images that contain noisy data.

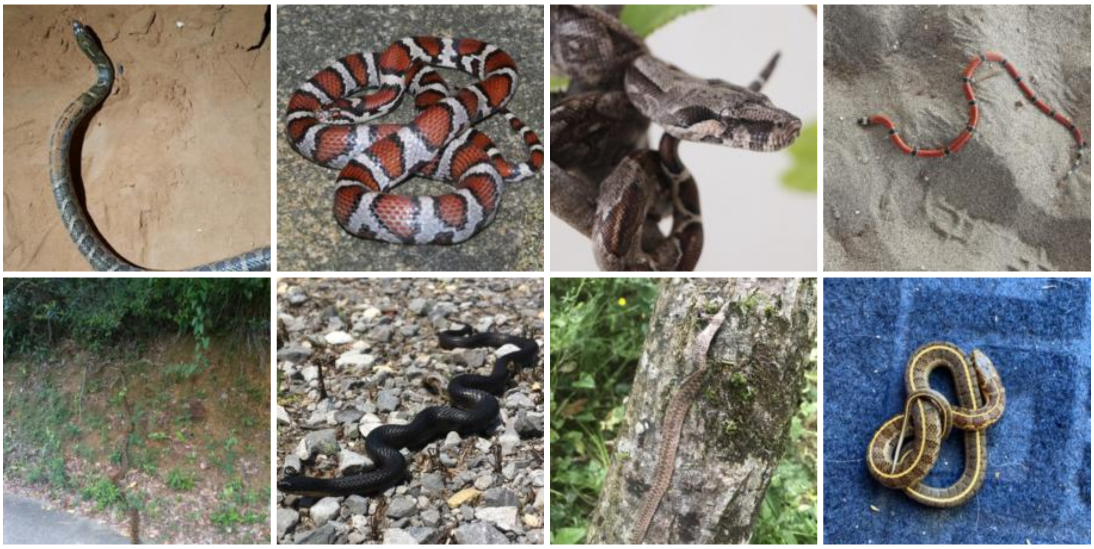

Figure 1: Image examples from the SnakeCLEF 2021 dataset. The images are resized and center cropped to 224×224. CC-BY-NC images from iNaturalist: ©srhein, ©jance, ©roig10, ©arturobtzz, ©John Clough, ©William Wimley, ©nobiscuits, ©feistygirl75.

In total, 386,006 images with annotations were provided. Training with external data was not allowed.

The challenge organizers suggested a subset of 70,208 images, referenced as a mini-subset in the rest of this paper, made of samples from INaturalist and HerpMapper. The experiments described in Section 5 are based on the said subset.

In addition to the images, the dataset contains metadata with information about the country where the image was taken. In total, the training dataset includes images from 188 countries. The dataset is fine-grained with a long tail class distribution. More than 22,000 images represent the most frequent species, while the least frequent species have only 10 images. The least represented species are often found in regions such as Middle and South America, South Africa and Australia. Table 1 shows the distribution of images in geographical regions. For some images, information about the geographical location is missing.

Furthermore, the challenge organizers provided 28,418 images without annotations. Top one species predictions for the test images were sent to the organizers to participate in the challenge.

3.2. Data Preparation

During the data exploration phase, we discovered that training and validation datasets contain noisy data from Flickr. The noisy data are non-relevant images with various animal species or objects. We estimate[1] that the percentage of non-relevant images is 10.6 ± 0.1, with 95% confidence interval. We decided to remove all Flickr images and proceeded with verified images from iNaturalist and HerpMapper.

[1] We used the Student's t-distribution with $n = 20$ samples and $n - 1$ degrees of freedom where each sample denotes the percentage of non-relevant images in a set of randomly selected 100 images.

Table 1
Geographical distribution of SnakeCLEF 2021 images.

