1st Place Solution for FungiCLEF 2022 Competition: Fine-grained Open-set Fungi Recognition

Zihua Xiong, Yumeng Ruan, Yifei Hu, Yue Zhang, Yuke Zhu, Sheng Guo and Bing Han

MYbank, Ant Group, China

Abstract

In this paper, we describe our method for Fine-Grained Fungi Recognition at FungiCLEF 2022, which aims to recognize the fungi belonging to 1,604 known species and many other unknown species, termed as a fine-grained, open-set machine learning problem. For the purpose of building a strong close-set classifier, we taken MetaFormer [1] and ConvNext [2] as our strong baseline, then we applied hyper-parameter tuning and some modern training techniques to improve it. To deal with long tailed class distribution problem, we adapt the Seesaw Loss [3] to balance the training process between head classes and tail classes. Furthermore, to avoid tail categories being misclassified as open-set categories, we intuitively design a post process to alleviate the confusion. As a common practice, test time augmentations and model ensemble are used. With all these techniques together, our method achieves superior mean $f1$ score on test set, that is 83.78% on public leaderboard, and 80.43% on private leaderboard which is the **1st** place among the participators. The code will be made available at https://github.com/guoshengcv/fgvc9_fungiclef.

Keywords

FungiCLEF, fungi recognition, long tail, open-set, fine-grained, classification

1. Introduction

FungiCLEF 2022 [4] is a competition held jointly by CLEF 2022 conference [5, 6] and FGVC9 workshop at CVPR 2022 conference. The competition release the train data based on Danish Fungi 2020 [7], which aims at fine-grained fungi recognition. It includes both image and meta-information such as habitat, substrate, time, longitude, latitude etc, and contains 295,938 samples belonging to 1,604 species. The competition also release test data which contains 59,420 observations with 118,676 images and 3,134 species, it includes meta-information as train data but miss some attributes such as longitude and latitude. After data analyze, we found the category distribution of the train dataset is long-tailed, as a result, in this work we tackle the competition as a fine-grained, long-tailed, open-set classification task.

Different from common classification tasks that try to distinguish objects with large inter-class variations, Fine-Grained Visual Classification (FGVC) aims at capturing the subtle difference within similar categories, such as differentiating bird species, car types, etc. It is acknowledged that FGVC is a challenging task due to small inter-class variations and large intra-class variations.

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

xiongzihua.xzh@alibaba-inc.com (Z. Xiong); ruanyumeng.rym@alibaba-inc.com (Y. Ruan); huyifei.hyf@alibaba-inc.com (Y. Hu); mosay.zy@alibaba-inc.com (Y. Zhang); felix.yk@alibaba-inc.com (Y. Zhu); guosheng.guosheng@mybank.cn (S. Guo); hanbing.hanbing@antgroup.com (B. Han)

CEUR Workshop Proceedings (CEUR-WS.org)

Numerous methods for FGVC are mainly focused on modeling discriminative regions, such as part-based model [8, 9, 10] and attention-based model [11, 12]. Recently, inspired by the fact that human experts use meta-information to distinguish visually similar species, there are many works [13, 14, 1, 15] utilize additional information to enhance fine-grained classification performance. Among them, Metaformer [1] is the state-of-the-art work proposed recently, it is a hybrid framework that convolution and transformer are both used. In this work, we taken Metaformer as a strong baseline and improve it progressively.

Recently, transformers have leading the research in the filed of computer vision, starting from Vision Transformer [16], there are many various transformer backbones achieve SoTA performance in a wild range of vision tasks, such as Swin Transformer [17], CSwin Transformer [18], etc. On the other side, ConvNext [2] is a pure convolution backbone, it applies modern training techniques, macro and micro design of the network architecture, achieves comparable results with transformers. In this work, in order to obtain models with distinct difference and enhance the performance of model ensemble, we taken ConvNext as another baseline backbone.

In real world scenarios, the distribution of the categories is often long-tailed. It is well known that major class will dominate the training process and suppress the performance of tail class. Many works designed loss function to deal with the problem of long-tail classification, such as Adaptive Class Suppression Loss [19], Equalization loss [20], Seesaw Loss [3], etc. In our method, we utilize Seesaw Loss to dynamically balance the training process between head classes and tail classes.

