Few-Shot Fungi Classification with Prototypical Networks Using Multiple Pretrained Embedding Models

Notebook for the Etheredge Lab at CLEF 2025

Jack N. Etheredge[1,*]

[1]Twosense, New York, New York, United States

Abstract

The FungiCLEF 2025 challenge encourages improvement on few-shot fine-grained classification with large scale (2,427 classes). This represents a real-world application with a dataset of rare species in Danish Fungi. In this paper, we present our approach to the challenge, which aims to classify images of fungi given few example images per species. This method utilizes pretrained embedding models *DINOv2*, *BEIT*, and *SAM*. Simple image augmentations are applied at both train and test time. Embeddings from each model are concatenated into a single embedding along the feature dimension per augmented version of the image. A simple projection network was trained to improve the discriminative performance of the embeddings on the training samples. Cosine similarity between the class centroid and the observation centroid is used for class prediction, as in Prototypical Networks. Finally, an ensemble of these pipelines is utilized to further boost performance. Image augmentation is shown to be the largest contributor to the performance of the solution, followed by learning an embedding projection, and utilizing multiple embedding models. Our method secured 1st place in the FungiCLEF 2025 competition on the private leaderboard. Code is available at https://github.com/Jack-Etheredge/fungiclef2025.

Keywords

Embedding, FungiCLEF, Fungi Classification, Few-shot, FungiTastic

1. Introduction

FungiCLEF 2025 [1] is a competition held as part of the LifeCLEF 2025 [2] lab [1]. FungiCLEF 2025 is a fine-grained few-shot classification task. This represents a real-world scenario as described in the study associated with the FungiTastic dataset [3]. Namely, the distribution of fungi species observed in the parent FungiTastic dataset is long-tailed and there are many rare fungi species, which means that these rare species must be considered for few-shot learning, treated as unknown species, or otherwise excluded from the dataset. The FungiCLEF 2025 challenge dataset represents this long tail of the FungiTastic dataset. This work describes the top-ranked solution to the competition. The primary contributions of this work, in order of their impact on the final performance according to ablation studies, are to 1) supplement prototypical networks with geometric augmentations of the images at both training and test time, 2) learn a projection of the embedding, 3) use multiple pretrained embedding models instead of a single model, and 4) ensemble multiple of these prototypical network embedding pipelines.

2. Related work

FungiCLEF 2025 is a few-shot fine-grained image classification task. It has 2,427 classes, which is many more than most few-shot benchmarks. Even in cases where there are more classes available, most performance benchmarks report the 5-way performance on a fraction of a larger dataset (e.g.,

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain

*Corresponding author.

jack.etheredge@gmail.com (J. N. Etheredge)

https://github.com/Jack-Etheredge (J. N. Etheredge)

0000-0001-5467-3866 (J. N. Etheredge)

[1]https://www.imageclef.org/LifeCLEF2025

5-way, 5-shot). ImageNet [4] (21,841 classes, but most commonly used with 1,000 classes), Omniglot [5] (1,623 classes), Meta-Dataset [6] (which comprises 10 datasets including ImageNet and Omniglot), and iNaturalist [7] with 5,089 classes are some of the only other example datasets that are commonly used for large-scale few-shot image classification with over 1,000 classes.

Last year's FungiCLEF 2024 challenge [8] focused on open-set recognition and minimizing confusion between poisonous and edible species. The average number of training and validation images per class were comparatively much greater, with 1,604 known species and 1,629 unknown species represented across a combined 222,191 observations with 387,169 total instances. The training set for FungiCLEF 2024 was from Danish Fungi 2020 [9], while the validation set was collected from 2022.

3. Methodology

3.1. Dataset

The FungiCLEF 2025 challenge [1] tasked participants with classifying fungi species from images. The dataset is created from images and metadata submitted to the Atlas of Danish Fungi before the end of 2023. Each species label was assigned by mycologists. The challenge dataset is drawn from the few-shot dataset from [3], which describes the dataset in depth.

