When Large Kernel Meets Vision Transformer: A Solution for SnakeCLEF & FungiCLEF

Yang Shen[1], Xuhao Sun[1] and Zijian Zhu[1]

[1]*School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China*

Abstract

LifeCLEF 2022 is an evaluation campaign that is being organized as part of the CLEF initiative labs. This paper record solutions of two competitions in LifeCLEF 2022, *i.e.*, SnakeCLEF 2022 and FungiCLEF 2022. The SnakeCLEF aims at building an automatic and robust image-based system for snake species identification while FungiCLEF aims at automatic recognize fungi species with both images and rich metadata. These two competitions contain a number of challenges, such as fine-grained image recognition, long-tailed recognition and openset recognition. In this paper, we utilize existing efficient techniques and tricks to deal with the long-tailed challenge in both SnakeCLEF and FungiCLEF. We also propose a new backbone by combining the large kernel convolution and vision transformer as both of them have shown superior performance in recognition tasks. For the SnakeCLEF competition, our team achieves a 85.4% Macro F1-Score on the private leaderboard while for the FungiCLEF competition, we achieve a 78.9% Macro F1-Score. Codes are available at: https://github.com/sinbais/CLEF2022.

Keywords

FungiCLEF, SnakeCLEF, Openset recognition, Fine-grained image recognition, Long-tailed recognition

1. Introduction

Building accurate knowledge of the identity and the evolution of species is essential for the sustainable development of humanity, as well as for biodiversity conservation. However, the difficulty of identifying animals and fungi is hindering the aggregation of new data and knowledge. Identifying and naming living organisms is almost impossible for the general public and is often difficult even for professionals and naturalists [1]. The LifeCLEF Lab has been promoting and evaluating advances in this domain for over 10 years and has has achieved a lot of meaningful results [2, 3, 4].

In this paper, we combine existing effective techniques for the state-of-the-art pre-trained models and utilize advanced methods in the long-tailed recognition task to give solutions for both competitions (cf. Section 4). We also design a new backbone by combining two recent hot topics, *i.e.*, large kernel convolution and vision transformers (cf. Section 3.1). Our overall strategy was to test as many models as possible and spend less time on fine-tuning. The goal was to have many diverse models for ensembling rather than some highly tuned ones. In the ensemble stage, our strategy was to choose the best combination that we thought can avoid overfitting and spend the rest of the time fitting the public leaderboard. In the following of this section, we introduce these two competitions once more.

CLEF 2022: Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy

shenyang_98@njust.edu.cn (Y. Shen); sunxh@njust.edu.cn (X. Sun); zhuzijian@njust.edu.cn (Z. Zhu)

The SnakeCLEF competition aims at building an automatic and robust image-based system for snake species identification, which is an important goal for biodiversity, conservation, and global health. With recent estimates of 81,410–137,880 deaths and up to three times as many victims of amputations, permanent disability and disfigurement (globally each year) caused by venomous snakebite, such a system has the potential to improve eco-epidemiological data and treatment outcomes (e.g. based on the specific use of antivenoms). This applies especially in remote geographic areas and developing countries, where automatic snake species identification has the greatest potential to save lives. [5, 1, 6].

The FungiCLEF competition focuses on recognize fungi in the open world. Automatic recognition of fungi species assists mycologists, citizen scientists and nature enthusiasts in species identification in the wild. Its availability supports the collection of valuable biodiversity data. In practice, species identification typically does not depend solely on the visual observation of the specimen but also on other information available to the observer — such as habitat, substrate, location and time. Thanks to rich metadata, precise annotations, and baselines available to all competitors, the challenge provides a benchmark for image recognition with the use of additional information. Moreover, the toxicity of a mushroom can be crucial for the decision of a mushroom picker [7, 1, 6].

