Transfer Learning and Mixup for Fine-Grained Few-Shot Fungi Classification

Jason Kahei Tam[1,*], Murilo Gustineli[1,*] and Anthony Miyaguchi[1,*]

[1]Georgia Institute of Technology, North Ave NW, Atlanta, GA 30332

Abstract

Accurate identification of fungi species presents a unique challenge in computer vision due to fine-grained inter-species variation and high intra-species variation. This paper presents our approach for the FungiCLEF 2025 competition, which focuses on few-shot fine-grained visual categorization (FGVC) using the FungiTastic Few-Shot dataset. Our team (DS@GT) experimented with multiple vision transformer models, data augmentation, weighted sampling, and incorporating textual information. We also explored generative AI models for zero-shot classification using structured prompting but found them to significantly underperform relative to vision-based models. Our final model outperformed both competition baselines and highlighted the effectiveness of domain-specific pretraining and balanced sampling strategies. Our approach ranked 35/74 on the private test set in post-completion evaluation, this suggests additional work can be done on metadata selection and domain-adapted multi-modal learning. Our code is available at https://github.com/dsgt-arc/fungiclef-2025.

Keywords

LifeCLEF, FungiCLEF, Fine-Grained Visual Categorization (FGVC), Vision Transformers, fungi, species identification, machine learning, computer vision, CEUR-WS

1. Introduction

Accurate identification of fungi species is challenging and often requires microscopic examination [1], which places this task beyond simple image classification and falls into fine-grained visual categorization (FGVC). This paper documents and evaluates our approaches as part of the FungiCLEF 2025 [2] @ CVPR-FGVC & LifeCLEF 2025 [3] competition. The goal of this competition is fungi species recognition and the evaluation metric is top-5 accuracy.

Multiple aspects make this task challenging. First, there is intra-species variation due to factors such as genotype, local conditions, time of year, and age (Figure 1) [4]. Second, there is subtle inter-class variation even at higher order taxonomic ranks, which makes it difficult for pattern learning algorithms to differentiate (Figure 2) [4].

Figure 1: Example of intra-species variation - Species: Comatricha alta

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

*Corresponding author.

jtam30@gatech.edu (J. K. Tam); murilogustineli@gatech.edu (M. Gustineli); acmiyaguchi@gatech.edu (A. Miyaguchi)

0009-0005-0853-0414 (J. K. Tam); 0009-0003-9818-496X (M. Gustineli); 0000-0002-9165-8718 (A. Miyaguchi)

Figure 2: Example of inter-species similarity and variation from three distinct families. Species left to right: Psathyrella citerinii, Inocybe assimilata, and Entoloma favrei

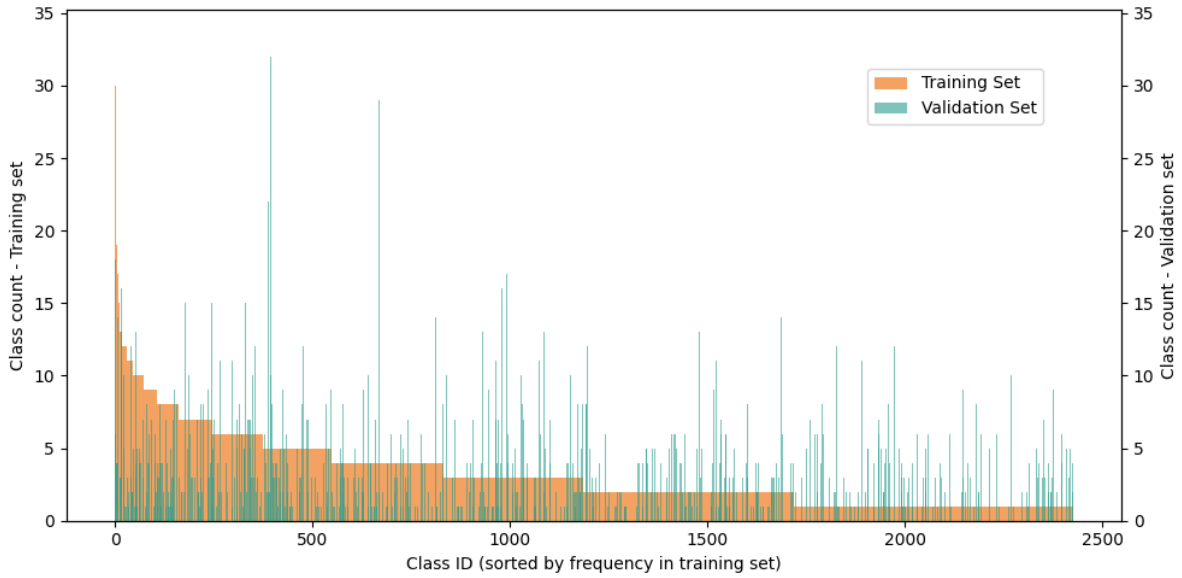

Figure 3: Distribution of classes in training and validation datasets

1.1. Dataset Overview

The dataset provided for the competition is the few-shot subset of the FungiTastic dataset, a collection of fungal records continuously collected over a twenty-year span [5]. Each observation in the dataset contains associated images, metadata, and vision language model (VLM), Molmo [6], generated caption. The metadata contains information such as date, location, substrate, and full taxonomic ranks. It is important to note that the task is to classify images based on category_id, which has a slightly different count than species. The training dataset contains 7,819 images with 2,413 unique species and 2,427 unique category_id. The validation dataset contains 2,285 images with 569 species and 570 unique category_id. The test dataset contains 1,911 images with no taxonomic ranks. The provided image dataset contains the training, validation, and testing sub-datasets. Each sub-dataset contains images in different maximum pixel sizes, ranging from 300p to full-size images.