An observation refers to a real-world occurrence of fungi, which may include, but is not limited to, an individual mushroom, a cluster of mushrooms, or mold growing on a surface, either in a natural environment or as a collected sample. Each observation comprises one or more instances. An instance is an individual data point associated with an observation and consists of an image, its associated metadata, and a generated caption. For example, an individual mushroom may constitute an observation, but multiple images of this mushroom might be captured from different angles. Each of these images (along with its metadata and caption) would represent a distinct instance linked to the same observation. The solution proposed in this paper only utilizes the images, since initial experiments with captions and metadata were not promising (data not shown).

The dataset contained 2,427 classes with 5,392 observations comprising 10,104 instances between the training and validation sets. Most classes have a single observation and most observations have a single instance. All classes had fewer than 5 observations. Combining the training and validation sets into a single dataset, the class with the most instances has 39 instances. Though not as extreme as the parent FungiTastic dataset, the challenge dataset still exhibits severe class imbalance, with most classes having only a single observation while the largest class by instance count has 39 instances, creating a long-tailed distribution.

3.2. Competition objective and evaluation metrics

The objective of FungiCLEF 2025 was to achieve the best average performance predicting the class of each test observation given one or more instances per observation. The public and private leaderboards for the competition both used average recall at rank $k = 5$ (recall@5), which we refer to as *Top-5 accuracy* or simply *Top-5* hereafter.

For each test observation x, let y_x denote its true class label, and $\{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_5\}$ be the top 5 predicted classes. The recall@5 for observation x is defined as:

$$\text{recall@5}(x) = \begin{cases} 1 & \text{if } y_x \in \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_5\} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The average recall@5 over the entire test set $\mathcal{T}$ is then computed as:

$$\text{Average Recall@5} = \frac{1}{|\mathcal{T}|} \sum_{x \in \mathcal{T}} \text{recall@5}(x). \tag{2}$$

3.3. Overall solution architecture

The overall solution is illustrated in Figure 1. Figure 1A shows that training and test time differ only by the level of hierarchy that the embeddings are averaged to. For creating the prototype embeddings, all augmented versions of images belonging to each class are averaged together. Since the competition expects observation-level predictions, all instances belonging to each test observation and all augmentations of the instance images are averaged into a single embedding. For training of the projection network, all the augmented versions of the training images are used with their class labels as targets. Predictions are made by calculating the cosine similarity between the class-level prototype embeddings and the observation-level test embeddings. Hereafter, this series of functions to transform a collection of training images into prototype embeddings and test images into observation embeddings to produce class-wise cosine similarities through the use of a specific combination of image augmentations, frozen embedding models, and a trained projection network will be referred to as an embedding pipeline. Figure 1B shows how multiple of these pipelines are combined into an ensemble using the softmax probability of the cosine similarities. The only difference between members of the ensemble is the random initialization of the projection network and the random training-validation split at the instance level for training the projection network. Figure 1C shows how embeddings from multiple models are generated for each augmented image. The embedding from each model is normalized to unit length through L2 normalization. Per augmented image, these normalized embeddings are concatenated along the feature dimension. The four models used were: BEiT-Base/p16 [10] trained on the FungiTastic dataset [3], DINOv2-base [11], DINOv2-large [11], and Segment anything model (SAM) ViT Huge (ViT-H) [12].

Prototype embeddings were computed as the mean class embeddings, averaged over all augmentations, instances, and observations from all the provided data (both the training and validation datasets). To generate predictions, embeddings were averaged over all augmentations and instances in the test observation.

Let $f_\phi(\cdot)$ be an embedding function, defined as the concatenation of multiple frozen embedding model outputs along the embedding dimension, followed by a projection via a multilayer perceptron, and parameterized by ϕ. Let $\mathcal{S}_c$ denote the support set for class c. The class prototype $\mathbf{p}_c \in \mathbb{R}^d$ is defined as:

$$\mathbf{p}_c = \frac{1}{|\mathcal{S}_c|} \sum_{x_i \in \mathcal{S}_c} f_\phi(x_i) \tag{3}$$

Each observation x consists of a set of images $\mathcal{I}_x = \{x_1, x_2, \ldots, x_n\}$, and each image x_i has a set of augmentations $\mathcal{A}(x_i) = \{x_i^{(0)}, x_i^{(1)}, \ldots, x_i^{(k_i)}\}$, where $x_i^{(0)}$ is the original image. Let N_x denote the total number of augmented images in the observation:

$$N_x = \sum_{x_i \in \mathcal{I}_x} |\mathcal{A}(x_i)| \tag{4}$$

The final observation embedding $\mathbf{z}_x \in \mathbb{R}^d$ is computed as the mean of all augmented instance image embeddings belonging to the observation:

$$\mathbf{z}_x = \frac{1}{N_x} \sum_{x_i \in \mathcal{I}_x} \sum_{x_i^{(j)} \in \mathcal{A}(x_i)} f_\phi(x_i^{(j)}) \tag{5}$$