2. Datasets and Evaluation Protocol

2.1. Dataset for SnakeCLEF

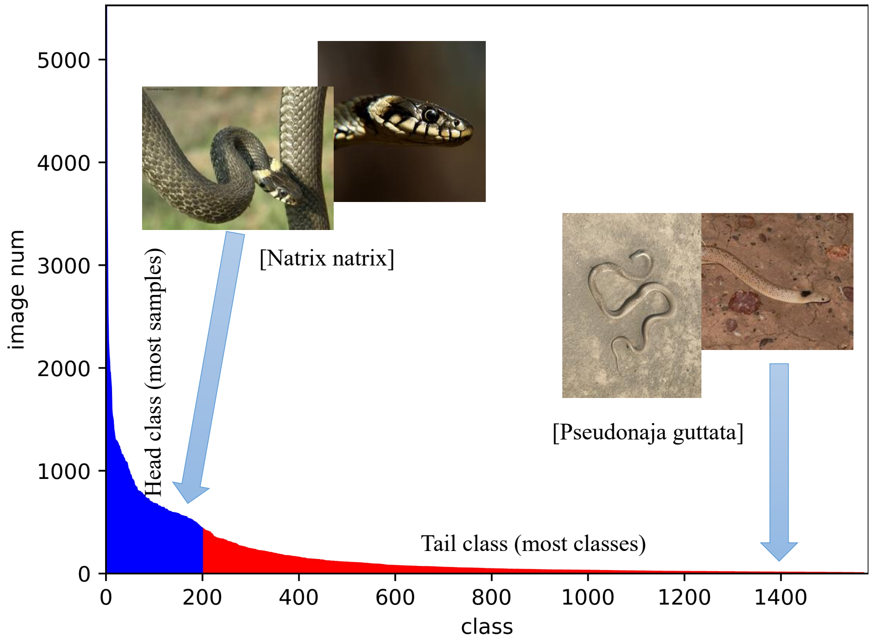

Figure 1: Class distribution of the training set of SnakeCLEF.

For this year challenge, organizers prepared a dataset based on 187,129 snake observations with 318,532 photographs belonging to 1,572 snake species and observed in 208 countries. The data were gathered from the online biodiversity platform – iNaturalist.[1]

The provided dataset has a heavy long-tailed class distribution, where the most frequent species is represented by 6,472 images and the least frequent species by just 5 samples [5].

2.2. Dataset for FungiCLEF

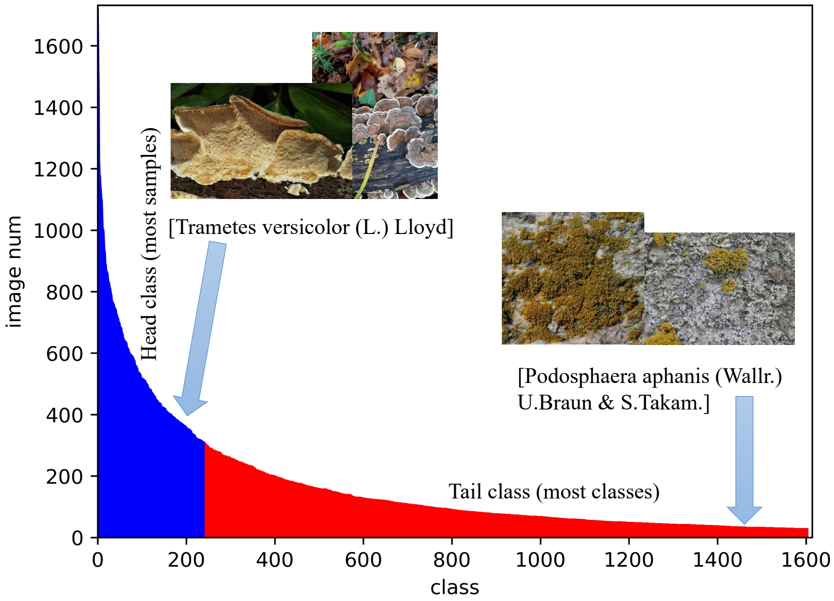

Figure 2: Class distribution of the training set of FungiCLEF.

The FungiCLEF challenge dataset is based on the data from the Danish Fungi 2020 dataset [8], which contains 295,938 training images belonging to 1,604 species observed mostly in Denmark. All training samples passed an expert validation process, guaranteeing high-quality labels. Rich observation metadata about habitat, substrate, time, location, EXIF are provided.

The test set contains 59,420 observations with 118,676 images and 3,134 species, covering the whole year and includes observations collected across all substrate and habitat types [7].

2.3. Evaluation Protocol

The evaluation metric for this competition is Mean (Macro) F1-Score:

$$\mathrm{Macro\ F_1} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i=i}}^{\mathrm{N}} \mathrm{F_{1_i}}\,, \tag{1}$$

where i is the species index and N is the number of classes/species. The F1 score, commonly used in information retrieval, measures accuracy using the statistics precision (P) and recall (R).

[1] www.inaturalist.org

The macro F1 score is not biased by class frequencies and is more suitable for the long-tailed class distributions observed in nature. Precision is the ratio of true positives (TP) to all predicted positives (TP + FP). The Recall is the ratio of true positives (TP) to all actual positives (TP + FN). The F1 metric weights recall and precision equally, and a good retrieval algorithm will maximize both precision and recall simultaneously. Thus, moderately good performance on both will be favoured over extremely good performance on one and poor performance on the other.

