OpenWGAN-GP for Fine-Grained Open-Set Fungi Classification

Jack N. Etheredge[1,*]

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

Abstract

Understanding and accurately classifying fungi is crucial for ecological studies, food safety, and public health. In this paper, I present my approach to the FungiCLEF 2024 challenge, which aims to classify images of fungi, identify open-set "unknown" fungi species in the test data, and reduce the confusion between edible and poisonous fungi. This method leverages a combination of *Metaformer-0*, *Metaformer-2*, and *CAFormer-S18* models, chosen for their strong classification performance relative to their computational efficiency. The *Metaformer-0* and *Metaformer-2* models utilize metadata, while *CAFormer-S18* does not, yet all belong to the same family of models known as *Metaformers* and employ convolutional blocks followed by multi-headed self-attention transformer blocks. My primary novel contribution is the application of *OpenGAN* to detect unknown fungi species, enhanced by incorporating *WGAN-GP* to improve training stability, resulting in a new open-set classifier training paradigm I term *OpenWGAN-GP*. This approach enables a lightweight discriminator to utilize the latent representations from the closed-set classifier for binary classification of open-set vs. closed-set species. My best-performing ensemble achieved public leaderboard scores of 0.2394 for *Track 1*, 0.1681 for *Track 2*, and 0.4075 for *Track 3*, along with a macro-averaged *F1* score of 49.81%. *Track 1* represents the classification loss with unknowns, *Track 2* represents the edible-poisonous confusion loss (weighted heavily for poisonous to edible misclassifications), and *Track 3* is the sum of *Track 1* and *Track 2*. My method secured 1st place in the FungiCLEF 2024 competition for *Track 1*, *F1*, and Accuracy on the private leaderboard. Code is available at https://github.com/Jack-Etheredge/fungiclef2024.

Keywords

OpenWGAN-GP, OpenGAN, Open-set recognition, Fine-grained classification, FungiCLEF

1. Introduction

Correctly identifying mushroom species and distinguishing between poisonous and edible varieties are critical for public health. In 2023 alone China had 1,303 reported cases of mushroom poisonings traced to 97 species of mushroom, of which 12 were newly discovered species [1]. FungiCLEF 2024 [2] is a competition held as part of the LifeCLEF 2024 [3] lab [1]. FungiCLEF is a long-tailed fine-grained open-set classification task with an additional asymmetrically weighted edible-poisonous confusion component. In this work, I propose a novel solution for fine-grained classification of fungi species that simultaneously minimizes confusion between edible and poisonous species as well as detecting species of fungi unknown to the training dataset. The primary contributions of this work are 1) the use of an open-set recognition classifier trained using the embeddings from the closed-set classifier applied to fine-grained open-set recognition as well as 2) leveraging an ensemble of computationally lean models with carefully selected test-time augmentations. Extensive experimentation was used to improve the training methodology of this discriminator. I refer to this optimized open-set classifier training paradigm as *OpenWGAN-GP*.

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France

*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/LifeCLEF2024

2. Related work

2.1. Fine-grained classification

Fine-grained classification involves classifying data that belongs to the same greater category. Fine-grained classification is challenging due to the large intra-set variation and low inter-set variation, in contrast to standard coarse grained recognition. Fine-grained data, like open-set data, are ubiquitous in the real world, and addressing the unique challenges posed by both has become the subject of more intense academic interest recently. *Metaformer* [4] has been shown to perform well across a diverse set of fine-grained datasets such as iNaturalist [5], NABirds [6], and CUB-200-2011 [7] by incorporating metadata information into a hybrid convolutional vision transformer architecture. Many methods have been employed recently to classify fungi for the FungiCLEF dataset [8], including the aforementioned *Metaformer*.

2.2. Open-set recognition

Most works on open-set recognition primarily utilize coarse-grained classification datasets such as CIFAR-10 [9], and Tiny-ImageNet [10], alongside the fine-grained digit recognition datasets MNIST [11] and SVHN [12], as outlined in [13]. Some studies even involve cross-dataset comparisons, which generally fall into the broader category of out-of-distribution detection. However, open-set recognition within fine-grained classification is more challenging because the open-set data shares low inter-class variation with the closed-set data, as both belong to the same coarse-grained category or super-category (e.g., all fungi). For instance, while an image of a dog is clearly out-of-distribution for a fungi dataset, an unseen species of fungus is more semantically similar to known fungi species, making it harder to identify as unknown. This highlights the unique challenge of fine-grained open-set recognition.