For practical application, it usually faces with open-set recognition challenge, the classifier should not only recognize the classes which have been seen during training, but also notice that if a instance comes from unknown classes. Motivated by [21], in this work, we trained the model on known classes to obtain a good close-set classifier, and determine whether a instance belonging to open-set based on the maximum value of it's logit score vector. Furthermore, to avoid tail categories being misclassified as open-set categories, we intuitively design a post process to alleviate the confusion.

Our main contributions in FungiCLEF 2022 competition can be summarized as follows:

- We take Metaformer and ConvNext as our strong baseline, then we apply hyper-parameter tuning and some modern training techniques to improve it's performance on fungi dataset.
- We find category distribution of the fungi dataset is long-tailed, thus we adapt the Seesaw Loss to balance the training process between head classes and tail classes, which lift up the baseline model performance.
- To avoid tail categories being misclassified as open-set categories, we intuitively design a post process to alleviate the confusion.
- Detailed ablation experiments have been done. With the techniques above, we achieve superior performance.

2. Approach

Motivated by [21], we divide the fine-grained, open-set recognition problem into two parts. Firstly we are attended to lift up the close-set recognition performance, including network architecture,

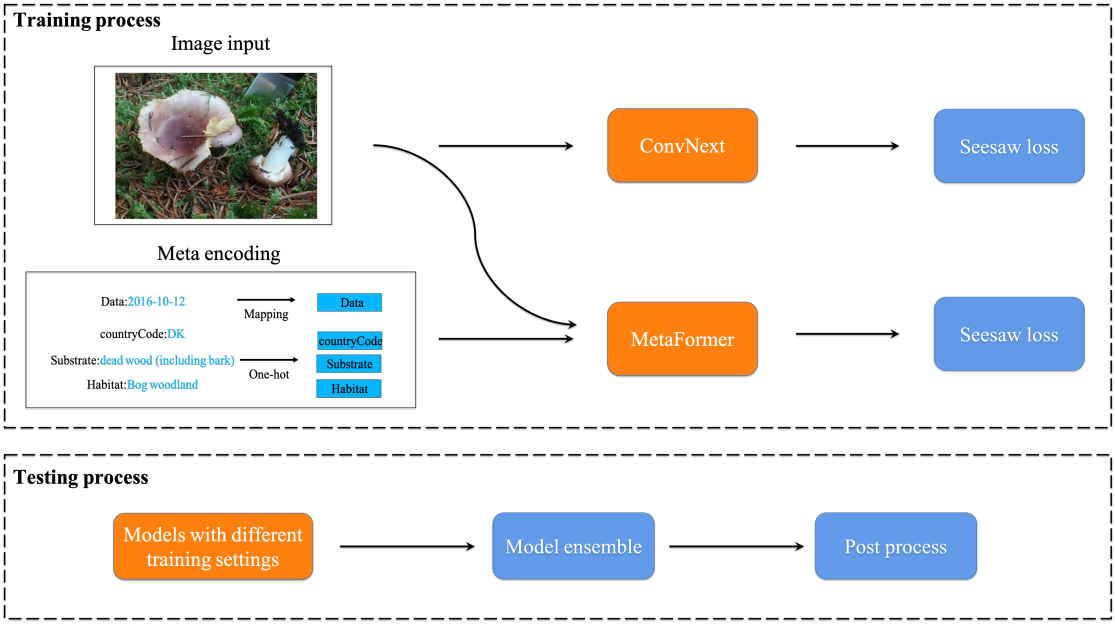

Figure 1: The overall of our apporach. We trained MetaFormer and Convnext with various settings, during testing process, model snsemble and post process are used.

hyper-parameter tuning and long tail loss. Then we analyze the data and logit score frequency distributions, and design the post process for open-set recognition. Model ensemble is done by averaging the output logits from multiple models.

2.1. Overview of the Approach

As shown in Figure 1, we taken MetaFormer [1] and ConvNext [2] as our initial baseline. MetaFormer is a hybrid framework that combines convolution and vision transformer, it also proposes a simple and effective solution for adding meta-information using the transformer layer. In our approach, we directly use MetaFormer and modify the input of meta-information. We perform the mapping $[month, day] \rightarrow [sin(\frac{2\pi month}{12}), cos(\frac{2\pi month}{12}), sin(\frac{2\pi day}{31}), cos(\frac{2\pi day}{31})]$ to encode temporal information. We use one-hot encoding to encode category meta-information such as *countryCode*, *Substrate* and *Habitat*. To enhance the model diversity for later model ensemble, we use ConvNext as another network architecture. We apply hyper-parameter tuning to improve their performance, and we will illustrate the ablation studies in Sec 3.2 to show the progressive process.