Region	Number of images
North America	258,732
Europe	18,689
Middle America	17,403
Asia	16,518
South America	12,735
Africa	6,017
Australia	4,313
Oceania	538
unknown	51,061

The challenge organizers suggested a data split with 90% training and 10% validation samples. However, after removing Flickr images, it turned out that some species were not represented in the proposed validation set. Table 2 displays the number of snake classes represented, i.e. classes with at least one image, in the dataset sources. iNaturalist and HerpMapper combined have 768 classes which are all represented across the training set but only 733 classes in the validation set. We thus created a new dataset split where all classes are represented in both training and validation splits if more than one image of the species is available. If not, the image is placed in the training set.

Technically, the last 10% of images, ordered by metadata ID, for every species and country combination were selected as the validation data. One validation image was selected for the cases that had fewer than 10 images. We assume the ID ordering is random w.r.t. image content and properties.

Table 2
Number of species included in SnakeCLEF 2021 dataset sources: iNaturalist, HerpMapper and Flickr. The last row represents our new dataset split, after removing all Flickr images due to noisy labels and sampling a new validation set covering as many species as possible.

Source	Training set	Validation set	Number of images
iNaturalist	762	716	277,025
HerpMapper	603	357	58,351
Flickr	730	585	50,630
iNaturalist + HerpMapper + Flickr	772	772	386,006
iNaturalist + HerpMapper	768	733	335,376
Mini-subset (introduced in Section 3.1)	768	763	70,208
iNaturalist + HerpMapper (new split)	768	765	335,376

3.3. Evaluation Metrics

The challenge used two metrics for the final evaluation. The primary metric is the macro averaged F1 Score across countries ("Country F1 Score"), shown in equation 4. The secondary metric is the macro averaged F1 Score ("F1 Score"), shown in equation 2.

The F1 Score for each species $s = 1, 2, ..., k$ is computed as a harmonic mean of precision p_s and recall r_s:

$$F_{1s} = \frac{2p_s r_s}{p_s + r_s}. \tag{1}$$

The macro averaged F1 Score is the average of the F_1 scores of all species:

$$\text{macro}(F_1) = \frac{1}{k} \sum_{s=1}^{k} F_{1s}. \tag{2}$$

Country F1 Score CF_{1c} for each country $c = 1, 2, ..., m$ is the macro averaged F1 Score computed only for species living in country c:

$$\text{CF}_{1c} = \frac{\sum_{s=1}^{k} F_{1s} A_{cs}}{\sum_{s=1}^{k} A_{cs}}, \tag{3}$$

where A is a $k \times m$ matrix with elements $A_{cs} = \begin{cases} 1, & \text{country } c \text{ is a habitat of species } s \\ 0, & \text{otherwise} \end{cases}$.

Similarly, macro averaged Country F1 Score is obtained by averaging CF_{1c} over all countries:

$$\text{macro}(\text{CF}_1) = \frac{1}{m} \sum_{c=1}^{m} \text{CF}_{1c}. \tag{4}$$

The macro averaged Country F1 Score thus increases the importance of species that appear in more countries.

4. Methodology

The proposed method is based on the state-of-the-art Convolutional Neural Networks (CNNs) for image classification, described in Subsection 4.1. The following subsections describe the optimization procedure, loss functions, the post-processing of the predictions, applying mixed precision training and implementation details.

4.1. Deep Residual Networks

All experiments are based on deep residual neural networks, namely the original ResNet [15], the ResNeXt [16], and the recent ResNeSt [17]. The ResNet architecture consists of a stack of residual blocks – building modules with residual connections that combine input and output by element-wise addition. The ResNeXt additionally includes a split-transform-merge strategy, where each block performs a set of transformations with the same topology whose outputs are aggregated by element-wise addition. For example, a single transformation can be a group of convolutions. The ResNeSt incorporates a channel-wise attention strategy within each split-transform-merge block: Each transformation consists of split groups over which the network calculates the channel-wise split attention weights.

All networks in our experiments were fine-tuned from ImageNet-1K [18] pre-trained checkpoints. Residual networks typically [15, 16] use input size about 224×224, the pre-trained ResNeSt-101 and ResNeSt-200 are available with a larger input sizes of 256×256 and 320×320, respectively.