Per embedding pipeline, the embeddings used for the prototype embeddings and the test observation embeddings were projected using the same trained projection network. An ensemble of 5 embedding pipelines was used to generate the final predictions. These embedding pipelines differed only by the training-validation split and initialization of the projection network. The validation portion was used for early stopping during training of the projection network. The softmax probabilities over the classes for each model were generated from the cosine similarities between each test observation and the prototype embedding for each class. The ensemble average softmax probability of the cosine similarities was used to rank the classes per observation. The top 10 classes were returned per observation as was

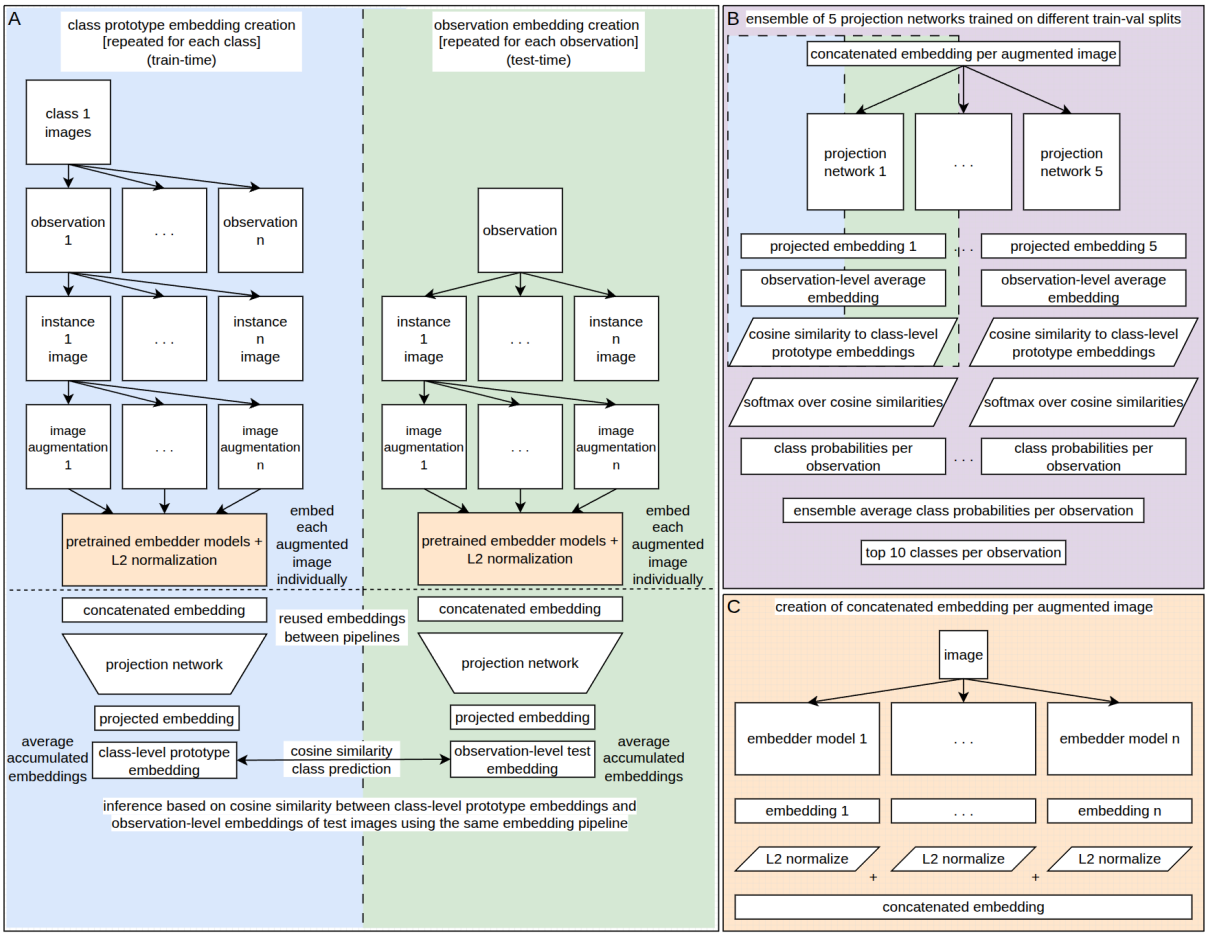

Figure 1: Solution overview. (A) Augmented image embeddings are used to train a projection network used to create prototype embeddings and observation embeddings. Predictions are made based on the cosine similarity between the prototype embeddings and the observation embeddings. (B) Ensemble of multiple embedding pipelines. Softmax is used to generate class probabilities from the cosine similarities for each pipeline. The probabilities from each pipeline are averaged to create the final class probabilities for each observation. (C) Creation of the embeddings from each augmented image. Multiple embedding models are used to create embeddings of augmented images. These embeddings are normalized to unit length and then concatenated.

expected of the participants. Only the first 5 of these 10 classes factored into the leaderboard ranking, however, since the competition evaluation metric was the recall at top-5.

3.4. Image augmentation

The same image augmentations were performed for both the training samples and test time augmentations. This was done both for simplicity and also to maximize agreement between the prototypes and the test embeddings. Only geometric augmentations were used in the winning solution. The specific augmentations utilized were: 80% center crop, 80% top left crop, 80% top right crop, 80% bottom left crop, 80% bottom right crop, horizontal flip, 90-degree rotation, 270-degree rotation, 15-degree rotation, and 345-degree rotation.

3.5. Embedding models

All experiments were performed on a machine with a single NVIDIA RTX 3090 graphics card and all models were trained using PyTorch [13]. Embeddings were generated from augmented images using pretrained models. A simple two-layer network was trained to project the embeddings from these models into a new embedding space as described in the follow section, but the pretrained models were

Table 1
