Transformer-based Fine-Grained Fungi Classification in an Open-Set Scenario

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 classification describes the task of estimating the species of a fungus. The FungiCLEF 2022 challenge started a competition for the best solution to solve this task in an open-set scenario. For our solution, we employ a modern transformer-based classification architecture, use a class-balanced training scheme to handle the class-imbalance and apply heavy data augmentation. We approach the open-set scenario by using the final confidence scores as an indicator for unknown species. With this classification model, we were able to achieve an F1 score of 80.6 and 77.5 on the challenge's public and private test set, respectively. This resulted in achieving the 7th place in the FungiCLEF 2022 challenge. We provide code at https://github.com/wolfstefan/fungi-classification.

Keywords

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

1. Introduction

Accurately estimating the species of fungi is an important task for mushroom picker in order to identify toxic fungi. However, this task is difficult to learn due to the fine details distinguishing different species and the high number of species. Thus, an automatic system for classification of fungi species can save lives. Nonetheless, a classifier has to cope with similar challenges like the small differences distinguishing classes. Therefore, the classification model requires careful design choices. We design a classification model that is targeted towards this use case while supporting open-set classification tasks by recognizing images showing unknown fungi species. We design the classification model with simplicity in mind and thus, refrain from using ensembles of multiple models. Our model is based on a Swin Transformer Large [1] backbone, a class-balancing training scheme [2], heavy data augmentation and a recognition of unknown species based on softmax scores.

With this classification model, we participated in the FungiCLEF 2022 [3] challenge - part of the LifeCLEF [4, 5] lab - and reached the 7th place. The task of the FungiCLEF 2022 challenge is to classify the species of fungi based on the Danish Fungi 2020 [6] datset as training set and the

CLEF 2022: Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy
stefan.wolf@iosb.fraunhofer.de (S. Wolf)

CEUR Workshop Proceedings (CEUR-WS.org)

Danish Fungi 2021 [3] dataset as test set in an open-set scenario. Thus, the classifier must be additionally able to recognize species not occurring in the Danish Fungi 2020 dataset.

In Section 2, we provide an overview over existing literature in the field of fine-grained fungi classification. In Section 3, we describe the method we use for this classification task. In Section 4, the experiments leading to several design decisions are shown. In Section 5, we conclude our contributions and indicate what future research directions might be.

2. Related work

Image-based classification of fungi species has been studied in a small number of publications which are described in this section. Zieliński et al. [7] explored the classification of fungi species using microscopic images with an approach based on deep learning and bag-of-words. Sulc et al. [8] proposed a system for fungi classification based on an ensemble of convolutional neural networks. Picek et al. [6] published the fine-grained fungi dataset called Danish Fungi 2020 which contains metadata like location, habitat, substrate and date of the observation as possible inputs for classification systems in addition to images. Moreover, the authors evaluated multiple classification architectures based on convolutional neural networks and vision transformers with latter have proven to be superior. Additionally, the benefit of the metadata for the classification is highlighted with a significant increase in accuracy. Kiss and Czùni [9] proposed a mushroom dataset and explored different strategies of learning convolutional neural networks for an optimal classification accuracy of mushroom types.

3. Method

In this section, the base model used for classification and the different adjustments to improve the accuracy of fine-grained fungi classification are described.

Deep learning model. We employ a Swin Transformer Large [1] model as the classification backbone to extract features. Swin Transformer is based on multi-head self-attention [10]. The input is grouped as windows and the multi-head self-attention is applied separately for each window to appropriately exploit the local relations of pixels in images. These windows are shifted after each layer to enable the use of cross-window relations. Similar to ResNet-like [11] architectures, the model is partitioned in stages with each stage reducing the resolution to create hierarchical feature maps.

Class-balancing. To cope with the imbalanced nature of the Danish Fungi 2020 [6] dataset, we apply the data resampling scheme proposed by Gupta et al. [2]. While this scheme was originally proposed for object detection, the application for single-label classification is straightforward. In the original implementation, a single image can increase the counter for multiple categories since multiple categories can occur in a single image in object detection tasks. For single-label image classification, the category-level counter is increased only for the single category of an image. We use a value of 10^{-2} as oversample threshold.

Training strategy. The weights of the model are initialized from a model pretrained on

ImageNet-21K [12]. AdamW [13] is employed as optimizer with a base learning rate of $7 \cdot 10^{-5}$, a weight decay of 0.05 and betas of 0.9 and 0.999. A cosine annealing schedule [14] is applied to reduce the learning rate over the course of the training. A learning rate warm-up is applied for the first 4200 iterations with the initial learning rate being 1% of the base learning rate. Due to the class-balancing heavily increasing the size of a single epoch, the total length of the training is just 6 epochs. The batch size per GPU is 12 with 6 GPUs being used for the training resulting in a total batch size of 48. A label smoothing loss is applied [15].

