Optimizing Fine-Grained Fungi Classification for Diverse Application-Oriented Open-Set Metrics

Stefan Wolf[1,2,3], Jürgen Beyerer[2,1,3]

[1] Vision and Fusion Lab, Karlsruhe Institute of Technology KIT, c/o Technologiefabrik, Haid-und-Neu-Str. 7, 76131 Karlsruhe, Germany

[2] Fraunhofer IOSB, Institute of Optronics, System Technologies and Image Exploitation, Fraunhoferstrasse 1, 76131 Karlsruhe, Germany

[3] Fraunhofer Center for Machine Learning

Abstract

Fine-grained fungi species classification is an important task to support distinguishing edible and poisonous fungi and thus, reducing the risk of accidental poisoning. Therefore, the FungiCLEF 2023 challenge seeks to find the best solution for this task considering multiple metrics with each having a different application in focus like e.g., a low confusion of edible and poisonous fungi. We propose a method to approach the different metrics by exploiting modern deep learning networks, strong data augmentation and class-balanced training. The challenge assumes an open-set scenario which includes unknown classes during evaluation which we identify by a confidence thresholding approach. With our method, we achieved the 2nd place in the challenge with good scores across all metrics. Code is available at: https://github.com/wolfstefan/fungi2023.

Keywords

Fungi classification, Open-set classification, FungiCLEF, Vision transformer

1. Introduction

While fungi are a common food in many cultures, distinguishing edible fungi from poisonous ones can be difficult. Thus, an automatic system for identifying the species of fungi can support in saving lives by reducing the risk of eating poisonous fungi. However, as common to many fine-grained classification tasks, fungi species classification is a difficult task which is still not solved due to the low inter-class variance of similar looking species. We propose a method for fine-grained fungi classification that is capable of achieving a high classification accuracy while rejecting samples of classes which were not part of the training dataset. With this approach, we achieved the 2nd place in the FungiCLEF 2023 [1] challenge, part of the LifeCLEF 2023 [2, 3] lab, that seeks to find the best approach for fine-grained fungi classification with an emphasis on open-set classification and reducing the chance of confusing edible with poisonous fungi.

CLEF 2023: Conference and Labs of the Evaluation Forum, September 18–21, 2023, Thessaloniki, Greece

stefan.wolf@iosb.fraunhofer.de (S. Wolf); juergen.beyerer@iosb.fraunhofer.de (J. Beyerer)

CEUR Workshop Proceedings (CEUR-WS.org)

2. Related work

Fine-grained classification of fungi species is approached by multiple studies. While Zieliński et al. [4] investigate the use of microscopic images for fine-grained fungi classification with a bag-of-words approach based on deep learning features, most works use wildlife images for the classification due to the easier applicability. While recent approaches are mostly based on deep learning architectures, multiple extensions for the task has been explored. Sulc et al. [5] apply multiple deep learning models in an esemble in order to improve the classification accuracy. Picek et al. [6] exploit additional metadata like location, habitat and substrate. Kiss and Czùni [7] evaluate multiple strategies of training deep learning models for improving the accuracy of mushroom type classification. In the context of the FungiCLEF 2022 [8] challenge, multiple methods have been explored that increase the accuracy of fine-grained fungi classification in an open-set scenario.

3. Method

Our method is based on the work of Wolf and Beyerer [9] and we explore the application of a newer Swin Transformer V2 [10] feature extraction backbone which shows advantageous for some of the evaluation metrics.

3.1. Feature extraction backbone

To extract the features, we explore a modern Swin Transformer V2 [10] backbone besides its predecessor Swin Transforer [11]. Swin Transformer V2 extends the Swin Transformer by multiple extensions to stabilize the training for larger number of parameters. I.e., these are a novel post normalization technique, a scaled cosine attention and a log-spaced continuous position bias technique.

3.2. Classification head

The feature extraction backbone provides a feature vector for each image. During training, the classification head is applied for each single image. During evaluation, the feature vectors of an observation are averaged and the classification head is applied on the averaged feature vector. As Wolf and Beyerer [9] have shown, this fusion strategy is advantageous compared to averaging the class-wise scores after the application of the classification head. The classification head itself consists of a linear layer predicting logits and a softmax activation layer resulting in normalized confidence scores. To identify samples of unknown classes, a threshold is applied on the maximum score. If the highest score is below the threshold, the sample is rejected as unknown.