Learned loss weight values. Weights explore widely during training but converge to stable values. Final epoch values show mean ± standard deviation across 5 seeds. Full training range shows min-max values across all epochs and seeds; final epoch range shows min-max values at the end of training across seeds.

Loss Weight	Final Epoch Value	Full Training Range	Final Epoch Range
cross-entropy	1.178 ± 0.041	0.849–1.220	1.120–1.220
InfoNCE	1.627 ± 0.057	1.001–1.687	1.549–1.687

not fine-tuned.

For all models, after the geometric augmentations were performed, the augmented image was resized with bicubic interpolation to 1.14x the final image size used for that model and then center cropped to the final image size. 1.14 was taken from the widely adopted practice of resizing to 256 before taking a square crop of 224. This is common in ImageNet [4] pre-processing and can be seen in AlexNet [14].

The final image sizes used are as follows:

- BEiT-Base/p16: 384^2
- DINOv2-Base: 434^2
- DINOv2-Large: 518^2
- SAM-ViT-Huge: 1024^2

3.6. Projection network training

A two-layer network was trained to project the concatenated embeddings into a new embedding that better discriminated between the classes. Using the labels for each augmented image per instance, the network was trained using PyTorch to project the concatenated embeddings into an embedding with dimensionality of 768. The model consists of an input layer mapped to a hidden layer with dimensionality 2048, followed by an output layer with dimensionality 768. Both layers are fully connected, with ReLU activation after the first layer. A batch size of 64 was used. The *AdamW* optimizer [15] was used with a learning rate of 1e-4 and a weight decay of 1e-4. Early stopping was used with a patience of 5 along with a random validation split. Training was stopped when the projection model validation loss did not improve for 5 consecutive epochs and the weights with the best validation loss were restored. The model was trained with cross-entropy and infoNCE [16] with temperature of 0.07. The infoNCE implementation was used from [17]. The per-class probability for cross-entropy was determined based on the softmax of the cosine similarity. The balance between the cross-entropy and infoNCE losses was determined through two additional learned loss weighting parameters as in [18]. Across 5 random seeds, we report the final learned weights immediately before early stopping was triggered, as well as the range of values both weights explored during training (Table 1). These results indicate that while both weights are learned dynamically, they converge to stable values with modest variation across seeds. We observe a consistent upward trend in the InfoNCE weight over training, while the cross-entropy weight first decreases and then increases again over the course of training. The mean projection network wall-clock training time for 5 seeds was 294 seconds. For an ensemble, this scales linearly with the number of pipelines.