3. Methods

3.1. Proposed CoLKANet

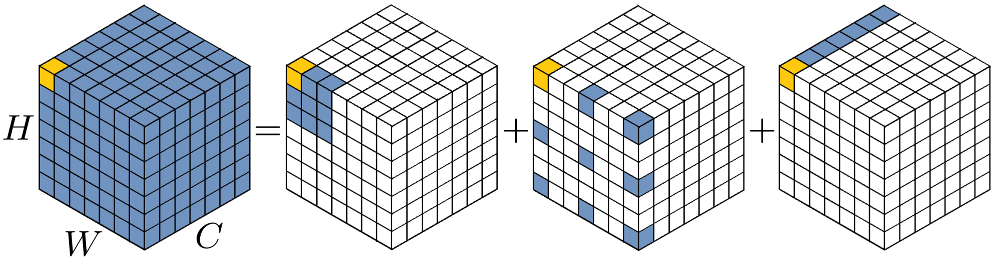

Figure 3: Decomposition diagram of large-kernel convolution [9]. H, W, C represents for the height, weight and channel of a tensor. A standard convolution can be decomposed into three parts: a depth-wise convolution (DW-Conv), a depth-wise dilation convolution (DW-D-Conv) and a 1×1 convolution (1×1 Conv). The colored grids represent the location of convolution kernel and the yellow grid means the center point. The diagram shows that a 13×13 convolution is decomposed into a 5×5 depth-wise convolution, a 5×5 depth-wise dilation convolution with dilation rate 3 and 1×1 convolution. Note: zero paddings are omitted in above figure.

We first introduce the CoLKANet. Actually, it is a combination of large kernel attention (cf. Fig. 3) and vision transformer. Since the breakthrough of AlexNet [10] and ResNet [11] Convolutional Neural Networks (CNNs) have been the dominating model architecture for computer vision. Meanwhile, with the success of self-attention models in natural language processing, many previous works have attempted to bring in the power of attention into computer vision. When pre-trained on large-scale weakly labeled JFT-300M dataset, ViT can achieve comparable results to state-of-the-art CNNs. In this year, researchers revisited large kernel design in CNNs and found that using a few large convolutional kernels instead of a stack of small kernels could be a more powerful paradigm [12]. From previous research, we can see that earlier convolution helps transformer see better. Therefore, following this idea, we combine large kernel attention with vision transformer. We use the structure in VAN [9] and by following CoAtnet [13], we replace those earlier CNN structures. Overall structure of the CoLKANet can refer to Fig. 4.

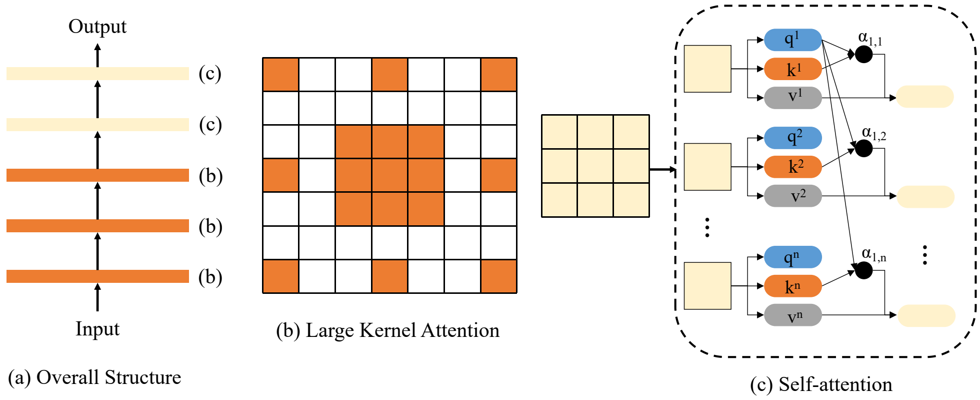

Figure 4: The proposed structure of the CoLKANet.

3.2. Other Classification Models

For other classification models, we have tried EfficientNet [14], RepVGG [15], EfficientNet-V2 [16], Swin Transformer [17], VOLO [18], ViTAE [19], ViT [20] and ConvNeXt [21]. We choose the combination of Swin Transformer, VOLO, ConvNeXt, ViT and our CoLKANet as the final solution. We found that CNNs are more likely to overfit in this competition (except ConvNeXt) while Transformer seems to be more stable. However, when raising the input resolution of ConvNeXt and EfficientNet, it will gain a substantial improvement.