FungiCLEF presents an opportunity to address this challenge by applying open-set recognition techniques to fine-grained data, a task where many methods have shown promise [8]. Various techniques for open-set recognition and out-of-distribution detection exist, ranging from simple threshold-based methods such as maximum softmax probability [14], maximum logit [15, 16], or softmax entropy [17], to K+1 way classifier which treats all unknowns as an additional class [18, 13]. More advanced methods involve specialized models for handling open-set data.

In this work, I utilized *OpenGAN* [13], which belongs to the category of specialized models. *OpenGAN* trains a binary classifier (the discriminator) using labeled data from both the closed-set and open-set, with a generator creating synthetic examples of open-set data during discriminator training to aid the discriminator in generalizing beyond the labeled open-set data. I enhanced this approach by improving the training stability of the open-set classifier, applying it to the fine-grained FungiCLEF 2024 dataset to effectively detect unknown fungi species.

3. Methodology

3.1. Dataset

The FungiCLEF 2024 challenge [2] dataset originates from the Danish Fungi 2020 dataset [19], which comprises 295,938 training images representing 1,604 species primarily observed within Denmark. Each training sample underwent rigorous expert validation to ensure precise labeling. In addition to images, comprehensive observation metadata including habitat, substrate, time, location are provided. The validation set consists of 30,131 observations encompassing 60,832 images across 2,713 species, spanning the entirety of the year and capturing observations from diverse substrate and habitat categories.

3.2. Competition objective

FungiCLEF represents two distinct challenges. One challenge is fine-grained and long-tailed classification. Seesaw loss [20] was used to counteract the effect of the long-tailed class distribution. Seesaw loss is a modification to standard cross-entropy loss. Given predicted logits z and predicted probabilities σ from the classifier, and y_i is the one-hot encoded ground truth label with $1 <= i <= C$, seesaw loss is defined as

$$L_{\text{seesaw}}(z) = -\sum_{i=1}^{C} y_i \log(\sigma_i), \tag{1}$$

over classes C where σ_i is defined as

$$\sigma_i = \frac{e^{z_i}}{\sum_{j \neq i}^{C} S_{ij} e^{z_j} + e^{z_i}}. \tag{2}$$

S_{ij} is a balancing coefficient between different classes. S_{ij} is determined by a combination of a mitigation factor M_{ij} and a compensation factor C_{ij}:

$$S_{ij} = M_{ij} \cdot C_{ij} \tag{3}$$

M_{ij} mitigates the penalty on tail classes based on their instance ratio compared to head classes by decreasing the penalty on class j relative to the ratio of instance counts between the less abundant tail class j and the more abundant head class i. Conversely, C_{ij} increases the penalty on class j whenever a misclassification occurs from class i to class j. This dual-factor approach in S_{ij} allows Seesaw loss to dynamically adjust penalties based on both instance distribution and misclassification behavior, optimizing the learning process in long-tailed multi-class classification tasks. The loss function is explained in greater detail in the original paper [20].

The closed-set image classification models used to classify the images belong to a family of hybrid convolutional and self-attention transformer models known as *Metaformers* [21]. These models are explained in detail in Section 3.5.

The other challenge is the recognition of the open-set "unknown" class. *OpenWGAN-GP* was used to classify images as belonging to the closed-set or open-set datasets. The architecture of the open-set discriminator and the *OpenWGAN-GP* training methodology are described in greater detail in Section 3.6.

3.3. Evaluation metrics

The public leaderboard for the competition reported multiple metrics for each submission. *Track 1* was a classification loss that included unknowns. *Track 2* was an edible-poisonous confusion loss with a ×100 weight for poisonous ⟶ edible misclassification. *Track 3* was the sum of the *Track 1* and *Track 2* losses. Additionally, the macro-averaged *F1* score and accuracy were reported. The accuracy has been ignored for all experimental results reported here. Apart from the macro-averaged *F1* score, none of the metrics correct for class imbalance. Thus the impact of classification performance on each class impacts final performance for Tracks 1-3 proportional to the number of observations for that class.