2.2. Loss for Long Tail Classification

It is known that instances from head categories dominate the training process, the biased learning lead to misclassification for tail categories. In this work, we borrow the idea from Seesaw Loss [3] to alleviate this problem. During training process, Seesaw Loss dynamically balances positive and negative gradients for each category with a dynamic factor, it reformulate the Cross Entropy

loss as

$$
\begin{aligned}
L_{seesaw}(z) &= -\sum_{i=1}^{C} y_i \log(\widehat{p}_i), \\
\text{with } \widehat{p}_i &= \frac{e^{z_i}}{\sum_{j \neq i}^{C} S_{ij} e^{z_j} + e^{z_i}}.
\end{aligned}
\tag{1}
$$

where y is the category label, usually represented by one-hot, z is the outputs of model, $\widehat{p}$ is the probability calculated by $Softmax(z)$ with a dynamic factor S. For more detail, please refer to Seesaw Loss [3].

2.3. Post Process

The post process is intuitively designed based on several observations, and applied on final ensemble results. In order to be as clear as possible the process, we write the post process in python-style pseudocode, refer to the Algorithm 1. We detailed it in the following.

Threshold for selecting open-set samples. It is acknowledged that the predict confidence score of open-set samples are relatively low. This phenomenon can be used as the criterion for their recognition. Specifically, we draw the logit (the direct output of the model) frequency distribution of both validation set and test set, shown in Figure 2. As the open set samples are only contained in the test set, we can compare the low confidence areas of the two distributions to approximately get the logit threshold for open set samples. For example, we can set threshold to 5 as an approximate for the Figure 2. On this basis, we have draw a rough conclusion that the test set contains approximate $1000 \sim 2000$ open set samples. It should be noted that this rough conclusion may be wrong, since we have no information about the reality that how many open-set samples in test set. Despite of it, in the rest of the experiments, we set the samples with top-k lowest confidence as the open set, the value of k is set based on this rough conclusion. As shown from line 7 to line 9 in Algorithm 1, we adjust the threshold to obtain open-set samples. For different experiment setting and model, it is hard and needless to have exactly same number of open-set samples, experimentally, we set k to ~ 1000 at first, and adjust it to ~ 1500 at the final based on the public test set performance.

Alleviate the influence of microscopy images. In the test set, we find that one test sample may contains several images as shown in Figure 3. For such case, we average the model outputs of them to get the confidence and use argmax to get predicted category. During the above process, we also find that there are images showing huge visual discrepancy with the majority. Specifically, we find that some test samples contain microscopy images such as sample_c shown in Figure 3, these microscopy images tend to produce low confidence due to little training data, it will influence the naive average strategy. To cope with this problem, we delicately design the post process. As shown in Algorithm 1, from line 29 to line 35, if the maximum logit of averaged outputs is lower than a certain threshold, we will look into the maximum logit of all images from a test sample, if it is greater than a certain threshold, we will get the corresponding category as the prediction. In this case, we think that the test samples with low averaged outputs may be caused by containing too many microscopy images, the high confidence prediction of one image from test sample is sufficient to infer the category, and the test sample should not be considered as open-set categories.

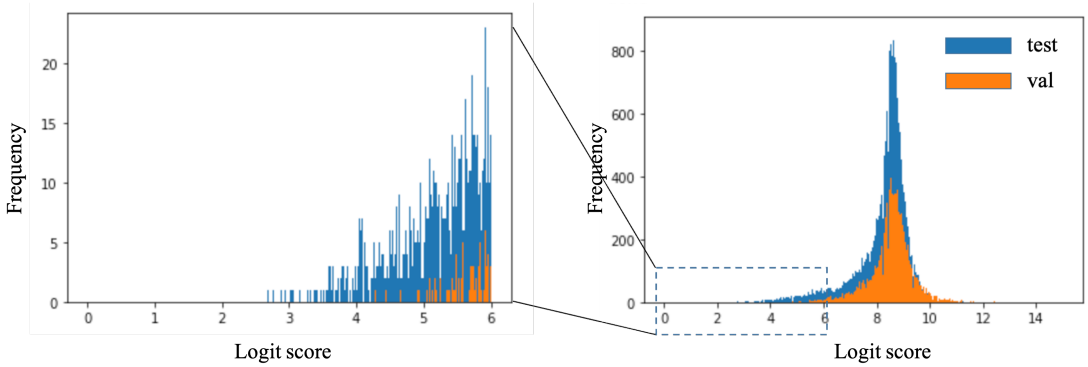

Figure 2: Logit score frequency distribution of val and test. We draw the Logit score frequency distribution on both validation and test set, with the output logits of a single model.