4. Details and Tricks

In this section, we report the details for different backbones and describe all the tricks that we have used to generate the final submission. We also report the scores we have recorded on both public leaderboard and final leaderboard.

4.1. Details for Vision Transformers

For different competitions, we use different settings for vision transformers.

SnakeCLEF. We use ViT, Swin Transformer, VOLO and the proposed CoLKANet as backbones. We calculate the 5-fold cross validation accuracy while training with the image resolution of 384×384 for Swin Transformer and 448×448 for VOLO. We trained the ViT and CoLKANet by using the whole dataset with the image resolution of 384×384. We choose 1.2×10^4 as the initial learning rate while $10^{-5}/7$ as the minimum learning rate and set weight decay as 2×10^{-5}. A good technique to reduce overfitting is to stop the model from becoming overconfident. This can be achieved by softening the ground truth using Label Smoothing [22]. We set the Label Smoothing value as 0.1 according to the original paper [22]. All these backbone are trained for 15 epochs without warmup. AdamW [23] optimizer is utilized for training.

FungiCLEF. We use Swin Transformer, VOLO and the proposed CoLKANet as backbones. We calculate the 5-fold cross validation accuracy while training with the image resolution of 384×384 for Swin Transformer while 448×448 for VOLO. We trained the CoLKANet by using the whole dataset with the image resolution of 384×384. We choose 1.2×10^4 as the initial learning rate while $10^{-5}/7$ as the minimum learning rate and set weight decay as 2×10^{-5}. Label Smoothing is set as 0.1. All these backbone are trained for 15 epochs without warmup. AdamW [23] optimizer is utilized for training.

4.2. Details for CNNs

We only use ConvNeXt as the CNN backbone in the final submission for both SnakeCLEF and FungiCLEF. We calculate the 5-fold cross validation points while training with the resolution of 384×384. We also train a single model which use the whole dataset with the image resolution of 448×448. We choose 1.2×10^4 as the maximum learning rate while $10^{-5}/7$ as the minimum learning rate and set weight decay as 2×10^{-5}. Label Smoothing is set as 0.1. We apply warmup and gradually increase the learning rate for 3 epochs. Then, another optimization is to apply Cosine Schedule to adjust our LR during the following 15 epochs. AdamW [23] optimizer is utilized for training.

4.3. Loss Functions

As the evaluation metric is Macro F1-Score, we have to deal with the long-tailed problem (If the evaluation metric is Micro F1-Score, adpot the original Cross Entropy Loss is enough.). Because the head classes contain much more training examples, the network makes the weight norm of the head classes larger to approach the optimal solution. It results in predicted probabilities mainly near 1.0. Another fact is that distributions of predicted probability are related to instance numbers. Unlike balanced recognition, applying different strategies for these classes is necessary for solving the long-tailed problem. Therefore, we adopt label-aware smoothing [24] to solve the over-confidence in cross-entropy and varying distributions of predicted probability issues for both SnakeCLEF and FungiCLEF. It is expressed as:

$$l(\boldsymbol{q}, \boldsymbol{p}) = -\sum_{i=1}^{K} \boldsymbol{q}_i log \boldsymbol{p}_i \, , \boldsymbol{q}_i = \begin{cases} 1 - \epsilon_y = 1 - f(N_y), & i = y \, , \\ \frac{\epsilon_y}{K-1} = \frac{f(N_y)}{K-1}, & \text{otherwise} \, , \end{cases} \tag{2}$$

where ϵ_y is a small label smoothing factor for Class-y, relating to its class number N_y. Details about label-aware smoothing can refer the original paper [24].

It also worth noting that if we increase the weight of some tail classes and head classes, we will get a higher score on the public leaderboard (but useless in private leaderboard). We explain this phenomenon in Section 6.1.

4.4. Important Tricks

Bellow are 8 important tricks we used during both SnakeCLEF and FungiCLEF.

• Augmentation: Augmentation is very import during training stage. We tried combination of RandomResizedCrop, Transpose, HorizontalFlip, VerticalFlip, ShiftScaleRotate, IAAPiecewiseAffine, HueSaturationValue, RandomBrightnessContrast, OpticalDistortion, GridDistortion, ElasticTransform, Cutout and CoarseDropout designed in Albumentations [25] at the begining of the competitions. During the middle stage, we replaced them with TrivialAugment [26] and gain around 0.5% improvement for each single model. We also used Random Erasing [27], CutMix [28] and Mixup [29] throughout the competitions. They provide strong regularization effects by softening not only the labels but also the images.