The datasets do not have the same category_id distribution (Figure 3). In the chart, the category_id is mapped to class ID and then sorted by frequency, with category_id 2383 appearing most frequently. Both datasets exhibit class imbalance, with the most common class having approximately 30 images and multiple classes having only 1 image.

2. Related Work

Previous work by the DS@GT group for FungiCLEF 2024 demonstrated the strong performance of using DINOv2 vision transformers in image classification [7]. Last year's winner, Team IES, combined image embeddings from Swin Transformer V2 [8] with metadata features from multi-layer-perception for species classification [9].

3. Methodology

Our benchmark approach uses PlantCLEF 2024 [10] embeddings, weighted sampling [11], and Mixup [12]. Using off-the-shelf generative AI models, multi-modal approach of combining text embeddings with image embeddings, and multi-objective loss were also explored. The competition evaluation metric is top-k accuracy, with k = 5.

$$\text{Top-k Accuracy} = \frac{1}{N}\sum_{i=1}^{N} 1\left[y_i \in \hat{Y}_i^{(k)}\right] \tag{1}$$

The cloud computing resources were funded by the Data Science at Georgia Tech (DS@GT). Data and computing was hosted by the Partnership for an Advanced Computing Environment (PACE) at Georgia Tech [13].

3.1. Benchmark Methodology

Our benchmark methodology can be summarized in a few steps:

1. Image embeddings from PlantCLEF 2024 model
2. Weighted sampling to balance the training dataset
3. Mixup on batches during training
4. Linear classifier
5. Mixup loss with cross-entropy [14]

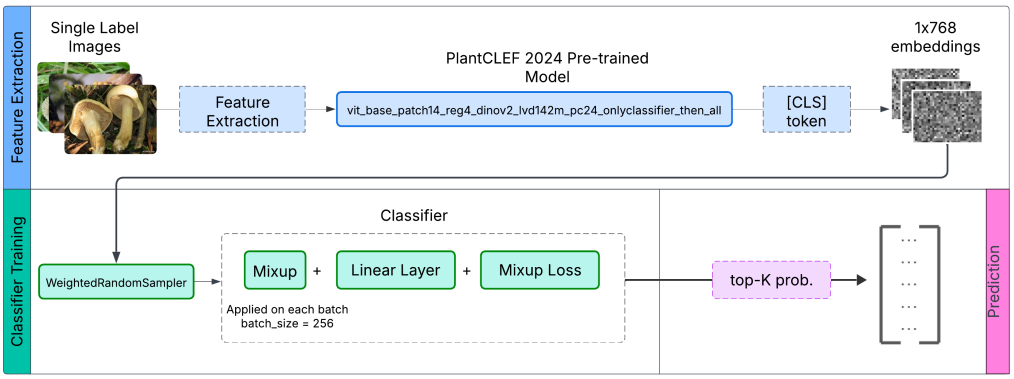

Figure 4: Pipeline of the benchmark methodology

3.1.1. Dataset Preparation

We pre-computed both the image and text embeddings and stored them in parquet files for a modular experimentation workflow. We encountered "Premature End of JPEG file" error when reading images because some images did not end with the default hex code. This may result in unintended side effects during training. This error was solved by loading the image with OpenCV and then saving it again [15].

There was one corrupted image in the validation 720p set; we did not use this image since we only used the full-size images for our pipeline.

All approaches except Generative AI involved using the pytorch lightning library in training the classifier. The hyperparameters used are as follows: batch size of 256, 50 maximum epochs with early stopping of 3, Adam [16] optimizer, learning rate of $5 \cdot 10^{-4}$ and no learning rate scheduler.

3.1.2. Image Embeddings

We experimented with multiple transformer models: Facebook DINOv2 [17], PlantCLEF 2024 pre-trained model[10], FungiTastic BEiT [5], and FungiTastic ViT[5]. A summary of the models used is shown in Table 1 and Table 2.

DINOv2 was selected for its state-of-the-art performance in computer vision tasks and its strong results in FungiCLEF 2024[7]. The PlantCLEF model was selected due to its foundation in DINOv2 and its pre-training on 1.4 million plant images from the Pl@ntNet database, offering the potential benefits of transfer learning. The two FungiTastic models were selected because they were pre-trained on the fungi images.

The image embeddings from the PlantCLEF 2024 model used in our benchmark methodology have a size of 768.