4.2. Optimization Procedure

We use two optimization algorithms for training CNN models: stochastic gradient descent with momentum (SGD) and Adam [19]. Our preliminary experiments showed that Adam optimizer is able to converge quickly, but the prediction score is inferior compared to SGD. The application of the one cycle schedule policy [20] (one cycle) improved the results when applied with the Adam optimizer while applying it with SGD did not work well in our preliminary experiments.

The training hyper-parameters, such as learning rate, momentum and weight decay, are listed in Table 3 and were set the same as in the network pre-training. Batch sizes were adjusted to fit the network on the graphics processing unit (GPU). The input image size stays the same as in the pre-trained networks.

During the training, we select the best checkpoint based on the highest validation Country F1 Score.

Table 3
The hyper-parameter setting used for training the challenge submissions.

Network	ResNeSt-101	ResNeSt-200	ResNeXt-101	ResNet-101
Optimizer	SGD	SGD	Adam	Adam
LR Scheduler	-	-	one cycle	one cycle
Learning Rate	0.1	0.1	0.01	0.01
Weight Decay	0.0001	0.0001	0.01	0.01
Batch Size	128	64	128	128

4.3. Country-specific Removal of Predictions

For each image, the dataset metadata include the country where the image was taken. Additionally, the dataset comes with a list of countries and snake species that live there. We utilize this information to adjust the model predictions to the country of the query as follows: The classifier predictions are set to 0 for all species that do not live in the country of the query. This adjustment is applied only at test time.

4.4. Mixed Precision Training

When training large CNN architectures, fitting the model into limited GPU memory is a bottleneck. We considered the following workarounds: selecting a smaller batch size or applying mixed precision training [21]. Both approaches have an accuracy trade-off.

Mixed precision training is a technique that combines single-precision (32-bit floats, "FP32") and half-precision (16-bit floats, "FP16") float numbers. In order to lower the memory requirements, the forward and backward pass with the large batch size only use a half-precision version of the model. Then, the gradient descent is applied to the single-precision version of the model. In every training step following procedure is applied:

1. Apply the forward pass, compute the loss and apply backward pass on a model in FP16.
2. Convert the gradients from FP16 to FP32.
3. Apply the update on the primary model in FP32.
4. Create a copy of the primary model in FP16.

4.5. Loss Functions

The baseline loss function for training the classifiers is the standard cross entropy loss:

$$\ell_{\text{ce}} = -\sum_{i=1}^{n} \log y_{i,t_i}, \tag{5}$$

where t_i is the ground truth target and $\mathbf{y}_i$ are the classifier predictions for the i-th example, and y_{i,t_i} is the prediction for the ground truth class of the i-th example.

The following subsections describe the loss functions proposed to use the challenge metrics, described in Section 3.3, as a loss measure.

4.5.1. F1 Loss with Soft Assignments

The F1 Score from Equation 2 is not differentiable and thus cannot be utilized as a loss function for back-propagation. We use an approximation of the F1 Score, referenced as soft F1 loss in the rest of this paper, which uses soft assignments that make the function differentiable:

- the true positives for species s are estimated using the softmax predictions $\mathbf{y}$ and one-hot encoded target vector $\mathbf{t}$ as follows: $\widehat{\text{TP}}_s = \sum_{i=1}^{n} \mathbf{y}_i \mathbf{t}_i$
- the false positives for species s are estimated using the softmax predictions $\mathbf{y}$ and one-hot encoded target vector $\mathbf{t}$ as follows: $\widehat{\text{FP}}_s = \sum_{i=1}^{n} \mathbf{y}_i (1 - \mathbf{t}_i)$
- the false negatives for species s are estimated using the softmax predictions $\mathbf{y}$ and one-hot encoded target vector $\mathbf{t}$ as follows: $\widehat{\text{FN}}_s = \sum_{i=1}^{n} (1 - \mathbf{y}_i) \mathbf{t}_i$