• Confusion Matrix: A typical way to analyze model performance is using the confusion matrix. As we split the training images into 5 equal parts (*i.e.*, 5-fold), we get the confusion matrix from the valid part when training those 5-fold models (We did not use confusion matrix for models trained with the whole dataset). This trick gain around 0.2% improvement when ensemble different models.

• Normalization of Output: Besides confusion matrix, we hope to vote for each model (*e.g.*, Models A, B and C predict that the image belongs to category 1, 2, 1 respectively. Finally we set the image as category 1.) but it performs poor. So we temporarily set another trick, *i.e.*, scale the model output and normalize the maximum value to α, α can not be one for it may be too close to the voting strategy, formula is as follows:

$$Norm(f(x)) = (1/max(f(x)))^{\alpha} f(x)\,. \tag{3}$$

It is found that models performs well when α is 0.15 or 0.20. This trick gain around 0.1% improvement when ensemble different models.

• Test Time Augementation: It is an very important trick during all the competitions. However, it may lead to overfitting. Performing this trick is very easy: just crop the test images for around 8 to 13 times and calculate the average score. It gains around 0.6% improvement for each single model but unfortunately, it did not work on the private leaderboard in SnakeCLEF.

• Pseudo Labelling: We only perform pseudo labelling for SnakeCLEF. This trick did not work in FungiCLEF (We only tried it once and it did not work on public leaderboard. This may be caused by the openset problem.). We generate pseudo labels on test dataset only for tail categories (train data less than 100) by clustering methods, and finetune the model on both training and these test data. It gains around 0.9% improvement on the public leaderboard but useless on the private leaderboard.

• Weight Decay Tuning: Our standard recipe uses $\ell 2$ regularization to reduce overfitting. The Weight Decay parameter controls the degree of the regularization (the larger the stronger) and is applied universally to all learned parameters of the model by default [30]. More about separating the Normalization parameters from the rest can refer ClassyVision [31].

• Exponential Moving Average (EMA): EMA [32] is a technique that allows one to push the accuracy of a model without increasing its complexity or inference time. It performs an exponential moving average on the model weights and this leads to increased accuracy and more stable models. The averaging happens every few iterations and its decay parameter was tuned via grid search.

• FixRes mitigations: It is a very important trick in CNNs (Transformers fix image size so it can not be used) and we only use this trick in ConvNeXt. The model performed significantly better if the resolution used during validation was increased from the training size. This effect is studied in detail on the FixRes paper [33] and two mitigations are proposed: a) one could try to reduce the training resolution so that the accuracy on the validation resolution is maximized or b) one could fine-tune the model on a two-phase training so that it adjusts on the target resolution. Another very important phenomenon is that if we improve the resolution of training images, we will easily gain a better score. The image scaling ratio is set as 0.758 and 0.875 according to the experiments in the original paper [33] and [30] . It gains around 0.8% improvement in SnakeCLEF but only 0.2% improvement in FungiCLEF.

5. Main Results

5.1. Generating the Final Submission

It is very easy for us to generate the final csv file. Specifically, we generate only one prediction for each observation. If one observation has several different test images, we use the model to calculate the predicted value of each image and then calculate the average value for that observation.

Table 1

Results on SnakeCLEF.

Backbone	Train Resolution	Score	Comments
Swin baseline	384× 384	75.6%	Without any tricks.
ViT	384× 384	79.8%	Pretained model from MAE[34].
Swin single model	384× 384	77.4±0.5%	5-fold single model.
Swin 5-fold ensemble	384× 384	80.7%	All the folds trained with LabelAwaerSmoothing and CrossEntropy.
VOLO single model	448× 448	78.7±0.7%	5-fold single model.
VOLO 5-fold + Swin 5-fold	-	82.6%	All the VOLO models are trained with LabelAwaerSmoothing.
CoLKANet	384× 384	80.1%	Train with LabelAwaerSmoothing.
ConvNeXt single model	384× 384	77.9±0.8%	5-fold single model.
ConvNeXt 5-fold ensemble	384× 384	81.4%	0,2,4 fold trained with LabelAwaerSmoothing 1,3 fold trained with CrossEntropy.
ConvNeXt+CoLKANet	-	83.9%	ConvNeXt 448 without FixRes + 5-fold CoLKANet 384 + CoLKANet Weight ratio is 2:2:1.
Ensemble	-	85.4%	ConvNeXt 448 + 5-fold ConvNeXt 384 + 5-fold VOLO + CoLKANet + 5-fold Swin + ViT with all the tricks in Section 4.4.