Table 1
Summary of models explored for image embeddings generation

Shorthand	Model Name
DINOv2	dinov2-base [17]
PlantCLEF 2024	vit_base_patch14_reg4_dinov2_lvd142m_pc24_onlyclassifier_then_all [10]
FungiTastic BEiT	beit_base_patch16_224.in1k_ft_fungitastic_224[5]
FungiTastic ViT	vit_base_patch16_224.in1k_ft_fungitastic_224 [5]

Table 2
Details of models explored for image embeddings generation

Shorthand	Architecture	Hidden Size	Pretraining/Finetuning
DINOv2	ViT-B/14 [17]	768	LVD-142M/None
PlantCLEF 2024	ViT-B/14 [17]	768	DINOv2 on LVD-142M/Pl@ntNet
FungiTastic BEiT	BEiT-B/16[18]	768	ImageNet-1k/FungiTastic
FungiTastic ViT	ViT-B/16 [19]	768	ImageNet-1k/FungiTastic

3.1.3. Weighted Sampling and Mixup

To mitigate the effects of class imbalance observed in Figure 3, we experimented with the PyTorch's `WeightedRandomSampler` [11] using weights calculated with inverse class frequency via `compute_sample_weight` from the `sklearn` library, on the training dataset. This sampling strategy was implemented in the data loader to ensure that minority classes were sampled more frequently during training.

We also experimented with Mixup [12] to increase the influence of minority classes. Mixup was implemented in the classifier and applied to batches provided by the data loader. Mixup encourages the classifier model to generalize better by interpolating features and labels between classes. In our implementation, the embeddings extracted from the training dataset were linearly combined with a shuffled version to generate an augmented set using the equations:

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j \tag{2}$$

$$\mathcal{L}_{\text{Mixup}} = \lambda \cdot \mathcal{L}(f(\tilde{x}), y_i) + (1 - \lambda) \cdot \mathcal{L}(f(\tilde{x}), y_j) \tag{3}$$

where $\lambda \sim \text{Beta}(\alpha, \alpha)$, x denotes image embeddings, y denotes the label targets, i indexes the original mini-batch, and j indexes a randomly shuffled version of the same mini-batch. In the competition approach, $\alpha = 2.0$, 256 batch size, and 10 epochs were used to evaluate the impact of Mixup. An $\alpha = 2.0$ was inspired by Manifold Mixup [20] to encourage an greater generalization due to the small dataset.

In our post-competition evaluation, we increased the epochs used in the Mixup [12] only approach to 50 to make its results more comparable with the other approaches discussed in Section 4 Results as all other approaches used 50 max epochs. Here, we evaluated the results using α values ranging from [0.1, 2.0], which encompasses the recommended ranges from the Mixup [12] paper and the Manifold Mixup [20] paper (Figure 5). An $\alpha = 1.20$ and $\alpha = 1.45$ achieved the highest public score and these two values were used to run additional experiments with a finetuned Mixup and weighted sampling.