Figure 3: Selected test samples. We found many test samples contain several images, we predict the category of the test sample by averaging the model outputs of them. We also notice that some test samples such as sample_c contains one image pictured from natural environment, and the other images come from microscope view, it will disturb the average results.

Distinguish tail categories and open-set categories. We put the tail categories that are never been predicted by the model into hard tail categories, we argue that there are many hard tail categories misclassified as open-set categories. To deal with the problem above, we design the post process, refer to the Algorithm 1. From line 18 to line 27, to avoid the misclassification, we mining hard tail categories from top-3 predictions with low threshold filtering.

Algorithm 1 Pseudocode of Post Process in a python-like style.

```
# logit_scores: logit scores of a test sample
# high_t: high threshold
# low_t: low threshold
# hard_tail_categories: list of hard tail categories

# we dynamically adjust the threshold to have certain number of open-set samples
high_t = 9.8
low_t = high_t - 0.7

# we average logit scores of all images from test sample, then get top 3 classes and scores
mean_scores = mean(logit_scores, dim=0)
top1_cls, top2_cls, top3_cls, top1_score, top2_score, top3_score = topk(mean_scores, topk=3)

# we maximum logit scores of all images from test sample
max_scores = max(logit_scores, dim=0)
max_cls, max_score = argmax(max_scores)

# we distinguish tail categories and open-set categories
# by mining hard tail categories from top-3 predictions
if top1_score > low_t and top1_cls in hard_tail_categories:
   predicted_cls = top1_cls
elif max_score > low_t and max_cls in hard_tail_categories:
   predicted_cls = max_cls
elif top2_score > low_t and top2_cls in hard_tail_categories and top1_score - top2_score < 1.2:
   predicted_cls = top2_cls
elif top3_score > low_t and top3_cls in hard_tail_categories and top1_score - top3_score < 1.2:
   predicted_cls = top3_cls

# to alleviate the influence of microscopy images
if top1_score > high_t:
   predicted_cls = top1_cls
# if the maximum logit of averaged outputs is lower than a certain threshold,
# we will look into the maximum logit of all images from a test sample
elif max_score > 15:
   predicted_cls = max_cls
# recognized as open-set categories
else:
   predicted_cls = -1
```

3. Experiments

In this section, we first elaborate on the implementation and training details. Then we introduce ablation studies on loss functions and bag of training settings. Then we list some other attempts and it's results. Finally we study on different test time augmentations, and show the effectiveness of post process for tail categories recognition and open-set categories recognition.

3.1. Implementation Details

We trained the model on Danish Fungi 2020 dataset [7] which contains 295,938 training images belonging to 1,604 species observed mostly in Denmark, the dataset has been divided into train and validation set. We use both train and validation for training in most settings. We report the results on test set which contains 59,420 observations with 118,676 images and 3,134 species. The test set is divided into 2 parts, the public set contains 20% of the data, the private set contains 80% of the data. As the performance of open-set recognition affects the mean $f1$ score, to make relative fair comparison in ablation studies, based on the observation illustrated in Sec 2.3, we utilize threshold to select $\sim$1000 samples which have low confidence score as open-set samples for most experiments. We conduct all the experiments with Tesla V100 (32G). We use AdamW optimizer with cosine learning scheduler, initialize the learning rate to $5e^{-5}$ and scale it by batch size, we follow most of the augmentation and regularization strategies of [17] in training.