5.2. SnakeCLEF

In this section, we report the scores we had recorded in SnakeCLEF. Details can be found in Table 1. Except ViT and VOLO, all the pretrained models are from ImageNet-22k [35]. Each backbone can be found in timm [36]. Swin refers to swin_large_patch4_window12_384_in22k, ConvNeXt refers to convnext_large_in22k, VOLO refers to volo_d4_448.

5.3. FungiCLEF

In this section, we report the scores we had recorded in FungiCLEF. Details can be found in Table 2. Except VOLO, all the pretrained models are from ImageNet-22k [35]. It worth mention that in this competition, combine models train with LabelAwaerSmoothing (*i.e.*, deal with the long-tailed problem) and CrossEntropy (*i.e.*, do not deal with the long-tailed problem) will get around 1.6% improvement on each single model.

Table 2

Results on FungiCLEF.

Backbone	Train Resolution	Score	Comments
Swin baseline	384× 384	71.7%	Without any tricks.
Swin single model	384× 384	73.4±0.5%	5-fold single model.
Swin 5-fold ensemble	384× 384	76.5%	All the folds trained with LabelAwaerSmoothing and CrossEntropy.
VOLO single model	448× 448	73.8±0.8%	5-fold single model.
VOLO 5-fold ensemble	448× 448	76.2%	0,2,4 fold trained with LabelAwaerSmoothing 1,3 fold trained with CrossEntropy
CoLKANet	384× 384	75.1%	Train with LabelAwaerSmoothing.
ConvNeXt single model	384× 384	73.6±0.5%	5-fold single model.
ConvNeXt 5-fold ensemble	384× 384	76.1%	All the folds trained with LabelAwaerSmoothing and CrossEntropy.
ConvNeXt	448× 448	75.7%	Train with LabelAwaerSmoothing.
Ensemble	-	78.9%	ConvNeXt 448 + 5-fold ConvNeXt 384 + 5-fold VOLO + CoLKANet + 5-fold Swin with all the tricks in Section 4.4.

6. Discussions

6.1. Gap Between the Public Leaderboard and Private Leaderboard.

First of all, at the begining of these two competitions, the most important thing is to guess the data distribution of the test dataset. Based on our experience, we guessed that the data distribution for the test dataset should be roughly the same as the provided train/val dataset, and the scores for the 20% data which showed on the public leaderboard should also fit the

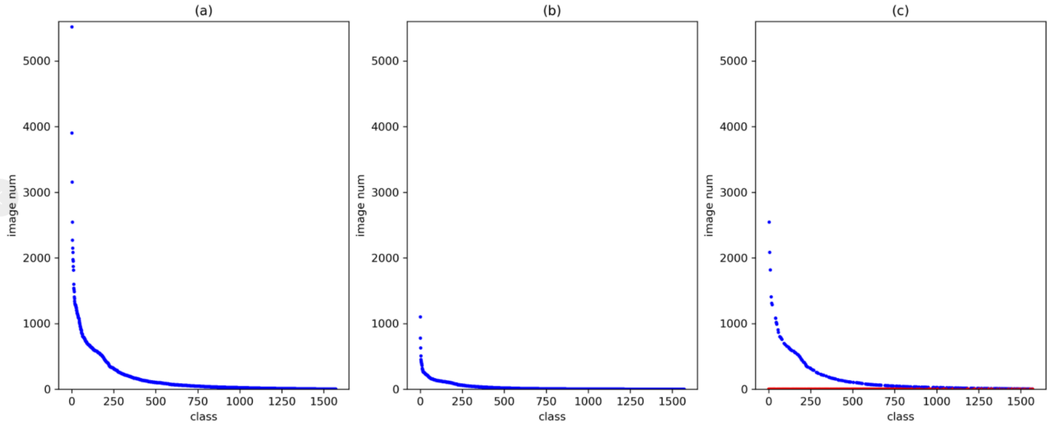

Figure 5: Data distribution of SnakeCLEF. (a) A smaple distribution of the original training data. (b) Our guess about the data distribution of the test dataset on the public leaderborad. (c) The actual distribution of the test dataset on the public leaderboard. The blue dots represent the number of samples contained in a category while the red dots indicate that the number of samples is 0.

distribution of the test dataset (cf. Fig. 5 (b)). However, during these two competitions, we found if we artificially increase the weight of some categories (especially on SnakeCLEF which did not suffer from OSR), we would get a better score on the public leaderboard. These categories were chosen at random and the best random state gain around 1% improvement on the public leaderborad. At the outset, we thought it a mismatch in the number of some tail categories between the training dataset and the test dataset.