3.7. Embedding pipeline ensemble

Multiple embedding pipelines are combined into an ensemble for the final predictions. For each embedding pipeline in the ensemble, the softmax of the cosine similarities between the prototype embedding for each class and the test embedding were calculated. The softmax probabilities per embedding pipeline in the ensemble were then averaged to get the final class probabilities. The mean inference wall-clock time for 5 seeds was 7.83 milliseconds per observation. For an ensemble, this scales linearly with the number of pipelines unless inference is performed in parallel.

Table 2
Augmentation ablations. Geometric augmentations are critical to the performance of proposed solution. Interestingly, the inclusion of 90-degree or 270-degree rotation alone performs nearly as well as the combination of the 10 augmentations used. Results show Top-5 mean and standard deviation over 5 random seeds.

Original	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Center crop	✓	–	✓	–	–	–	–	–	–	–	–	–
Top left crop	✓	–	–	✓	–	–	–	–	–	–	–	–
Top right crop	✓	–	–	–	✓	–	–	–	–	–	–	–
Bottom left crop	✓	–	–	–	–	✓	–	–	–	–	–	–
Bottom right crop	✓	–	–	–	–	–	✓	–	–	–	–	–
Horizontal flip	✓	–	–	–	–	–	–	✓	–	–	–	–
Rot 90	✓	–	–	–	–	–	–	–	✓	–	–	–
Rot 270	✓	–	–	–	–	–	–	–	–	✓	–	–
Rot 15	✓	–	–	–	–	–	–	–	–	–	✓	–
Rot 345	✓	–	–	–	–	–	–	–	–	–	–	✓
Top-5 Mean	**63.5**	36.4	61.7	62.2	62.1	61.7	61.2	61.7	62.7	62.8	62.0	61.9
Top-5 Std Dev	**0.6**	0.7	0.9	0.5	0.7	0.6	0.0	0.5	0.6	1.0	0.7	0.4
Δ	-	-27.1	-1.8	-1.3	-1.4	-1.8	-2.3	-1.8	-0.8	-0.7	-1.5	-1.6

Table 3
Impact of test-time augmentations. Test-time augmentations (TTA) are extremely important for model performance. Their removal causes a 42% reduction in the *Top-5 accuracy*. Results show mean ± standard deviation over 5 random seeds.

TTA	Top-5	Δ
True	**63.5 ± 0.6**	-
False	37.1 ± 0.6	-26.4

4. Evaluation performance and ablation studies

The solution described in this study achieved 1st place on the private leaderboard for FungiCLEF 2025. This section details the results of ablations for the various components of the solution described in the previous section. For ablation experiments, models were trained using a split of the official training set into new training and validation subsets (used for early stopping), and evaluated on the official validation set (treated as a test set). Unless explicitly stated otherwise (e.g., Table 8 showing the private leaderboard performance for the top teams), all results are reported on this official validation set. The baseline for each of these ablations is a single embedding pipeline (instead of the final ensemble) with the same seed for the training-validation split and projection network initialization. All ablation experiments use deterministic seeding as described in Section 4.4.

4.1. Image augmentation

As shown in Table 2 and Table 3, the inclusion of train and test time augmentations are the largest contributors to the performance of this solution. The inclusion of train time augmentations without test time augmentations results in a *Top-5 accuracy* reduction of 26.4 percentage points while the removal of both train time and test time augmentations results in a reduction of 27.1 percentage points.

4.2. Learned Projection

Learning a projection of the concatenated embeddings improves model performance as shown in Table 4. The projection networks utilized by our top-ranking solution were trained with a combination of cross-entropy and infoNCE losses. Table 5 shows that this combined loss outperforms either loss alone.

Table 4
Impact of learned projection. The removal of the projection network causes a large reduction in the *Top-5 accuracy*. Results show mean ± standard deviation over 5 random seeds.