Table 1
MetaFormer-0 baseline.

loss	batch size	accumulate steps	epochs	mixup	train+val	public mean $f1$	private mean $f1$
Soft Target CE	32	1	100	yes	no	78.76%	74.26%

Table 2
Mean $f1$ score on public/private test set with different **losses** and MetaFormer-0 as backbone.

loss	batch size	accumulate steps	epochs	mixup	train+val	public mean $f1$	private mean $f1$
Soft Target CE	32	1	32	yes	no	71.49%	67.6%
Label Smoothing CE	32	1	32	no	no	**76.9**$\%$	**72.48**$\%$
Label Smoothing CE	64	3	32	no	yes	79.45%	75.67%
Seesaw Loss	64	3	32	no	yes	**79.79**$\%$	**76.15**$\%$

Table 3
Mean $f1$ score on public/private test set with different **batch size** and MetaFormer-0 as backbone.

loss	batch size	accumulate steps	epochs	mixup	train+val	public mean $f1$	private mean $f1$
Label Smoothing CE	32	1	32	no	no	76.90%	72.48%
Label Smoothing CE	64	1	32	no	no	**77.49**$\%$	**74.27**$\%$

Table 4
Mean $f1$ score on public/private test set with different **accumulate steps**.

loss	batch size	accumulate steps	epochs	backbone	train+val	public mean $f1$	private mean $f1$
Seesaw Loss	64	3	32	MetaFormer-0	yes	79.79%	76.15%
Seesaw Loss	64	6	32	MetaFormer-0	yes	**80.22**$\%$	**76.90**$\%$
Seesaw Loss	32	3	64	MetaFormer-1	yes	**81.67**$\%$	77.62%
Seesaw Loss	32	6	64	MetaFormer-1	yes	81.66%	**77.94**$\%$

3.2. Ablation Studies

As shown in Table 1, we train MetaFormer-0 for 100 epochs, with Soft Target Cross Entropy loss and mixup [22] augmentation to build our baseline. For ablation studies, it should be noted that except the parameter to be compared, there are little other not consistent parameters, such as the *accumulate steps* in last row in Table 5, we argue that it will not affects the conclusion largely.

Losses. As shown in Table 2, we compare different losses with several common augmentation techniques. Specifically, we compare cross entropy loss with either mixup or label smoothing [23] and Seesaw loss. They are all devoted to alleviate the long-tail problem in training. It is found that label smooth converges faster than mixup in our experiments. The best performance is achieved when Seesaw loss is adopted.

Batch size. Table 3 illustrates that larger batch size improves the performance. in detail, by increasing batch size from 32 to 64, we improved $f1$ score from 76.90% to 77.49% on public set, consistently improve $f1$ score from 72.48% to 74.27% on private set. Similar techniques is to increase the accumulate steps. As shown in Table 4, enlarging accumulate steps improves mean $f1$ score in private test set consistently with MetaFormer-0 and MetaFormer-1.

Training epochs. We found the longer training epochs will not definitely improve the performance. As shown in Table 5, for MetaFormer-0 and MetaFormer-2, it is consistent that proper epochs is essential for better result. Following this line, we did not train models with dozens of epochs such as 100 epochs and above.

Image size. Usually, training with larger image size improves the overall performance, especially for fine-grained tasks. We use the 384 as the baseline image size and try several other larger settings. As shown in Table 6, the larger image size 448 does not consistently bring improvements

Table 5
Mean $f1$ score on public/private test set with different **training epochs**.

loss	batch size	accumulate steps	epochs	backbone	train+val	public mean $f1$	private mean $f1$
Seesaw Loss	64	3	32	MetaFormer-0	yes	79.79%	76.15%
Seesaw Loss	64	3	64	MetaFormer-0	yes	**80.46**%	**77.01**%
Seesaw Loss	64	3	100	MetaFormer-0	yes	80.18%	76.77%
Seesaw Loss	24	4	32	MetaFormer-2	yes	81.18%	77.56%
Seesaw Loss	24	4	48	MetaFormer-2	yes	**82.04**%	**77.92**%
Seesaw Loss	24	6	64	MetaFormer-2	yes	80.45%	77.63%

Table 6
Mean $f1$ score on public/private test set with different **image size** and MetaFormer-1 as backbone.

image size	batch size	accumulate steps	epoch	public mean $f1$	private mean $f1$
384	32	6	64	**81.76**%	78.25%
448	20	6	64	80.79%	**78.48**%

Table 7
Mean $f1$ score on public/private test set with different **pretrain dataset** and MetaFormer-2 as backbone.

pretrain dataset	batch size	accumulate steps	epochs	public mean $f1$	private mean $f1$
herbarium	12	6	32	80.90%	77.37%
imagenet22k	12	8	48	81.47%	77.86%
inaturalist21	24	6	48	82.04%	77.92%