While towards the end of these two competitions, we made a rather important discovery. In the FungiCLEF competition, if we submit a csv with all labels set as -1, we can get a score of about 0.0005 on the public leaderboard. As the FungiCLEF competition contains 1,604 categories. We derived this score backwards by using the formula for Macro-F1 and got a result that the whole dataset should contain about 60% to 70% openset images (*i.e.*, label$= -1$).

Give an example, assuming a total of 59,420 results are to be submitted, then 34,000 of these results have a ground truth of -1 and the other 25,000 samples between 0 and 1,603, a document with all -1's has a macro F1 value of approximately 0.00045. We added this threshold to our earlier submissions and, not surprisingly, the score dropped dramatically. Obviously this is an unreasonable result. Therefore, we had a wild guess: 20% of the data in the public leaderboard, which means, randomly select 20% of these 1,604 categories in FungiCLEF to calculate the score (cf. Fig. 5 (c)). The same goes for the SnakeCLEF.

In this setting, we calculate the openset images in the whole dataset should be around 10%. However, the overall strategy: fitting the public leaderboard, seems overfitted. The fungi competition, on the other hand, did not have a significant reduction on the private leaderboard after adjusting the appropriate threshold, as the -1 category had a far greater effect than the other categories.

6.2. Only A Threshold Strategy is Sufficient for the Openset Problem.

The ability to identify whether or not a test sample belongs to one of the semantic classes in a classifier's training set is critical to practical deployment of the model. Sagar Vaze et al. [37] demonstrated that the ability of a classifier to make the 'none-of-above' decision is highly correlated with its accuracy on the closed-set classes. They also use this correlation to boost the performance of the maximum softmax probability OSR 'baseline' by improving its closed-set accuracy and with this strong baseline achieve state-of-the-art on a number of OSR benchmarks. Therefore, we only use a simple threshold for the FungiCLEF competition. We also tried post-processing with meta data (by using MetaFormer [38]) but useless in this competition.

7. Conclusion

Fine-grained image recognition is an important problem in computer vision. Combined with the long-tailed problem and the openset problem, the SnakeCLEF and the FungiCLEF become more challenging. In this paper, we report the advanced techniques we had used to deal with these challenges. By combining the recent hot topics in computer vision tasks, *i.e.*, large kernel and vision transformer, we also construct a new model named CoLKANet. For the SnakeCLEF competition, our team achieves a 85.4% Macro F1-Score on the private leaderboard. For the FungiCLEF competition, our team achieves a 78.9% Macro F1-Score on the private leaderboard.

References

[1] A. Joly, H. Goëau, S. Kahl, L. Picek, T. Lorieul, E. Cole, B. Deneu, M. Servajean, A. Durso, I. Bolon, et al., LifeCLEF 2022 teaser: An evaluation of machine-learning based species identification and species distribution prediction, in: European Conference on Information Retrieval, Springer, 2022, pp. 390–399.

[2] A. Joly, H. Goëau, H. Glotin, C. Spampinato, P. Bonnet, W.-P. Vellinga, J.-C. Lombardo, R. Planqué, S. Palazzo, H. Müller, Lifeclef 2017 lab overview: multimedia species identification challenges, in: International conference of the cross-language evaluation forum for European languages, Springer, 2017, pp. 255–274.

[3] D. Casanova, J. B. Florindo, W. N. Gonçalves, O. M. Bruno, Ifsc/usp at imageclef 2012: Plant identification task., in: CLEF (Online Working Notes/Labs/Workshop), 2012.

[4] H. Goëau, P. Bonnet, A. Joly, Plant identification in an open-world (lifeclef 2016), in: CLEF: Conference and Labs of the Evaluation Forum, 1609, 2016, pp. 428–439.

[5] L. Picek, A. M. Durso, M. Hrúz, I. Bolon, Overview of SnakeCLEF 2022: Automated snake species identification on a global scale, in: Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, 2022.

[6] A. Joly, H. Goëau, S. Kahl, L. Picek, T. Lorieul, E. Cole, B. Deneu, M. Servajean, A. Durso, H. Glotin, R. Planqué, W.-P. Vellinga, A. Navine, H. Klinck, T. Denton, I. Eggel, P. Bonnet, M. Šulc, M. Hruz, Overview of LifeCLEF 2022: an evaluation of machine-learning based

species identification and species distribution prediction, in: International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, 2022.

[7] L. Picek, M. Šulc, J. Heilmann-Clausen, J. Matas, Overview of FungiCLEF 2022: Fungi recognition as an open set classification problem, in: Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, 2022.