Projection	Top-5	Δ
True	**63.5 ± 0.6**	-
False	54.6 ± 0.0	-8.9

Table 5
Impact of projection network loss function choice. A learned weighted combination of cross-entropy and infoNCE outperforms either loss function alone. Results show mean ± standard deviation over 5 random seeds.

Loss	Top-5	Δ
combined cross-entropy + InfoNCE	**63.5 ± 0.6**	-
cross-entropy	61.4 ± 0.4	-2.1
InfoNCE	61.9 ± 1.0	-1.6

Table 6
Model ablations. The combination of multiple models achieves better performance than any single model. SAM-ViT-H does not perform well alone and its removal from the model combination does not appear to degrade performance. Note: In all cases, a projection of the embedding is learned as in the overall solution. Results show mean ± standard deviation over 5 random seeds.

FungiTastic-BEiT-Base/p16	✓	✓	✓	–	–	–
DINOv2-Base	✓	✓	–	✓	–	–
DINOv2-Large	✓	✓	–	–	✓	–
SAM-ViT-H	✓	–	–	–	–	✓
Top-5	63.5 ± 0.6	**63.6 ± 0.8**	58.4 ± 0.4	57.6 ± 0.6	59.8 ± 0.6	12.4 ± 0.4
Δ	-	+0.1	-5.1	-5.9	-3.7	-51.1

4.3. Combining multiple embedding models

Both combining models at the feature level and also ensembling predictions from multiple learned projections of those embeddings improve performance. Table 6 shows that concatenating the embeddings from multiple pretrained models outperforms using the embedding from a single pretrained model. DINOv2-Large proves to be a particularly strong performer as a single model. Conversely, SAM-ViT-H performs quite poorly without the context of the other embedding models. It appears that SAM-ViT-H can be removed from the embedding model combination to decrease the computational demands of the solution without degrading performance.

Table 7 shows that an ensemble of embedding pipelines outperforms a single embedding pipeline. As previously described, each member of the ensemble differed only by the training-validation split used to train the projection model and the random initialization of the projection model. For this ensemble, the seed for the training-validation split and the projection network initialization were different for each member of the ensemble, since otherwise predictions from the ensemble would be identical to that of a single pipeline.

4.4. Seeding and Replicability

To ensure reproducibility and statistical robustness, all ablation experiments used deterministic seeding. Each configuration was run with 5 independent replicates, and we report mean ± standard deviation.

Seeds were computed hierarchically as

$$s_{r,m} = 1000 \cdot r + m, \tag{6}$$

Table 7
Impact of embedding pipeline ensemble. An ensemble of 5 pipelines outperforms a single pipeline. Results show mean ± standard deviation over 5 random seeds.

Ensemble	Top-5	Δ
5x	**64.7 ± 0.2**	+1.2
No	63.5 ± 0.6	-

Table 8
Private leaderboard performance for top 10 teams.

Rank	TeamName	Top-5
1	Jack Etheredge (ours)	**78.9**
2	hard_work	78.1
3	aixiaodeyanjing	76.6
4	hahahahahal	76.2
5	skhhhh	75.3
6	hahahalll	75.2
7	aurora_aur_	74.4
8	zhangchao111	73.9
9	Hasan Oetken	73.9
10	team	73.4

where $r \in \{0, 1, 2, 3, 4\}$ indexes the experimental replicate and $m \in \{0, 1, \ldots, M-1\}$ indexes the ensemble member ($m = 0$ for single pipelines, and M is the ensemble size). This structure ensures non-overlapping seeds across replicates and ensemble members while maintaining reproducibility.

For single pipelines, the same seed was used for both the training-validation split and the projection model initialization. In ensembles, each member differed only by its corresponding seed, ensuring diversity through variation in both data splits and projection model initializations.

4.5. Leaderboard performance

Private leaderboard performance for the top 10 ranking teams is shown in Table 8. Our models achieved the best performance for the competition metric (*Top-5 accuracy*).

5. Conclusions

Simple methods are sufficient to achieve state-of-the-art performance for few-shot classification of fungi from image data. In this study, we described our winning approach for the FungiCLEF 2025 challenge. Using pretrained image classification and feature extraction networks, embeddings can be cached and subsequently used to train lightweight projection networks. These networks can be ensembled to further boost performance. Concatenation of embeddings from multiple frozen embedding models and averaging embeddings from multiple image augmentations perform well despite their simplicity. Importantly, we show that test-time augmentation is critical to the performance of this method.