Data augmentation. We use a broad set of augmentation techniques to improve the results. First, a crop is applied that extracts a slice of the image with the size of the slice randomly chosen between 8% and 100% of the original image. Afterwards, the slice is resized to 384×384. The is image is flipped horizontally with a probability of 50%. Moreover, RandAugment [16] is used with the default settings of MMClassification [17] for training Swin Transformer. This contains geometrical transformations like adjustments of brightness and geometrical operations like shearing. The last augmentation used is a random erase [18] that masks out a region of a size between 2% and 1 / 3 of the cropped image with a probability of 25%. The last step of the pre-processing pipeline is an image normalization. In contrast to the default data augmentation settings of Swin Transformer, we do not apply Mixup [19] or Cutmix [20] since they do not seem appropriate to us for fine-grained classification and preliminary experiments showed a negative impact.

During inference, the image is scaled with the smaller side being 438 pixels and the resolution of the larger side being chosen so that the original ratio is kept. Afterwards, a crop with the size of 384×384 pixels is taken from the center of the image and an image normalization is applied.

Data. The training for the final challenge model is performed on the training and the validation set of the Danish Fungi 2020 [6] dataset.

Multi-view classification. The Danish Fungi 2020 [6] dataset used for training and validation and the Danish Fungi 2021 set used as test set for the FungiCLEF 2022 challenge contain multiple images per observation for many observations. For the sake of simplicity, we train the image in a single-view manner with each image treated as an individual sample. During inference on the test set of the challenge, the features of all images for a single observation are averaged channel-wise before the application of the final fully-connected classification layer. Additionally, we create a new image for each existing image by flipping it horizontally. These images are treated as additional views and the results are aggregated in the described way.

Open-set classification. The FungiCLEF 2022 challenge is using an open-set task with the Danish Fungi 2021 [3] set containing observations of fungi species without any observation in the the Danish Fungi 2020 [6] dataset. Thus, observations of unknown classes have to be marked as such. We mark all observations whose highest predicted score for a class after the application of the softmax is below 0.1 as an observation with an unknown class.

4. Experiments

Our final model described in Section 3 is the result of multi experiments which are described in this section. For our ablation studies, the training is performed only on the training subset of the Danish Fungi 2020 [6] dataset. Four metrics are reported as results. The *val top-1* and the *val F1* are the top-1 accuracy and the macro F1 score on the validation subset of the Danish Fungi 2020 [6] dataset, respectively. The results on the validation set are per image and not per observation. Thus, no multi-view consensus is applied for these results. The *test public F1* and the *test private F1* are the official challenge results on the Danish Fungi 2021 [3] set which are reported as public and private score by Kaggle, respectively. These are macro F1 scores on a per observation base with the consensus method described in Section 3.

All models are trained and evaluated with MMClassification [17] (version 0.23) which is a classification framework based on PyTorch [21] (version 1.8.2). To reduce the training time required, we train the models with FP16 and dynamic loss scaling enabled.

Contrary to the final model, some experiments use a target resolution of 224×224 pixels. In these cases, the resolution prior to the cropping in the inference pipelines is 256 for the shorter edge. All experiments until *Recognizing unknown species* ignore the open-set scenario and predict one of the known classes for every observation.

Datasets. The Danish Fungi 2020 [6] dataset contains a total of 295,938 images. Of these, 266,344 are used for training and 29,594 are used for validation. The images are taken from 1,604 different fungi species. The Danish Fungi 2021 [3] test set contains a total of 118,675 images from 59,420 different observations. While both datasets provide metadata as e.g. time and location, our solution does not make use of them to keep the complexity of the approach low.

Multi-view consensus. In case of post-classifier-consensus, the whole network is executed for each image independently and the per-class scores of the final fully-connected layer are aggregated by multiplication. The pre-classifier-consensus executes the network up to the last layer before the final fully-connected layer and averages the feature vectors of all images for a single observation. Afterwards, the average feature vector is fed into the final fully-connected layer to predict the class scores of the observation. The experiment is done with a ResNet-50 [11] model, a target resolution of 224 pixels, no label-smoothing, no class-balanced training, a total batch size of 256 spread across 2 GPUs and a learning rate of 10^{-4}. The results are shown in Table 1. No results are reported for the validation set since we performed the evaluation on the validation set image-wise instead of observation-wise. The results show a slight advantage for the pre-classifier-consensus mode. The pre-classifier-consensus mode is used for every further experiment.

Class-balanced training. We evaluate the class-balanced training strategy as described in Section 3. The number of epochs is reduced to compensate the higher number of iterations per epoch due to the class-balanced training. The evaluation is done with a ResNet-50 backbone trained on two GPUs with a batch size of 128 per GPU, a base learning rate of 10^{-4} and without label smoothing. The target resolution is 224×224 for these experiments. A lower oversample

Table 1
Results of two different multi-view consensus modes. Due to the slight advantage of the pre-classifier-consensus mode, we choose this one for the final model.

Consensus type	Test public F1	Test private F1
Post-classifier-consensus	66.6	63.6
Pre-classifier-consensus	66.7	63.6

Table 2
Results of class-balanced training with different oversample thresholds. Since a smaller oversample thresholds results in larger number of iterations per epoch, we reduce the number of epochs. The results indicate no significant difference for the F1 score and a reduced top-1 score on the validation set. However, results on both parts of test set show an improvement. Thus, we apply the class-balancing with an increased oversample threshold of 10^{-2} for the final model.