[8] L. Picek, M. Šulc, J. Matas, T. S. Jeppesen, J. Heilmann-Clausen, T. Læssøe, T. Frøslev, Danish fungi 2020-not just another image recognition dataset, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 1525–1535.

[9] M.-H. Guo, C.-Z. Lu, Z.-N. Liu, M.-M. Cheng, S.-M. Hu, Visual attention network, arXiv preprint arXiv:2202.09741 (2022).

[10] A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks, Advances in neural information processing systems 25 (2012).

[11] 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, 2016, pp. 770–778.

[12] X. Ding, X. Zhang, Y. Zhou, J. Han, G. Ding, J. Sun, Scaling up your kernels to 31x31: Revisiting large kernel design in CNNs, arXiv preprint arXiv:2203.06717 (2022).

[13] Z. Dai, H. Liu, Q. V. Le, M. Tan, Coatnet: Marrying convolution and attention for all data sizes, Advances in Neural Information Processing Systems 34 (2021) 3965–3977.

[14] M. Tan, Q. Le, EfficientNet: Rethinking model scaling for convolutional neural networks, in: International Conference on Machine Learning, PMLR, 2019, pp. 6105–6114.

[15] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, J. Sun, RepvVGG: Making VGG-style convnets great again, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 13733–13742.

[16] M. Tan, Q. V. Le, EfficientNetv2: Smaller models and faster training, arXiv preprint arXiv:2104.00298 (2021).

[17] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, arXiv preprint arXiv:2103.14030 (2021).

[18] L. Yuan, Q. Hou, Z. Jiang, J. Feng, S. Yan, VOLO: Vision outlooker for visual recognition, arXiv preprint arXiv:2106.13112 (2021).

[19] Y. Xu, Q. Zhang, J. Zhang, D. Tao, ViTAE: Vision transformer advanced by exploring intrinsic inductive bias, Advances in Neural Information Processing Systems 34 (2021).

[20] 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, arXiv preprint arXiv:2010.11929 (2020).

[21] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, S. Xie, A convnet for the 2020s, arXiv preprint arXiv:2201.03545 (2022).

[22] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2818–2826.

[23] I. Loshchilov, F. Hutter, Fixing weight decay regularization in adam (2018).

[24] Z. Zhong, J. Cui, S. Liu, J. Jia, Improving calibration for long-tailed recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,

2021, pp. 16489–16498.

[25] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, A. A. Kalinin, Albumentations: Fast and flexible image augmentations, Information 11 (2020). URL: https://www.mdpi.com/2078-2489/11/2/125. doi:10.3390/info11020125.

[26] S. G. Müller, F. Hutter, Trivialaugment: Tuning-free yet state-of-the-art data augmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 774–782.

[27] Z. Zhong, L. Zheng, G. Kang, S. Li, Y. Yang, Random erasing data augmentation, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2020, pp. 13001–13008.

[28] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, Y. Yoo, CutMix: Regularization strategy to train strong classifiers with localizable features, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6023–6032.

[29] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, arXiv preprint arXiv:1710.09412 (2017).

[30] Vasilis Vryniotis, How to train state-of-the-art models using torchvision's latest primitives, 2021. https://pytorch.org/blog/how-to-train-state-of-the-art-models-using-torchvision-latest-primitives/.

[31] A. Adcock, V. Reis, M. Singh, Z. Yan, L. van der Maaten, K. Zhang, S. Motwani, J. Guerin, N. Goyal, I. Misra, L. Gustafson, C. Changhan, P. Goyal, Classy vision, https://github.com/facebookresearch/ClassyVision, 2019.

[32] F. Klinker, Exponential moving average versus moving exponential average, Mathematische Semesterberichte 58 (2011) 97–107.

[33] H. Touvron, A. Vedaldi, M. Douze, H. Jégou, Fixing the train-test resolution discrepancy, Advances in neural information processing systems 32 (2019).

[34] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, R. Girshick, Masked autoencoders are scalable vision learners, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16000–16009.

[35] 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, Ieee, 2009, pp. 248–255.

[36] R. Wightman, Pytorch image models, https://github.com/rwightman/pytorch-image-models, 2019. doi:10.5281/zenodo.4414861.

[37] S. Vaze, K. Han, A. Vedaldi, A. Zisserman, Open-set recognition: A good closed-set classifier is all you need, arXiv preprint arXiv:2110.06207 (2021).

[38] Q. Diao, Y. Jiang, B. Wen, J. Sun, Z. Yuan, Metaformer: A unified meta framework for fine-grained recognition, arXiv preprint arXiv:2203.02751 (2022).