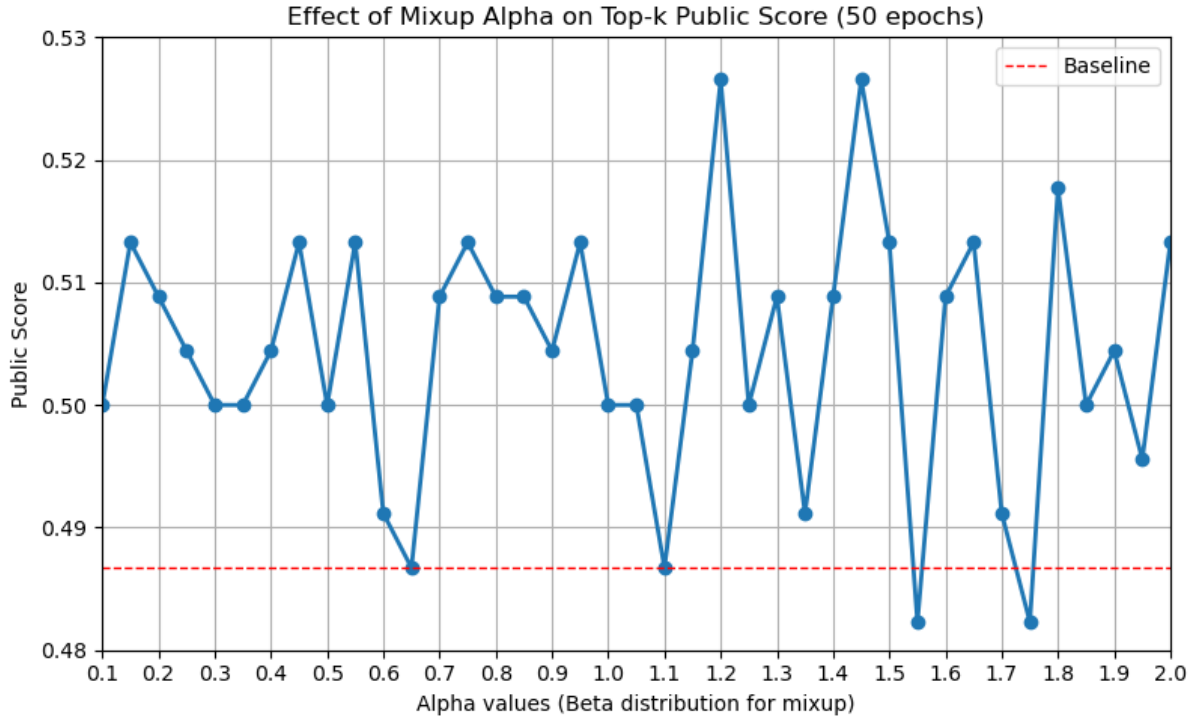

Figure 5: Effect of Mixup Alpha on Top-K Accuracy

3.2. Additional Methodologies

3.2.1. Text Embeddings

We used ModernBERT-Large[21], a state-of-the-art BERT variant optimized for efficiency, to compute 1024-dimensional text embeddings. We concatenated text from categories present in the test metadata file with the generated captions to form a single string. The results were saved in a parquet file.

In post-competition evaluation, we also used BioBERT-Large [22], a domain specific BERT based model pre-trained on biomedical corpora to compute 1024-dimensional text embeddings. There is a potential for transfer learning between biomedical texts and fungi textual information since both fall under the domain of biology. Again, we concatenated text from categories present in the test metadata file with the generated captions to form a single string.

In a multi-modal classifier, the image and text embeddings are fed through their own linear layers with the same output size of 256. The image and text embeddings are then concatenated, normalized, and fed into a 512 input size linear classification layer.

3.2.2. Multi-Objective Loss GradNorm

Inspired by the evaluation metrics used in FungiCLEF 2024[7], we experimented with a multi-objective classification framework to jointly predict category_id, poisonous, genus, and species. Each objective has its own classification head and loss function: cross-entropy loss for category_id, genus, and species, and binary cross-entropy loss for poisonous. To prevent a single objective from dominating classification and to encourage balanced learning, we implemented GradNorm [23]. GradNorm allows dynamically assigning weights for each objective when calculating loss. We introduced a learnable weight for each objective and computed the gradient norms with respect to the shared parameters.

3.2.3. Generative AI

We explored the use of generative AI techniques to predict species in the dataset. Many commercially available multi-modal large language models are vision-language models, where vision and language modalities are fused through an attention mechanism. We implement a zero-shot prompting method across three API providers using the OpenRouter platform and leverage structured output to enforce the structural regularity of the results.

Table 3
Models used for experiments via OpenRouter.

Model Name	Release Date	Context (tokens)	Input ($/M)	Output ($/M)	Vision Input ($/K images)
google/gemini-2.0-flash-001	2025-02-05	1,048,576	$0.10	$0.40	$0.026
openai/gpt-4.1-mini-2025-04-14	2025-04-14	1,047,576	$0.40	$1.60	N/A
google/gemini-2.5-flash-preview-04-17	2025-04-17	1,048,576	$0.15	$0.60	$0.619
mistralai/mistral-medium-3	2025-05-07	131,072	$0.40	$2.00	N/A

We perform three rounds of prompting across family, genus, and species per test image to logarithmically reduce the search space and ensure that only species within the training set are used. Each round of prompting relies on the prompt used in listing 3.2.3. We append a yaml list of all the candidate items to rank. We append all available images for a single image ID (which can range from one to a dozen images) as context to the completion. We request a list of 20 ranked candidates, including an item name and a corresponding confidence score. The results are validated against the candidate list and accepted if at least half of the results are valid, i.e., there exists an item that is within 90% of the string by normalized edit distance. For human debugging purposes, we also have the LLM generate a reason for the decision.

```
Accurately identify and assign the correct {class_type} label to each image of
fungi, protozoa, or chromista utilizing all provided image views and associated
metadata (location, substrate, season) to ensure precision, especially for
fine-grained distinctions. Choose the top twenty most relevant labels ranked in
order from the available class labels, a confidence on the Likert scale between
1-5 on not-confident to confident and provide short reasoning (in under 50
words) for your selection.
```

In the first round of prompting, we provide a list of all families and ask the LLM to rank the top 20 species relevant to the test images. We use the most relevant families to generate a candidate list of genera. We then use this to generate a candidate list of species. We provide the top 10 species as the final result of the competition.

4. Results

4.1. Image Embeddings Results

The best performing models to pre-compute the image embeddings were the PlantCLEF 2024 [10] model and the FungiTastic ViT [5] model. The embeddings from each model were passed into a linear layer to generate the predictions. The top-5 accuracy public score is then used to select the model to use in our benchmark methodology (Table 4). PlantCLEF 2024 [10] was selected as our baseline classifier and incorporated into our best performing approach.