Notice, that $\widehat{\text{TP}}$, $\widehat{\text{FP}}$, and $\widehat{\text{FN}}$ are now real valued. Soft F1 Score for species s, $\widehat{F_{1s}}$, is obtained by computing the harmonic mean of precision $\widehat{p}_s$ and recall $\widehat{r}_s$:

$$\widehat{p}_s = \frac{\widehat{\text{TP}}_s}{\widehat{\text{TP}}_s + \widehat{\text{FP}}_s}, \qquad \widehat{r}_s = \frac{\widehat{\text{TP}}_s}{\widehat{\text{TP}}_s + \widehat{\text{FN}}_s}, \qquad \widehat{F_{1s}} = \frac{2\widehat{p}_s\widehat{r}_s}{\widehat{p}_s + \widehat{r}_s}. \tag{6}$$

The macro averaged soft F1 Score is obtained by averaging $\widehat{F_{1s}}$ over all species:

$$\text{macro}(\widehat{F_1}) = \frac{1}{k}\sum_{s=1}^{k}\widehat{F_{1s}}. \tag{7}$$

The final loss function is $\ell_{\widehat{F_1}} = 1 - \text{macro}(\widehat{F_1})$, so that it ranges from 0 (perfect) to 1 (worst).

4.5.2. Weighted Cross Entropy

Because the macro averaged Country F1 Score from Equation 4 increases the importance of species appearing in more countries, we propose a weighted variant of the cross entropy loss with species weights w_s based on the number of countries in which it appears:

$$\ell_{\text{wce}} = -\sum_{i=1}^{n} w_{t_i} \log y_{i,t_i}, \tag{8}$$

The Maximum Likelihood Estimation (MLE) of w_s would simply count the relative frequencies f_s in the provided species-country incidence list. In order to avoid zero weights, we add Laplace smoothing:

$$w_s = \frac{f_s + 1}{\sum\limits_{j=1}^{k}(f_j + 1)}. \tag{9}$$

4.6. Implementation Details

The proposed method was developed using the PyTorch [22] machine learning framework and the fastai framework [23] built on top of PyTorch. The code is available online[2]. All models were fine-tuned from ImageNet-1K [18] pre-trained PyTorch Image Models [24] on one NVIDIA Tesla V100 with 32GB graphic memory.

5. Experiments

5.1. Comparison of Residual Networks

Table 4 shows classification scores of residual networks ResNet, ResNeXt and ResNeSt with 50 and 101 layers. All networks are fine-tuned for 30 epochs on images of size 224×224, minimizing the cross-entropy loss using SGD with momentum. One ResNeSt-101 version is fine-tuned on a larger image size 256×256 to match the image size of the ImageNet pre-trained checkpoint. Both ResNeSt versions, ResNeSt-50 and ResNeSt-101, achieve higher scores compared to the corresponding ResNet and ResNeXt architectures.

[2] https://github.com/chamidullinr/snake-species-identification

Table 4

Classification scores of residual networks fine-tuned for 30 epochs on the mini-subset from Section 3.1. The results are computed on our validation set.

Architecture	Input Size	Accuracy	F1 Score	Country F1 Score
ResNet-50	224	44.0%	0.331	0.300
ResNeXt-50	224	47.2%	0.352	0.333
ResNeSt-50	224	**53.8%**	**0.447**	**0.409**
ResNet-101	224	42.4%	0.290	0.273
ResNeXt-101	224	50.5%	0.428	0.396
ResNeSt-101	224	**56.7%**	**0.475**	**0.432**
ResNeSt-101	256	**58.8%**	**0.500**	**0.455**

5.2. Results of Mixed Precision Training

As observed in the previous section, ResNeSt-101 with a higher input size achieves the highest scores of the experimented residual networks. Since its deeper version, ResNeSt-200, does not fit into our GPU memory with larger batch sizes, we experiment with the mixed precision training from Section 4.4.