3.3. Class-balancing

In order to cope with the large imbalance of the used datasets as common for fine-grained classification tasks, we apply a class-balancing strategy. I.e., we apply the data resampling scheme proposed by Gupta et al. [12] that increases the likelihood of samples of rare classes occurring in training. It is parameterized by an oversampling threshold that controls how rare a class must occur in the training set so that images of the class are resampled.

4. Evaluation

4.1. Datasets

For training and evaluation, we use two datasets. The first is the Danish Fungi 2020 [6] dataset which includes 295,938 images from 1,604 different fungi species. All of these images are used for training. For the evaluation, we use the Danish Fungi 2021 [8] dataset with 59,420 observations and 118,675 images in total. Of these observations, 30,131 obersvations with 60,832 images are part of the official FungiCLEF 2023 [1] validation set which we use to calculate the reported metrics.

4.2. Training setup

All models are trained with MMPreTrain [13], a classification framework based on PyTorch [14]. The weights are initialized from checkpoints pretrained on ImageNet-21k [15]. We employ an AdamW [16] optimizer with an initial learning rate of $1.25 \cdot 10^{-4}$ and a batch size of 128. For models which are to large to fit the VRAM, we reduce the batch size and scale the learning rate according to the linear scaling rule. Of the images, a random crop of the size of 8% to 100% of the original image is taken and resized to 384×384 pixels. Afterwards, the images are flipped horizontally with a 50% chance. In order to increase the generalization of the model, RandAugment [17] and random erasing [18] is applied. The images are normalized with the default settings of PyTorch. We apply a label smoothing loss [19] with a smoothing value of 0.1. For experiments involving class balancing, an oversample threshold of 10^{-2} is applied. Additionally, the training duration is shortened from 24 epochs to 6 epochs to keep the total number of iterations consistent with oversampling. If not mentioned otherwise, class balancing is enabled for all experiments. The training is performed image-wise ignoring that images are part of observations. The models have been trained on 6 Nvidia V100, 4 Nvidia A100 or 4 Nvidia H100 depending on availability.

4.3. Evaluation setup

During evaluation, each image is scaled to 438 pixels on the shorter edge and afterwards, a center crop of 384×384 pixels is taken. The normalization is applied similar to the training configuration. Each image of an observation is fed independently to the feature extraction backbone and all resulting feature vectors of an observation are averaged. Finally, the classification result is calculated by applying the linear classification layer onto the averaged feature vector followed by a softmax activation.

4.4. Metrics

For the experiments, we evaluate the five metrics from the challenge:

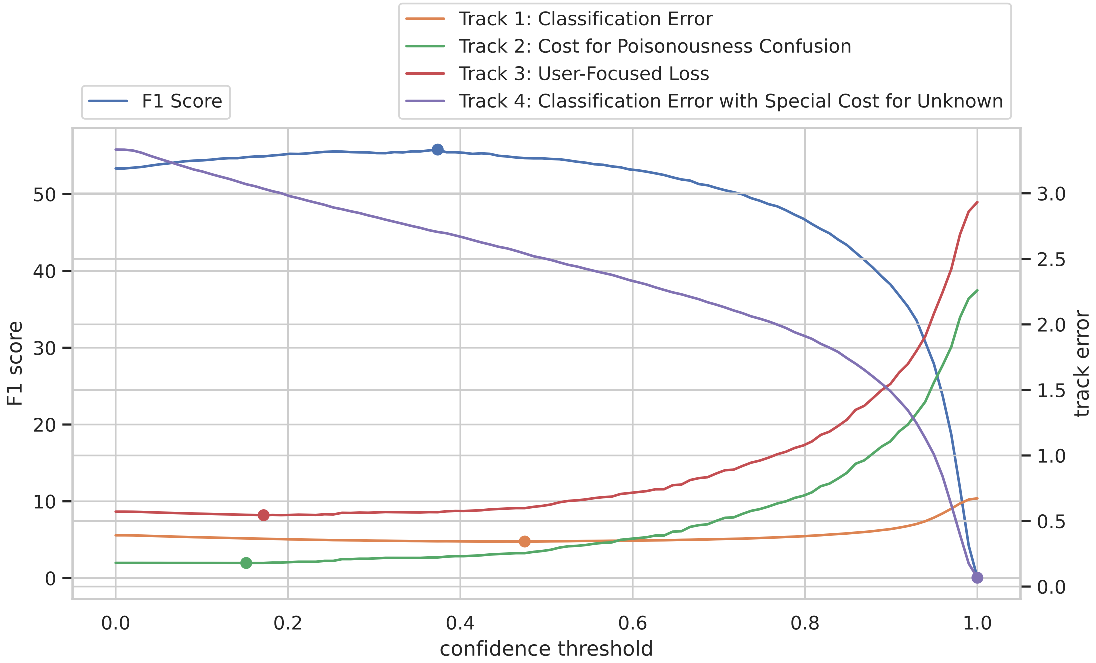

Figure 1: The relevant metrics depending on the confidence threshold for the open-set recognition for a Swin Transformer Base model. The best threshold for each metric is marked with a point. The optimal threshold is highly dependent on the metric which should be optimized. While track 2 and 3 are profiting mostly from a low threshold below of 0.2, the F1 score and track 1 are optimized with a medium range threshold between 0.3 and 0.5. Track 4 can be optimized to an almost perfect value with a threshold of 1.0 meaning that no classification is taking place and all samples are classified as unknown samples.

- F1 Score: the macro-average of the class-wise F1 score.
- Track 1: Classification Error: the ratio of misclassified samples to total number of samples.
- Track 2: Cost for Poisonousness Confusion: the ratio of samples which are confused between edible and non-edible to the total number of samples whereby non-edible samples classified as edible are weighted 100×.
- Track 3: User-Focused Loss: the sum of the track 1 and track 2 error.
- Track 4: Classification Error with Special Cost for Unknown: similar to track 1, a ratio of misclassified samples to the total number of samples. However, samples with unknown classes classified as a known class are weighted 100× while samples of known classes which are classified as unknown are weighted 0.1×.

4.5. Confidence threshold

First on, we analyze the impact of the applied confidence threshold due to the importance of properly choosing the confidence value for further experiments. The results for a Swin Base model are shown in Figure 1. Overall, they indicate that a single confidence threshold is not well suited for all metrics. The metrics can be roughly grouped in three sets. The first group are

Table 1
Comparison of backbone architectures. We evaluate the Base and Large variants of Swin Transformer and Swin Transformer V2. While the SwinV2 Base model shows a slight advantage over the Swin Base model, the Swin Large model shows the best performance overall. Interestingly, it shows the highest error for the Track 4 metric even though it is the best model for all other metrics. In contrast to our expectation, the SwinV2 Large model is not outperforming the other models for any metric.

Backbone	F1 Score	Track 1	Track 2	Track 3	Track 4
Swin Base	55.80	0.344	0.180	0.544	2.705
Swin Large	56.05	0.339	0.153	0.518	2.842
SwinV2 Base W24	55.92	0.343	0.160	0.527	2.701
SwinV2 Large W24	55.23	0.343	0.170	0.534	2.771

track 2 and track 3 which are showing an optimal score for thresholds slightly below 0.2. The F1 score and the track 1 score are optimized for a threshold between 0.3 and 0.5. The track 4 score is showing a drastically different behavior with a minimal error reached at a confidence threshold of 1.0. Thus, the minimal error corresponds to not executing any classification at all but to reject all samples and mark them as unknown. Therefore, optimizing the track 4 error is only sensible when another metric is optimized in parallel. While we report the best score for each metric but track 4 in the following experiments, for the track 4 score, we report the result for the confidence threshold with the highest F1 score.

4.6. Backbone architectures

Based on the results of Wolf and Beyerer [9], we start with evaluating Swin Transformer [11] models. Additionally, we evaluate a Swin Transformer V2 [10] models. The results as shown in Table 1 indicate that the Swin Transformer V2 Base model outperforms the Swin Transformer counterpart by a slight margin. However, it can not outweigh the advantage of the higher capacity of the Swin Transformer Large model which performs best overall. Just for the track 4 error, the Swin Transformer Large model shows the worst results in this comparison. In contrast to our expectation, the SwinV2 Large model does not perform best. In fact, it does not outperform the other models for any metric. The reasons for the lack of performance needs to be figured out in future research. More generalization like data augmentation or stochastic depth might be required. Other possible reasons might be inappropriate hyper parameter selection for generalization techniques like the label smoothing value.