Table 4
Performance of models used for image embeddings

	Top-5 Acc., Public (%)	Difference (%)
PlantCLEF 2024	**48.672**	-
DINOv2	47.345	-1.327
FungiTastic BEiT	42.477	-6.195
FungiTastic ViT	**48.672**	0

4.2. Ablation Study

The results from our various approaches are compiled in Table 5. Our best in-competition approach was with Mixup [12] $\alpha = 2.0$ and weighted sampling with a private top-5 accuracy score of 40.75. In post-competition evaluation, a finetuned $\alpha = 1.20$ achieved the highest private score when combined with weighted sampling and $\alpha = 1.45$ achieved the highest private score when used by itself.

We found that Mixup with a tuned α is the single technique with the greatest positive impact with an increase of 4.27% on the private score. Weighted sampling provided a much smaller increase in accuracy on its own and had minimal effect when combined with a tuned α. Lastly, we find that incorporating metadata + caption and a multi-objective GradNorm [23] approach of classifying category_id, poisonous, species, and genus had a negative impact on the prediction accuracy.

Table 5
Ablation study of our different approaches

Classification method	**Top-5 Acc.**	
	Public (%)	**Private (%)**
PlantCLEF 2024 Model Baseline	48.672	43.078
w/ weighted sampling	50.442	44.372
w/ Mixup (α = 2.00)* (Competition)	46.460	40.750
w/ Mixup (α = 1.20) (Post-Comp)	52.654	44.760
w/ Mixup (α = 1.45) (Post-Comp)	52.654	**47.347**
w/ Mixup (α = 2.00) & weighted sampling (Competition)	**49.557**	**45.407**
w/ Mixup (α = 1.20) & weighted sampling (Post-Comp)	50.884	**46.830**
w/ Mixup (α = 1.45) & weighted sampling (Post-Comp)	**53.982**	46.054
w/ text (ModernBert) (Competition)	46.460	38.421
w/ text (BioBert) (Post-Competition)	45.132	40.620
w/ GradNorm and weighted sampling	42.920	39.197
w/ text, GradNorm, and weighted sampling	39.823	36.093
Gemini 2.0 Flash	14.159	12.548
Gemini 2.5 Flash	11.504	13.583
OpenAI GPT 4.1 Mini	5.309	6.209
MistralAI Mistral Medium 3	4.424	3.104

* Trained with 10 epochs. All other classifier approaches are trained with 50 epochs

4.3. Leaderboard Results

Our team's result beat both competition baselines: BioCLIP [24] + FAISS + Prototypes and BioCLIP + FAISS + NN. However, our result falls short of the leaders in the competition. We are ranked 37/74 on the public leaderboard and 35/74 on the private leaderboard. These results are summarized in Table 6

Table 6
Public and Private test set scores of the leaderboard

	Name	Top-5 Acc. (%)
Public	PlantCLEF + Mixup (α = 2.00) + Weighted Sampling (Competition)	49.557
Private	PlantCLEF + Mixup α = 2.00) + Weighted Sampling (Competition)	45.407
Public	PlantCLEF + Mixup (α = 1.20) + Weighted Sampling (Post-Comp)	50.884
Private	PlantCLEF + Mixup (α = 1.20) + Weighted Sampling (Post-Comp)	46.830
Public	Rank 1 - Jack Etheredge	**81.858**
Private	Rank 1 - Jack Etheredge	**78.913**
Public	Rank 2 - Hasan Oetken	80.530
Private	Rank 2 - hard_work	78.137
Public	BioCLIP + FAISS + Prototypes Baseline	33.185
Private	BioCLIP + FAISS + Prototypes Baseline	26.649
Public	BioCLIP + FAISS + NN Baseline	28.318
Private	BioCLIP + FAISS + NN Baseline	24.708

4.4. Validation Dataset Performance

In Figure 6, we plot the class frequency versus the top-5 accuracy on the validation dataset. The class frequency is calculated on a per image basis, and not a per observation basis. There can be multiple images under an observation. The concentration of points at accuracy 1.0 and 0.0 (shown as darker points) at the rarer classes shows that the classifier often achieves perfect or zero Top-5 accuracy due to the small sample size. This highlights the volatility in the classification accuracy in class imbalanced datasets.

5. Discussion

5.1. Weighted Sampling and Mixup

As seen in Table 5, Mixup [12] with a tuned $\alpha = 1.20$ and $\alpha = 1.45$ had the greatest positive impact on our baseline accuracy. This is different from the recommended range of [0.1,0.4] for α suggested by the Mixup paper and could be because the Mixup is applied at the feature level instead at the raw inputs. Applying Mixup at the feature level is closer to the Manifold Mixup [20] approach which applies Mixup between intermediate layers of a neural network and uses $\alpha = 2.00$. From Figure 5, there is no clear monotonic trend as α changes, however all values of α except for two resulted in an improvement over the baseline. The lack of a monotonic trend may suggest adding more learnable layers to our classifier is needed to flatten class boundaries and reduce volatility as seen in Manifold Mixup [20].

Weighted sampling [11] was another approach that had a positive impact on our baseline accuracy, albeit to a smaller extent than a tuned Mixup. The modest increase in accuracy indicates that while it helps the model see rare classes more often, it alone does not sufficiently address the challenges of learning robust patterns for underrepresented classes. The mixed results when combined with Mixup shows that there is diminishing returns in applying multiple sampling approaches.