Table 5 compares the training time and accuracy of ResNeSt-101 and ResNeSt-200 when training with and without the mixed precision technique. Note that in our computational environment, mixed precision runs slower than single precision. The prediction scores after 10 epochs show that mixed precision has little impact on prediction accuracy in setups with the same architecture and batch size. Increasing the batch size from 32 to 64 has a much larger impact on the accuracy. Thus the network trained on a larger batch size with mixed precision achieves better scores than the single-precision network with a smaller batch size.

Table 5

Classification scores and training times when fine-tuning for 10 epochs with and without the mixed precision technique. Cells with "×" denote setups for which the network did not fit into the 32GB GPU memory. The networks are fine-tuned on the mini-subset from Section 3.1 and the results are computed on our validation set.

Architecture	BS	Precision type	Accuracy	F1 Score	Country F1 Score	Epoch time
ResNeSt-101	128	Mixed	48.0%	0.387	0.355	14 min
ResNeSt-101	128	Single	47.6%	0.385	0.348	10 min
ResNeSt-200	128	Mixed	×	×	×	×
ResNeSt-200	128	Single	×	×	×	×
ResNeSt-200	64	Mixed	52.7%	0.424	0.398	40 min
ResNeSt-200	64	Single	×	×	×	×
ResNeSt-200	32	Mixed	46.9%	0.376	0.345	41 min
ResNeSt-200	32	Single	46.9%	0.371	0.345	28 min

Table 6
Classification scores on ResNeSt-101 with different loss functions. Standard cross entropy achieves superior results. The networks are fine-tuned on the mini-subset from Section 3.1 and the results are computed on our validation set.

Loss Function	Accuracy	F1 Score	Country F1 Score
Cross Entropy Loss	**58.8%**	**0.500**	**0.455**
Weighted Cross Entropy Loss	48.4%	0.349	0.385
F1 Loss	0.2%	0.001	0.000

5.3. Evaluation of Different Loss Functions

The loss functions introduced in Section 4.5, namely the soft F1 loss and the weighted cross entropy loss, resulted in inferior classification scores compared to cross entropy loss, see Table 6. We, therefore, fine-tune the CNN classifiers with cross entropy loss, and then choose the best training checkpoint based on the highest validation Country F1 Score.

One possible explanation for the failure of the soft F1 loss is that the batch size of 64 is significantly smaller than the total number of classes, 772. This leads to the classes not being represented in every mini-batch, making the approximation of the F1 loss inaccurate. Figure 2 illustrates the inaccurate approximation of the F1 loss on an example, where the loss values are mostly 0s or 1s.

5.4. Evaluation of Country-specific Removal of Predictions

We measure the prediction scores of ResNeSt-200 with and without the removal of species predictions based on the country incidence information. Table 7 compares the prediction scores on our validation set. The improvement is 0.150 in F1 Score and 0.193 in Country F1 Score.