4.7. Window size for SwinV2

Since one extension of Swin Transformer V2 is an increased window size, we evaluate this parameter separately. While Swin Transformer employs a window size of 7x7 for all stages, Swin Transformer V2 halves the window size in the last stage. We report the window size for the first three stages. Swin Transformer V2 is commonly pretrained on ImageNet-21k with a window size of 12x12 and an image resolution of 192x192. We employ the same pretraining but we increase the image resolution to 384x384 for fine-tuning on the Danish Fungi dataset in order to pick up the fine details of the fungi important for distinguishing the species. For the

Table 2
Comparison of the window size for SwinV2 Base. A larger window size shows a significant increase in terms of F1 score and a reduction of the track 4 error. However, all other metrics are either equal or worse with a larger window size.

Window Size	F1 Score	Track 1	Track 2	Track 3	Track 4
12	54.72	0.345	0.154	0.529	2.774
24	55.92	0.343	0.160	0.527	2.701

Table 3
Evaluating the class balancing strategy. While the class balancing improves the F1 score and the track 4 error, the advantage for other metrics is unclear.

Class balancing	F1 Score	Track 1	Track 2	Track 3	Track 4
No	55.59	0.344	0.157	0.527	2.787
Yes	55.80	0.344	0.180	0.544	2.705

Table 4
Evaluating the impact of removing the margin of the center crop in the image pre-processing. A center crop is still applied to provide a square image. The results show that no advantage can be gained by removing the commonly applied margin for the center crop even though it might be reasonable for large fungi spanning the whole image.

Margin	F1 Score	Track 1	Track 2	Track 3	Track 4
Yes	55.92	0.343	0.160	0.527	2.701
No	55.56	0.342	0.170	0.535	2.712

first experiments, we keep the window size at 12x12 and afterwards, we evaluate increasing it to 24x24. The results are shown in Table 2. Increasing the window size improves the F1 score and the track 4 error by a significant margin. Nonetheless, the track 1 and 3 errors show no significant change and the error of track 2 shows a slight increase. So, overall the choice of window size depends on the metric on which the user is focused.

4.8. Class-balancing

To cope with the high class imbalance of the Danish Fungi dataset, we employ a simple class balancing scheme based on resampling. We evaluate the impact of the class balancing with a Swin Transformer Base model. We show the results in Table 3. The results are mixed with class balancing providing an advantage for the F1 score and the track 4 error while having no significant impact on the error for track 1 and deteriorating the performance on track 2 and 3.

4.9. Margin for center crop

For image classification, a center crop is usually applied to the input image before classification. For coarse-grained image classification, this provides an advantage due to the important parts

Table 5
Evaluating multiple models and configurations on the test set. In general, the results indicate an advantage of the Swin Large model over the Swin Base model and Swin V2 over Swin. The impact of the margin of the cencter crop and the confidence threshold depends on the observed metric. Similar to the validation set, a high confidence threshold drastically improves the track 4 error while hurting all other metrics.

Backbone	Threshold	Epochs	Margin	F1	Track 1	Track 2	Track 3	Track 4
Swin Base	0.0	6	Yes	52.53	0.417	0.194	0.611	3.626
Swin Base	0.2	6	Yes	54.91	0.369	0.202	0.571	3.095
Swin Base	0.3	6	Yes	54.74	0.360	0.230	0.590	2.920
Swin Large	0.2	6	No	55.21	0.365	0.191	**0.556**	3.064
Swin Large	0.2	6	Yes	55.27	0.367	0.197	0.564	3.071
Swin Large	0.5	6	No	54.67	**0.347**	0.316	0.663	2.611
Swin Large	0.85	6	No	44.92	0.384	0.822	1.206	**1.905**
SwinV2 Base	0.08	9	Yes	54.97	0.367	0.191	0.558	3.077
SwinV2 Base	0.1	6	Yes	**55.31**	0.366	**0.190**	**0.556**	3.047

of the image mostly being centered while the background may contain distracting cues that might still be recognized due to the wide range of classes. However, for fine-grained fungi classification, the relevant fungus might not be centered if it spans a large area and distracting objects in the background are rather unlikely due to the focus of fungi during training. Thus, we evaluate removing the margin of the center crop. To keep the aspect ratio of the image intact, we still apply a center crop but only for the longer side of the image. The results are shown in Table 4. While the results for both evaluations are quite close, they show a slight decrease for all metrics but the track 1 error which shows no significant change. So, performing a center crop with margin is also advantageous for fine-grained fungi classification. Likely, a large part of the images are showing mushrooms which are not spanning the whole image and thus, performing a center crop highlights the important part of the image containing a mushroom and provides a higher resolution.