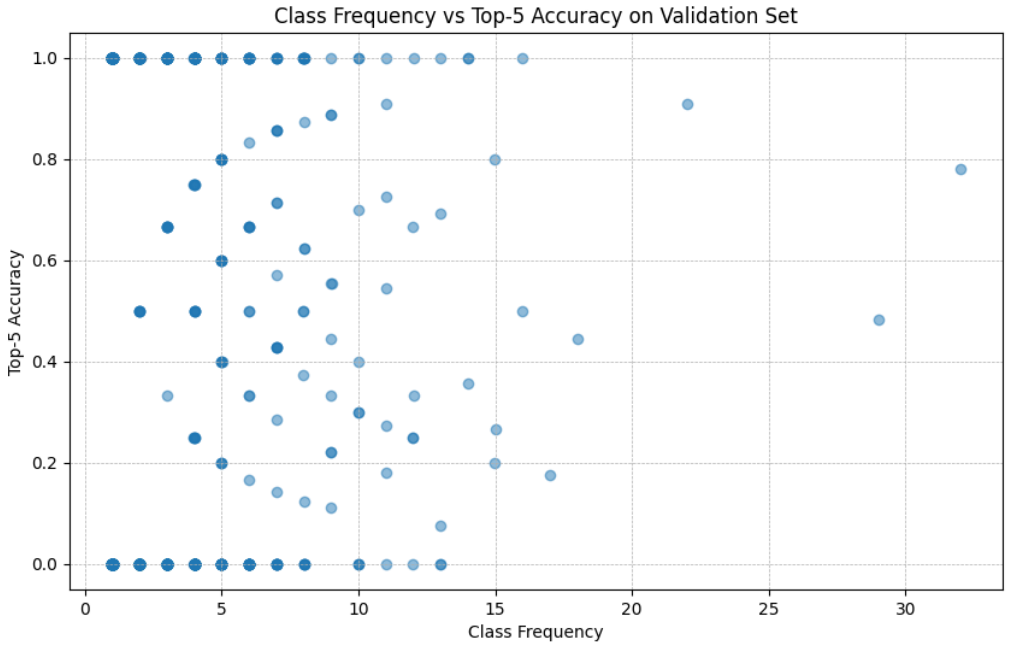

Figure 6: Class Frequency vs Top-5 Accuracy on Validation Set

5.2. Image Embeddings Generation

Among the different models evaluated for image embedding generation, the PlantCLEF 2024 and FungiTastic ViT models performed the best (Table 4. These two models slightly edged out general use DINOV2. Although the improvement is small, this suggests that domain-adapted models can offer an advantage over general use models in few-shot fine grained species classification. This finding is similar to the few-shot results presented in the FungiTastic paper [5], in which BioCLIP[24] outperformed DINOv2 and CLIP[25].

5.3. Text Embeddings

The inclusion of metadata and captions had a negative impact on our classifier performance. This is contrary to the findings presented in the FungiTastic paper [5], where the incorporation of metadata did improve performance. This discrepancy is likely due to our inclusion of extraneous or weakly informative metadata such as `district`, `countryCode`, and `hasCoordinate`.

5.4. Generative AI Results

Current-generation multi-modal LLMs are not effective at generalizing to the domain-specific task of labeling fungus images, at least within our price range. Our best model is from the Gemini family of models, scoring around 13% on the private leaderboard. Our choice of models is dictated by cost. For example, Gemini Pro is about 10 times as expensive as the Flash series of models. Gemini Flash was at a level of cost-effectiveness that we were willing to experiment with, and initial experimentation led us to hold off on trying models with higher token usage, which is generally associated with "thinking" or "reasoning" capabilities. We then chose GPT-4.1-mini and Mistral as models that had both structured output and image inputs. The set of models that accepted both of these constraints is much smaller than we would have liked and precluded models such as Anthropic Claude. We summarize the costs associated with this approach in table 7, which comes close to a total of $30 over 15k requests.

Table 7
Model Usage and Cost Summary.

Model Name	Total Tokens (M)	Total Requests (K)	Total Cost ($)
mistralai/mistral-medium-3	25.20	3.60	13.80
openai/gpt-4.1_mini-2025-04-14	28.10	3.04	12.70
google/gemini-2.5_flash-preview-04-17	7.84	3.65	2.29
google/gemini-2.0_flash-001	8.36	3.99	2.06

Note that while we limited models to structured outputs, we can simulate this in a two-pass methodology, where a stronger model generates results in a particular semi-structured shape, and a second, smaller but cheaper model converts this into a structured output via JSON Schema. However, this requires more boilerplate code and effort than we were willing to explore at this point, given the performance relative to stronger vision-first approaches. We also note that our three-round approach was necessary because there are limits to the structured schema API. For example, one of the first things we tried was to return a list of strings where a string must be part of a particular enumeration. However, enumerations are supported only up to a certain number of elements, which are undocumented, if supported at all. Another reason is that the context window significantly influences which elements are recalled from the list of available class elements. If we were to include all species in one big list, there is a good chance that not every species would be considered from that list due to limitations of context locality. This behavior is challenging to describe quantitatively due to the accelerated pace of development of these models in production and the associated cost of running experiments.