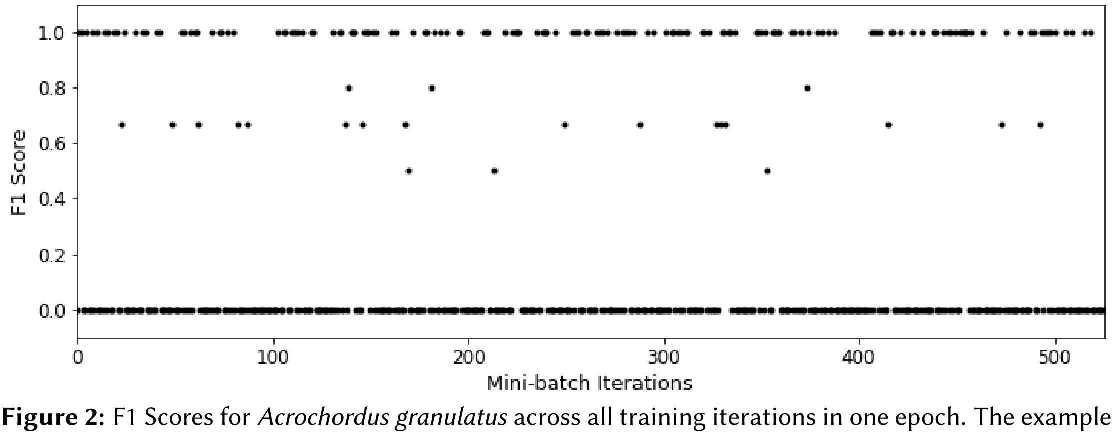

Figure 2: F1 Scores for *Acrochordus granulatus* across all training iterations in one epoch. The example illustrates the inaccurate approximation of the F1 loss: the loss rarely takes values other than 0 and 1.

Table 7
Comparing classification scores of ResNeSt-200 with and without the removal of species predictions based on the country incidence information. The networks are fine-tuned on the mini-subset from Section 3.1 and the results are computed on our validation set.

Country-specific removal of predictions	Accuracy	F1 Score	Country F1 Score
No	74.8%	0.483	0.504
Yes	**79.0%**	**0.633**	**0.697**

5.5. Challenge Submissions

We submitted the following five runs to the SnakeCLEF 2021 challenge:

CMP_S1: ResNeSt-200 fine-tuned for 20 epochs on the full dataset with SGD.

CMP_S2: ResNeSt-200 from CMP_S1 fine-tuned for additional 10 epochs on the full dataset with SGD.

CMP_S3: ResNet-101 fine-tuned for 25 epochs on the full dataset with Adam and one cycle.

CMP_S4: ResNeXt-101 fine-tuned for 30 epochs on the mini-subset from Section 3.1 with Adam and one cycle.

CMP_S5: An ensemble of all four previous runs, combining the top one predictions by majority voting strategy. In case of ties, predictions of CMP_S1 are preferred.

Table 8 shows the final challenge scores on the test set. While different in accuracy, the CNN architectures ResNeSt-200, ResNeXt-101 and ResNet-101 achieve similar results in the primary challenge metric, the Country F1 Score. The highest scores are achieved by the ensemble.

We recognize a shortcoming of the ensemble submission (CMP_S5), which inclines towards the ResNeSt-200 submissions related to each other (CMP_S2 is fine-tuned from CMP_S1). The remaining networks cannot outvote an agreement of CMP_S1 and CMP_S2.

Table 8
Classification scores of the submitted challenge runs on the SnakeCLEF 2021 challenge test set. The networks are fine-tuned either on the full dataset (Full) or on the mini-subset (Mini) from Section 3.1. The CNN architectures ResNeSt-200, ResNeXt-101 and ResNet-101 achieve similar results in the Country F1 Score. The highest scores are achieved by the ensemble of all networks.

Submission	Architecture	Dataset	Accuracy	F1 Score	Country F1 Score
CMP_S1	ResNeSt-200	Full	90.6%	0.772	**0.839**
CMP_S2	ResNeSt-200	Full	89.5%	0.779	0.819
CMP_S3	ResNet-101	Full	90.7%	0.795	0.837
CMP_S4	ResNeXt-101	Mini	77.6%	**0.796**	**0.839**
CMP_S5	Ensemble of CMP_S1-S4	-	**91.6%**	**0.830**	**0.860**

6. Conclusions

The paper presents a deep learning method for image-based snake species identification, a fine-grained classification problem with a long tail class distribution. The method is based on deep residual neural networks – ResNeSt, ResNeXt and ResNet – fine-tuned from ImageNet pre-trained checkpoints. We achieve performance improvements by: discarding predictions of species that do not occur in the country of the query; combining predictions from an ensemble of classifiers; and applying mixed precision training, which allows training neural networks with larger batch size.

The experimented soft F1 loss and weighted cross entropy loss produced inferior results compared to the standard cross entropy minimization. Thus, the competition submissions are fine-tuned with the standard cross entropy loss.

The proposed method scored third in the SnakeCLEF 2021 challenge, achieving 91.6% classification accuracy, Country F1 Score of 0.860, and F1 Score of 0.830.