Declaration on Generative AI

During the preparation of this work, the author used Anthropic Claude Sonnet 4 in order to: Paraphrase and reword. After using this tool/service, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] K. Janouskova, J. Matas, L. Picek, Overview of FungiCLEF 2025: Few-shot classification with rare fungi species, in: Working Notes of CLEF 2025 - Conference and Labs of the Evaluation Forum, 2025.

[2] L. Picek, S. Kahl, H. Goëau, L. Adam, T. Larcher, C. Leblanc, M. Servajean, K. Janoušková, J. Matas, V. Čermák, K. Papafitsoros, R. Planqué, W.-P. Vellinga, H. Klinck, T. Denton, J. S. Cañas, G. Martellucci, F. Vinatier, P. Bonnet, A. Joly, Overview of lifeclef 2025: Challenges on species presence prediction and identification, and individual animal identification, in: International Conference of the Cross-Language Evaluation Forum for European Languages (CLEF), Springer, 2025.

[3] L. Picek, K. Janouskova, V. Cermak, J. Matas, Fungitastic: A multi-modal dataset and benchmark for image categorization, 2025. URL: https://arxiv.org/abs/2408.13632. arXiv:2408.13632.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255. URL: https://ieeexplore.ieee.org/document/5206848. doi:10.1109/CVPR.2009.5206848, iSSN: 1063-6919.

[5] B. M. Lake, R. Salakhutdinov, J. B. Tenenbaum, Human-level concept learning through probabilistic program induction, Science 350 (2015) 1332–1338.

[6] E. Triantafillou, T. Zhu, V. Dumoulin, P. Lamblin, U. Evci, K. Xu, R. Goroshin, C. Gelada, K. Swersky, P.-A. Manzagol, H. Larochelle, Meta-dataset: A dataset of datasets for learning to learn from few examples, 2020. URL: https://arxiv.org/abs/1903.03096. arXiv:1903.03096.

[7] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, S. Belongie, The inaturalist species classification and detection dataset, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[8] L. Picek, M. Sulc, J. Matas, Overview of FungiCLEF 2024: Revisiting fungi species recognition beyond 0-1 cost, in: Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, 2024.

[9] 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, in: 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2022, pp. 3281–3291. URL: http://arxiv.org/abs/2103.10107. doi:10.1109/WACV51458.2022.00334, arXiv:2103.10107 [cs, eess].

[10] H. Bao, L. Dong, S. Piao, F. Wei, Beit: Bert pre-training of image transformers, 2022. URL: https://arxiv.org/abs/2106.08254. arXiv:2106.08254.

[11] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, R. Howes, P.-Y. Huang, H. Xu, V. Sharma, S.-W. Li, W. Galuba, M. Rabbat, M. Assran, N. Ballas, G. Synnaeve, I. Misra, H. Jegou, J. Mairal, P. Labatut, A. Joulin, P. Bojanowski, Dinov2: Learning robust visual features without supervision, 2023.

[12] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, R. Girshick, Segment anything, 2023. URL: https://arxiv.org/abs/2304.02643. arXiv:2304.02643.

[13] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An Imperative Style, High-Performance Deep Learning Library, in: Advances in Neural Information Processing Systems, volume 32, Curran Associates, Inc., 2019. URL: https://papers.nips.cc/paper_files/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

[14] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: F. Pereira, C. Burges, L. Bottou, K. Weinberger (Eds.), Advances in Neural Information Processing Systems, volume 25, Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[15] I. Loshchilov, F. Hutter, Decoupled Weight Decay Regularization, 2019. URL: http://arxiv.org/abs/1711.05101. doi:10.48550/arXiv.1711.05101, arXiv:1711.05101 [cs, math].

[16] A. van den Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, 2019. URL: https://arxiv.org/abs/1807.03748. arXiv:1807.03748.
[17] K. Musgrave, S. J. Belongie, S.-N. Lim, Pytorch metric learning, ArXiv abs/2008.09164 (2020).
[18] A. Kendall, Y. Gal, R. Cipolla, Multi-task learning using uncertainty to weigh losses for scene geometry and semantics, 2018. URL: https://arxiv.org/abs/1705.07115. arXiv:1705.07115.