We also note a few limitations in our methodology. First, LLMs are strongly affected by the amount of stochasticity introduced at token generation time (i.e., temperature). As such, the runs of our algorithm will change significantly over time, making it challenging to reproduce our results exactly. However, there are two approaches to mitigate reproduction issues, given that the cost of a single Gemini test run is about $2. The first option is to lower the temperature of the model, which is often supported. Another approach is to run several iterations of a model and aggregate the final results. The ideal solution would take on a Monte-Carlo tree search flavor, where we would sample the top-k elements many times and produce some probabilistic taxonomic tree based on knowledge embedded in the LLM.

FungiCLEF is a domain-specific task that is relatively resource-poor relative to the general task of information recall from large pools of publicly available text. However, it is impressive that these models can get any results at all. It would be interesting to gain a deeper understanding of the vision-question-based capabilities of these models, perhaps by using a smaller subset that considers the general challenges of the fungi dataset while managing costs. What might make the most sense is fine-tuning a smaller VLM, such as Gemma [26], Phi [27], or Llama [28], on the FungiCLEF dataset and seeing whether these smaller models can be effectively tuned for domain-specific language queries.

6. Future Work

Improvements can be made to address to disparity in class distributions between the training and validation datasets as observed in Figure 3. One approach, seen in previous research, is to combine the provided datasets [7], then we can resplit them to have similar distributions. In addition, improvements could be made to the selection and processing of textual data. Rather than incorporating all available metadata fields, future work should prioritize informative features. As a starting point, we propose using the three metadata attributes highlighted in the FungiTastic paper [5], which have been shown to be effective in improving model performance. In addition, more learnable layers can be added to the classifier with Mixup [12] applied at a random layer to more closely follow the approach proposed in Manifold Mixup [20]. Continuing with our generative AI approach, one can experiment with costlier models, or implement a Monte-Carlo tree search as discussed in Section 5.4.

7. Conclusions

In this paper, we present our approach to tackle the challenge of FungiCLEF 2025 few-shot fine-grained visual classification (FGVC) using vision transformer embeddings. We explored a range of models, such as DINOv2, PlantCLEF 2024, and FungiTastic pre-trained models, ultimately selecting the PlantCLEF 2024 model for our benchmark approach due to its strong performance and transfer learning. To mitigate the class imbalance in the dataset, we implemented weighted sampling and Mixup, with Mixup providing the most significant performance gain. We also experimented with incorporating textual metadata and multi-objective learning GradNorm, but found these approaches to be detrimental, likely due to noisy or weakly informative inputs. Our final competition and post-competition classifiers outperformed both competition baselines and demonstrated the importance of domain-specific embeddings and balancing strategies. However, a significant performance gap with the leaders in the competition indicates the need for further exploration of alternate classifier architectures and improved metadata integration.

Acknowledgements

We thank the Data Science at Georgia Tech (DS@GT) CLEF competition group for their support. This research was supported in part through research cyberinfrastructure resources and services provided by the Partnership for an Advanced Computing Environment (PACE) at the Georgia Institute of Technology, Atlanta, Georgia, USA [13].

8. Declaration on Generative AI

During the preparation of this work, the authors used Chat-GPT-4 in order to: drafting content, Grammar and spelling check. After using these tool(s)/service(s), the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] R. Lücking, M. Aime, B. Robbertse, et al., Unambiguous identification of fungi: where do we stand and how accurate and precise is fungal dna barcoding?, IMA Fungus (2020).

[2] Klarka, picekl, FungiCLEF 2025 @ CVPR-FGVC & LifeCLEF, 2025. URL: https://kaggle.com/competitions/fungi-clef-2025.

[3] A. Joly, L. Picek, S. Kahl, H. Goëau, L. Adam, C. Botella, M. Servajean, D. Marcos, C. Leblanc, T. Larcher, J. Matas, K. Janoušková, V. Čermák, K. Papafitsoros, R. Planqué, W.-P. Vellinga, H. Klinck, T. Denton, P. Bonnet, H. Müller, Lifeclef 2025 teaser: Challenges on species presence prediction and identification, and individual animal identification, Advances in Information Retrieval 2025 (2025).

[4] L. Picek, M. Šulc, J. Matas, Overview of Fungiclef 2024: Revisiting fungi species recognition beyond 0–1 cost, CLEF 2024 Working Notes CEUR-WS (2024).

[5] L. Picek, K. Janoušková, V. Cermak, J. Matas, FungiTastic: A multi-modal dataset and benchmark for image categorization, arXiv:2408.13632 (2025).