Acknowledgments

This research was supported by the OP VVV funded project CZ.02.1.01/0.0/0.0/16_019/0000765. LP was supported by the UWB grant, project No. SGS-2019-027.

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 CLEF 2021 - Conference and Labs of the Evaluation Forum, 2021.

[2] A. Joly, H. Goëau, S. Kahl, L. Picek, T. Lorieul, E. Cole, B. Deneu, M. Servajean, R. Ruiz De Castañeda, G. H. Bolon, Isabelle, 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 Twelfth International Conference of the CLEF Association (CLEF 2021), 2021.

[3] I. Bolon, A. M. Durso, S. Botero Mesa, N. Ray, G. Alcoba, F. Chappuis, R. Ruiz de Castañeda, Identifying the snake: First scoping review on practices of communities and healthcare providers confronted with snakebite across the world, PLOS ONE 15 (2020). URL: https://doi.org/10.1371/journal.pone.0229989. doi:10.1371/journal.pone.0229989.

[4] C. Abeysinghe, A. Welivita, I. Perera, Snake Image Classification Using Siamese Networks, in: Proceedings of the 2019 3rd International Conference on Graphics and Signal Processing, 2019. URL: https://doi.org/10.1145/3338472.3338476.

[5] G. Koch, R. Zemel, R. Salakhutdinov, Siamese Neural Networks for One-Shot Image Recognition, in: ICML Deep Learning Workshop, 2015.

[6] L. Picek, I. Bolon, A. M. Durso, R. Ruiz De Castañeda, Overview of the snakeclef 2020: Automatic snake species identification challenge, in: CLEF task overview 2020, CLEF: Conference and Labs of the Evaluation Forum, 2020.

[7] G. K. Moorthy, Impact of Pretrained Networks For Snake Species Classification, in: CLEF working notes 2020, CLEF: Conference and Labs of the Evaluation Forum, 2020.

[8] K. He, X. Zhang, S. Ren, J. Sun, Identity Mappings in Deep Residual Networks, in: Computer Vision – ECCV 2016, 2016.

[9] T. Ridnik, E. Ben-Baruch, A. Noy, L. Zelnik-Manor, ImageNet-21K Pretraining for the Masses, 2021. arXiv:2104.10972.

[10] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond Empirical Risk Minimization, in: International Conference on Learning Representations, 2018. URL: https://openreview.net/forum?id=r1Ddp1-Rb.

[11] Y. Wu, K. He, Group Normalization, in: Computer Vision – ECCV 2018, 2018.

[12] L. Bloch, A. Boketta, C. Keibel, E. Mense, A. 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: CLEF working notes 2020, CLEF: Conference and Labs of the Evaluation Forum, 2020.

[13] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017. doi:10.1109/ICCV.2017.322.

[14] M. Tan, Q. Le, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, in: Proceedings of the 36th International Conference on Machine Learning, 2019. URL: http://proceedings.mlr.press/v97/tan19a.html.

[15] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[16] S. Xie, R. Girshick, P. Dollar, Z. Tu, K. He, Aggregated Residual Transformations for Deep Neural Networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[17] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Mueller, R. Manmatha, M. Li, A. Smola, ResNeSt: Split-Attention Networks, 2020. arXiv:2004.08955.

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[19] D. P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, CoRR abs/1412.6980 (2015).

[20] L. N. Smith, N. Topin, Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates, 2018. arXiv:1708.07120.

[21] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, H. Wu, Mixed Precision Training, in: International Conference on Learning Representations, 2018. URL: https://openreview.net/forum?id=r1gs9JgRZ.

[22] 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 32, Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[23] J. Howard, S. Gugger, Fastai: A Layered API for Deep Learning, Information 11 (2020) 108. URL: http://dx.doi.org/10.3390/info11020108. doi:10.3390/info11020108.

[24] R. Wightman, PyTorch Image Models, https://github.com/rwightman/pytorch-image-models, 2019. doi:10.5281/zenodo.4414861, visited on 2021-06-28.