Table 8
Mean $f1$ score on public/private test set with ConvNext-tiny, ConvNext-base and ConvNext-large as backbone. Notice that we only use image data to train ConvNext.

batch size	accumulate steps	epochs	+pseudo label	backbone	public mean $f1$	private mean $f1$
96	3	64	no	convnext-tiny	76.93%	73.46%
32	4	64	no	convnext-base	78.97%	75.46%
10	4	64	no	convnext-large	79.15%	75.59%
24	6	80	yes	convnext-large	80.65%	76.61%

on public test set. We blame it to the coupled training schedules with the image size, which we do not investigate into it. Finally, we adopt the image size 384 in all settings. Nevertheless, the performance with this baseline is satisfactory enough.

Pretrain dataset. We transfer MetaFormer-2 pretrained on different dataset such as herbarium, imagenet22k and inaturalist21. The results are shown in Table 7. Experimentally, we do not directly choose the best-performed pre-training model. Instead, we use ensemble techniques to combine them. We find that ensemble will produce a consistent improvement compared with single model. Even combining the best-performed single model with other slightly poorer-performed models will not affect the conclusion.

3.3. Other Attempts

ConvNext. In addition to MetaFormer, We also train ConvNext. The results are listed in Table 8. The experiments in ConvNext is not fully explored compared to MetaFormer. Although the results of ConvNext are inferior to MetaFormer, we still add it to the model ensemble process and the performance is also improved.

Table 9
Mean $f1$ score on public/private test set with pseudo label, ConvNext-large and MetaFormer-2 as backbone.

backbone	batch size	accumulate steps	epochs	public mean $f1$	private mean $f1$
ConvNext-large	24	6	80	80.65%	76.61%
MetaFormer-2	24	8	80	82.45%	77.93%

Table 10
Single model's mean $f1$ score on public/private test set with different test time augmentation.

test time augmentation	public mean $f1$	private mean $f1$
center crop / five crop	81.66% / **81.76**$\%$	77.94% / **78.25**$\%$
center crop / five crop	**81.67**$\%$ / 81.63%	77.62% / **77.69**$\%$
five crop / multi scale & ten crop	**80.46**$\%$ / 80.20%	77.02% / **77.31**$\%$

Table 11
Ensemble model's mean $f1$ score on public/private test set with different test time augmentation.

test time augmentation	public mean $f1$	private mean $f1$
center crop / multi scale & ten crops	83.20% / **83.26**$\%$	**79.51**$\%$ / 79.38%

Table 12
The effectiveness of post process for tail categories recognition and open-set categories recognition.

ensemble and post process	number of open-set samples	public mean $f1$	private mean $f1$
average ensemble (v1)	$\sim$1000	83.26%	79.38%
average ensemble (v2)	$\sim$1500	83.50%	79.60%
average ensemble (v3)	$\sim$1500	83.65%	79.79%
average ensemble (v4)	$\sim$1500	**83.78**$\%$	**80.43**$\%$

Pseudo label. After training models with various settings, we use model ensemble to get the best model currently, and take the model predictions on test samples as their label. We select top $\sim 50\%$ test samples by their confidence score. We trained MetaFormer-2 and ConvNext-large with train+val+pseudo, the results are listed in Table 9.

3.4. Test Time Augmentation and Post Process

Test Time Augmentation. For test time augmentation, we use center crop, five crop and multi scale & ten crop during test phase. The effects of test time augmentation(TTA) are shown in Table 10 and Table 11. It should be noted that the mean $f1$ score on public test set is out of accord with private test set in some experiments, and it is hard to decide which TTA is better only based on public score, in consideration of robustness, we have chosen multi scale & ten crop based on the public mean $f1$ in Table 11.

Post Process. For short, we name the different version of ensemble and post process as v1 (initial version), v2 (v1 + proper open-set threshold), v3 (v2 + models trained with pseudo label) and v4 (v3 + post process for tail categories).

For open-set recognition, we intuitively select samples with lower confidence score as open-set samples. As shown in Table 12, by increasing open-set sample from ~ 1000 to ~ 1500, we improve mean $f1$ score from 83.26% to 83.50% on public test set, from 79.38% to 79.60% on

private test set. It demonstrates the effectiveness to select a proper open-set threshold. Compare the average ensemble (v3) with average ensemble (v2), v3 improves the ensemble performance, the only difference between them is that v3 contains the models trained with pseudo label. It is acknowledged that the tail categories tend to have lower confidence score compared to head categories, so the tail categories are easier to be misclassified as head categories or wrongly identified as open-set categories. As shown in Table 12, with our post process for tail categories applied on average ensemble (v3), which termed as average ensemble (v4), the mean $f1$ score improves a lot on both public test set and private test set.