[6] M. Deitke, C. Clark, S. Lee, R. Tripathi, Y. Yang, J. S. Park, M. Salehi, N. Muennighoff, K. Lo, L. Soldaini, J. Lu, T. Anderson, E. Bransom, K. Ehsani, H. Ngo, Y. Chen, A. Patel, M. Yatskar, C. Callison-Burch, A. Head, R. Hendrix, F. Bastani, E. VanderBilt, N. Lambert, Y. Chou, A. Chheda, J. Sparks, S. Skjonsberg, M. Schmitz, A. Sarnat, B. Bischoff, P. Walsh, C. Newell, P. Wolters, T. Gupta, K.-H. Zeng, J. Borchardt, D. Groeneveld, C. Nam, S. Lebrecht, C. Wittlif, C. Schoenick, O. Michel, R. Krishna, L. Weihs, N. A. Smith, H. Hajishirzi, R. Girshick, A. Farhadi, A. Kembhavi, Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models, arXiv preprint arXiv:2409.17146 (2024).

[7] C. Chiu, M. Heil, T. Kim, A. Miyaguchi, Fine-grained classification for poisonous fungi identification with transfer learning, CLEF 2024 Working Notes CEUR-WS (2024).
[8] Z. Liu, H. Hu, Y. Lin, Y. Zhuliang, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, F. Wei, B. Guo, Swin transformer v2: Scaling up capacity and resolution, CVPR 2022, arXiv:2111.09883 (2022).
[9] S. Wolf, P. H. Thelen, J. Beyerer, Poison-aware open-set fungi classification: Reducing the risk of poisonous confusion, CLEF 2024 Working Notes CEUR-WS (2024).
[10] H. Goëau, J.-C. Lombardo, A. Affouard, V. Espitalier, P. Bonnet, A. Joly, PlantCLEF 2024 Pretrained Models on the Flora of Southwestern Europe Based on a Subset of Pl@ntNet Collaborative Images and a ViT Base Patch 14 DINOv2, 2024. URL: https://zenodo.org/records/10848263.
[11] C. Hughes, Demystifying PyTorch's WeightedRandomSampler by example, 2024. URL: https://medium.com/data-science/demystifying-pytorchs-weightedrandomsampler-by-example-a68aceccb452.
[12] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, ICLR 2018, arXiv:1710.09412 (2018).
[13] PACE, Partnership for an Advanced Computing Environment (PACE), 2017. URL: http://www.pace.gatech.edu.
[14] PyTorch, CrossEntropyLoss, 2025. URL: https://docs.pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html.
[15] K. Poulinakis, Img_Premature_Ending-Detect_Fix.py, 2021. URL: https://github.com/Poulinakis-Konstantinos/ML-util-functions/blob/master/scripts/Img_Premature_Ending-Detect_Fix.py.
[16] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, ICLR 2015, arXiv:1412.6980 (2017).
[17] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misa, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, P. Bojanowski, DINOv2: Learning robust visual features without supervision, Transactions on Machine Learning Research, arXiv:2304.07193 (2024).
[18] H. Bao, L. Dong, S. Piao, F. Wei, Beit: Bert pre-training of image transformers, ICLR 2022, arXiv:2106.08254 (2022).
[19] 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, ICLR 2021, arXiv:2010.11929 (2021).
[20] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, A. Courville, D. Lopez-Paz, Y. Bengio, Manifold mixup: Better representations by interpolating hidden states, ICML 2019, arXiv:1806.05236 (2019).
[21] B. Warner, A. Chaffin, B. Clavié, O. Weller, O. Hallström, S. Taghadouini, A. Gallapher, R. Biswas, F. Ladhak, T. Aarsen, N. Cooper, G. Adams, J. Howard, I. Poli, Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference, arXiv:2412.13663 (2024).
[22] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, J. Kang, Biobert: a pre-trained biomedical language representation model for biomedical text mining, Bioinformatics 2019, arXiv:1901.08746 (2019).
[23] Z. Chen, V. Badrinarayanan, C.-Y. Lee, A. Rabinovich, Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks, ICML 2018, arXiv:1711.02257 (2018).
[24] S. Stevens, J. Wu, M. J. Thompson, E. G. Campolongo, C. H. Song, D. E. Carlyn, L. Dong, W. M. Dahdul, C. Stewart, T. Berger-Wolf, W.-L. Chao, Y. Su, Bioclip: A vision foundation model for the tree of life, CVPR 2024, arXiv:2311.18803 (2024).
[25] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, I. Sutskever, Learning transferable visual models from natural language supervision, ICML 2021, arXiv:2103.00020 (2021).
[26] G. Team, T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, S. Pathak, L. Sifre, M. Rivière, M. S. Kale, J. Love, et al., Gemma: Open models based on gemini research and technology, arXiv preprint

arXiv:2403.08295 (2024).

[27] M. Abdin, J. Aneja, H. Awadalla, A. Awadallah, A. A. Awan, N. Bach, A. Bahree, A. Bakhtiari, J. Bao, H. Behl, et al., Phi-3 technical report: A highly capable language model locally on your phone, arXiv preprint arXiv:2404.14219 (2024).

[28] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., Llama: Open and efficient foundation language models, arXiv preprint arXiv:2302.13971 (2023).