Class-balancing	Oversample threshold	Iterations per epoch	Epochs	Val top-1	Val F1	Test public F1	Test private F1
No	-	1041	100	75.0	67.3	66.7	63.6
Yes	10^{-3}	1626	100	74.8	67.4	66.5	62.9
Yes	10^{-2}	4215	25	74.5	67.3	68.1	63.8

thresholds leads to a stronger oversampling of underrepresented classes. According to the results, the impact on the F1 score is not significant and the top-1 score is reduced since the model cannot exploited the class-imbalance anymore. However, the results on the challenge test set showed an improvement which lead to the decision to apply the class-balancing with an oversample threshold of 10^{-2} on the final model.

Choosing the base model and the resolution. We evaluate multiple deep learning backbones and two different resolutions for the classification. The models evaluated are ResNet-50 [11], Swin Base and Swin Large. The experiments are either done with a target resolution of 224×224 pixels or with the final resolution of 384×384 pixels as described in Section 3. The experiments are run with label smoothing, class-balanced training, a batch size of 16 per GPU and 4 GPUs. The base learning rate is $6 \cdot 10^{-5}$. The results are shown in Table 3. The results show a significant improvement from ResNet-50 to Swin Base and also a slight improvement from Swin Base to Swin Large. The largest improvement can be gained by switching from a resolution of 224 to a resolution of 384. We reduced the number of epochs for most experiments to save compute resources. While this leads to a significantly reduced accuracy for ResNet-50, a longer training schedule provides only a slight advantage for the Swin Large model. Since this advantage is nonexistent for the results on the test set, we apply the shorter schedule with 6 epochs for the final model.

Test-time augmentation. To evaluate the test-time augmentation using horizontal flip, we train a Swin Large model on the training and validation set with the final settings as described in Section 3 and only provide the results for the test set. The results are shown in Table 4. The results indicate a slight benefit for the combination of processing the original image and the

Table 3
Results of different backbones and different resolutions. To reduce the training time, we decrease the number of epochs for the ablation studies and increase it again for the final model. The results indicate a clear advantage of the newer Swin Transformer models over the ResNet-50 model and also an advantage of the higher resolution.

Backbone	Target resolution	Epochs	Val top-1	Val F1	Test public F1	Test private F1
ResNet-50	224	25	72.3	64.4	68.4	64.5
ResNet-50	224	6	62.7	53.5	61.8	57.3
Swin Base	224	6	79.5	73.3	75.2	71.7
Swin Base	384	6	85.0	79.7	79.2	75.5
Swin Large	384	6	85.9	81.3	79.2	75.8
Swin Large	384	12	87.0	82.3	79.1	76.0

Table 4
Results of horizontal flip as test-time augmentation. Processing both the original image and the horizontal image and combining both results lead to a slight increase in accuracy for the test set.

Test-time augmentation	Test public F1	Test private F1
None	79.6	76.5
Horizontal flip	79.8	76.6

Table 5
Results of different score thresholds. All observations below the threshold are marked as ones with an unknown species. For comparison, the total number of observations in both sets is 59420. While a threshold of 0.2 has shown the best accuracy after the end of the challenge, we chose 0.1 for our final model.

Threshold for unknown species	# unknown observation	Test public F1	Test private F1
0.05	554	80.3	77.1
0.1	1308	80.6	77.5
0.2	2651	80.7	77.7

flipped image.

Recognizing unknown species. As described in Section 3, we mark an observation as one with an unknown species if the highest score is below a certain threshold. In Table 5, the results for different thresholds are shown. The best F1 score was reached with a threshold of 0.2. However, due to a limited number of evaluations possible prior to the end of the challenge, we found this threshold to better only after the end of the challenge and chose a threshold of 0.1 for the final model. With this threshold, 1308 of the 59420 observations are marked as ones with an unknown species.

5. Conclusion

We proposed a classifier targeted towards fine-grained fungi classification with the support of open-set scenarios. The classifier is based on a modern Swin Transformer Large backbone generating highly expressive features and a multi-view aggregation step to exploit the availability of multiple images per observation. To improve the accuracy, we make use of heavy data augmentation and apply a class-balancing training scheme. Future directions for research might be the incorporation of metadata to aid the classification or a more advanced aggregation of the results from the different images of an observation.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[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] 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.

[8] 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.

[9] 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.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polo-

sukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[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] 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.

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

[14] I. Loshchilov, F. Hutter, Sgdr: Stochastic gradient descent with warm restarts, arXiv preprint arXiv:1608.03983 (2016).

[15] 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.

[16] 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.

[17] M. Contributors, Openmmlab's image classification toolbox and benchmark, https://github.com/open-mmlab/mmclassification, 2020.

[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] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, in: International Conference on Learning Representations, 2018.

[20] 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.

[21] 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).