4.10. Results on test set

A set of models were selected as final submissions and evaluated on the competitions' private test set. The models were chosen to provide a wide variety of settings instead of only submitting the models performing best on the validation set. Particularly, a wide variety of confidence thresholds is chosen. All models were trained with class balancing. The results are shown in Table 5. The best model and configuration heavily depends on the observed metric. For the F1 score and the track 2 and 3 errors, a SwinV2 Base model with a low confidence threshold performs best. For the track 1 and 4 errors, a Swin Large model with higher confidence thresholds performs better. However, evaluating a SwinV2 Base model with higher confidence thresholds might lead to even better results.

5. Conclusion

We presented a novel method for classification of fine-grained fungi species in an open-set scenario. The methods outperform previous methods by exploiting Swin Transformer V2 as modern feature extractor backbone at a similar model size. Our investigations show the difficulty of optimizing multiple metrics in parallel with the optimal design decision often being dependent on the considered metric. This insight emphasizes the importance of deep investigations considering multiple metrics and scenarios and the significance of proper requirements engineering for shifting from research to production. Future work might include the exploitation of metadata or more sophisticated fusion schemes of the images of an observation.

Acknowledgments

The authors acknowledge support by the state of Baden-Württemberg through bwHPC.

References

[1] L. Picek, M. Šulc, R. Chamidullin, J. Matas, Overview of fungiclef 2023: Fungi recognition beyond 1/0 cost, in: CLEF 2023-Conference and Labs of the Evaluation Forum, 2023.

[2] A. Joly, H. Goëau, S. Kahl, L. Picek, C. Botella, D. Marcos, M. Šulc, M. Hrúz, T. Lorieul, S. S. Moussi, M. Servajean, B. Kellenberger, E. Cole, A. Durso, H. Glotin, R. Planqué, W.-P. Vellinga, H. Klinck, T. Denton, I. Eggel, P. Bonnet, H. Müller, Lifeclef 2023 teaser: Species identification and prediction challenges, in: J. Kamps, L. Goeuriot, F. Crestani, M. Maistro, H. Joho, B. Davis, C. Gurrin, U. Kruschwitz, A. Caputo (Eds.), Advances in Information Retrieval, Springer Nature Switzerland, Cham, 2023, pp. 568–576.

[3] A. Joly, C. Botella, L. Picek, S. Kahl, H. Goëau, B. Deneu, D. Marcos, J. Estopinan, R. Chamidullin, M. Šulc, M. Hrúz, M. Servajean, B. Kellenberger, E. Cole, H. Glotin, Overview of lifeclef 2023: evaluation of ai models for the identification and prediction of birds, plants, snakes and fungi, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction: 14th International Conference of the CLEF Association, CLEF 2023, Thessaloniki, Greece, September 18–23, 2023, Proceedings, Springer, 2023.

[4] B. Zieliński, A. Sroka-Oleksiak, D. Rymarczyk, A. Piekarczyk, M. Brzychczy-Włoch, Deep learning approach to describe and classify fungi microscopic images, PloS one 15 (2020) e0234806.

[5] M. Sulc, L. Picek, J. Matas, T. Jeppesen, J. Heilmann-Clausen, Fungi recognition: A practical use case, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 2316–2324.

[6] 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.

[7] N. Kiss, L. Czùni, Mushroom image classification with cnns: A case-study of different learning strategies, in: 2021 12th International Symposium on Image and Signal Processing and Analysis (ISPA), IEEE, 2021, pp. 165–170.

[8] 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.

[9] S. Wolf, J. Beyerer, Transformer-based fine-grained fungi classification in an open-set scenario, Working Notes of CLEF (2022).

[10] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, F. Wei, B. Guo, Swin transformer v2: Scaling up capacity and resolution, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 12009–12019.

[11] 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, 2021, pp. 10012–10022.

[12] A. Gupta, P. Dollar, R. Girshick, Lvis: A dataset for large vocabulary instance segmentation, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 5356–5364.

[13] M. Contributors, Openmmlab's pre-training toolbox and benchmark, https://github.com/open-mmlab/mmpretrain, 2023.

[14] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Advances in neural information processing systems 32 (2019).

[15] 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.

[16] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, arXiv preprint arXiv:1711.05101 (2017).

[17] E. D. Cubuk, B. Zoph, J. Shlens, Q. V. Le, Randaugment: Practical automated data augmentation with a reduced search space, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 702–703.

[18] 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.

[19] 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.