4. Conclusion

In this paper, we introduce our solution for FungiCLEF 2022 competition. To solve this challenging fine-grained, open-set problem, we try a bunch of techniques, such as different network baseline, hyper-parameters tuning, modern training techniques, loss for long tail recognition and specially designed post process. With these endeavours we achieved 1st place among the participators. The experimental results show the progressive process for single model, and the effectiveness of test time augmentation and post process for tail categories. For future work, it is valuable to study the method that fuse meta-information and visual information for Fine-Grained Visual Classification, and the problem of distinguish between tail categories and open-set categories is also worth exploring.

References

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

[2] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, S. Xie, A convnet for the 2020s, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022).

[3] J. Wang, W. Zhang, Y. Zang, Y. Cao, J. Pang, T. Gong, K. Chen, Z. Liu, C. C. Loy, D. Lin, Seesaw loss for long-tailed instance segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 9695–9704.

[4] 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.

[5] 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.

[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. Matas, J. Heilmann-Clausen, T. S. Jeppesen, T. Læssøe, T. Frøslev, Danish fungi 2020 - not just another image recognition dataset (2021). arXiv:2103.10107.

[8] W. Ge, X. Lin, Y. Yu, Weakly supervised complementary parts models for fine-grained image

classification from the bottom up, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3034–3043.

[9] C. Liu, H. Xie, Z.-J. Zha, L. Ma, L. Yu, Y. Zhang, Filtration and distillation: Enhancing region attention for fine-grained visual categorization, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2020, pp. 11555–11562.

[10] Z. Yang, T. Luo, D. Wang, Z. Hu, J. Gao, L. Wang, Learning to navigate for fine-grained classification, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 420–435.

[11] J. He, J.-N. Chen, S. Liu, A. Kortylewski, C. Yang, Y. Bai, C. Wang, A. Yuille, Transfg: A transformer architecture for fine-grained recognition (2021). arXiv:2103.07976.

[12] Y. Gao, X. Han, X. Wang, W. Huang, M. Scott, Channel interaction networks for fine-grained image categorization, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2020, pp. 10818–10825.

[13] X. He, Y. Peng, Fine-grained image classification via combining vision and language, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5994–6002.

[14] G. Chu, B. Potetz, W. Wang, A. Howard, Y. Song, F. Brucher, T. Leung, H. Adam, Geo-aware networks for fine-grained recognition, in: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, 2019, pp. 0–0.

[15] O. Mac Aodha, E. Cole, P. Perona, Presence-only geographical priors for fine-grained image classification, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9596–9606.

[16] 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: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021, OpenReview.net, 2021. URL: https://openreview.net/forum?id=YicbFdNTTy.

[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, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021.

[18] X. Dong, J. Bao, D. Chen, W. Zhang, N. Yu, L. Yuan, D. Chen, B. Guo, Cswin transformer: A general vision transformer backbone with cross-shaped windows, 2021. arXiv:2107.00652.

[19] T. Wang, Y. Zhu, C. Zhao, W. Zeng, J. Wang, M. Tang, Adaptive class suppression loss for long-tail object detection, 2021.

[20] J. Tan, C. Wang, B. Li, Q. Li, W. Ouyang, C. Yin, J. Yan, Equalization loss for long-tailed object recognition, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020, Computer Vision Foundation / IEEE, 2020, pp. 11659–11668. URL: https://openaccess.thecvf.com/content_CVPR_2020/html/Tan_Equalization_Loss_for_Long-Tailed_Object_Recognition_CVPR_2020_paper.html. doi:10.1109/CVPR42600.2020.01168.

[21] S. Vaze, K. Han, A. Vedaldi, A. Zisserman, Open-set recognition: a good closed-set classifier is all you need?, in: International Conference on Learning Representations, 2022.

[22] H. Zhang, M. Cissé, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, OpenReview.net, 2018. URL: https://openreview.net/forum?id=r1Ddp1-Rb.

[23] R. Müller, S. Kornblith, G. Hinton, When does label smoothing help?, in: Proceedings of the 33rd International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, USA, 2019.