Track 1 loss is a standard classification error with an additional "unknown" class:

$$L_1 = \sum_i W_1(y_i, q(x_i)), \tag{4}$$

for class predictions $q(x)$ for observations x from a classifier q and true labels y. The cost function W_1 is defined as

$$W_1(y, q(x)) = \begin{cases} 0 & \text{if } q(x) = y \\ 1 & \text{otherwise} \end{cases}. \tag{5}$$

Track 2 loss penalizes the confusion of edible and poisonous species. Consider a function p that indicates poisonous species as $p(y) = 1$ if species y is poisonous, and $p(y) = 0$ otherwise. Let c_{PSC} denote the cost for poisonous ⟶ edible misclassification (a poisonous observation was predicted as edible) and c_{ESC} the cost for edible ⟶ poisonous misclassfication. $c_{ESC} = 1$ and $c_{PSC} = 100$. Track 2 loss is defined as:

$$L_2 = \sum_i W_2(y_i, q(x_i)), \tag{6}$$

for class predictions $q(x)$ for observations x from a classifier q and true labels y as in L_1. The cost function W_2 is defined as

$$W_2(y, q(x)) = \begin{cases} 0 & \text{if } p(y) = p(q(x)) \\ c_{PSC} & \text{if } p(y) = 1 \text{ and } p(q(x)) = 0 \\ c_{ESC} & \text{otherwise} \end{cases} \tag{7}$$

Track 3 (the “user-focused loss”) is simply the sum of Track 1 and Track 2 losses:

$$L_3 = L_1 + L_2. \tag{8}$$

3.4. Custom poison loss

A custom poison loss was used for all of the final models. The poison loss was formulated as a class weighted binary cross entropy loss.

Given the set of poisonous classes P and the set of edible classes ε, sum the probabilities of each class independently where P_i is the softmax probability output for class i.

$$P_{poisonous} = \sum_{i \in P} P_i. \tag{9}$$

$$P_{edible} = \sum_{i \in \varepsilon} P_i. \tag{10}$$

Let $y \in \{0, 1\}$ be the binary ground truth label for an image, where 1 indicates a poisonous class and 0 indicates an edible class. $\alpha = 100$ is the weight assigned to the edible class to penalize edible ⟶ poisonous misclassifications. Thus, the weighted cross-entropy loss is as follows:

$$L_{poison,weighted} = -[y \log(P_{poisonous}) + \alpha(1 - y) \log(P_{edible})] \tag{11}$$

Since the softmax probabilities output by the model sum to 1, the probabilities for all the poisonous classes and all the edible classes were summed independently and used as the prediction for the binary cross entropy criterion. A weight of 100 was assigned to the edible class, since edible ⟶ poisonous misclassifications (true label is edible, predicted label is poisonous) was penalized ×100 in the *Track 2* loss. The total training loss was the sum of the seesaw loss and the custom poison loss. This approach ensures that the training process emphasizes correctly classifying edible species as edible, thus reducing the risk of mistakenly classifying edible species as poisonous, which is heavily penalized in the evaluation metrics.

3.5. Model architectures

All experiments were performed on a machine with a single NVIDIA RTX 3090 graphics card and all models were trained using PyTorch [22]. Given that I was working on a single computer for this competition, efficient use of the limited compute available was of critical importance. Ensembling and test-time augmentations were used to increase performance while keeping training efficiency in check. An ensemble of computationally lean models have been shown to outperform a single larger model with respect to both training and inference cost [23]. Model architectures were chosen based on their performance on ImageNet and/or iNaturalist relative to the computational complexity of the models in TFLOPs, aiming for a final ensemble of at least two models. Test time augmentations allow for much greater performance without requiring any additional training of the models, making them a particularly attractive target for optimization when compute and time are both limited.

Metaformers [21] are a family of models that combine different tokenizers with a transformer backbone. A collection of *Metaformer* models (*Metaformer-0*, *Metaformer-1*, and *Metaformer-2*) were created in [4] that combine metadata with the images to improve the classification performance of the models on multiple fine-grained image datasets. *CAFormer* [24] models are very similar in architecture to the *Metaformer* variants proposed in [4], but do not make use of metadata. In the final ensemble, *Metaformer-0*, *Metaformer-2*, and *CAFormer-S18* were used. Hereafter *Metaformer* refers to the *Metaformer* models from [4] which incorporate metadata information.

I fine-tuned *CAFormer-S18* with weights pretrained on ImageNet-21K [25] while *Metaformer-0* and *Metaformer-2* models were pretrained on iNaturalist2021 [5]. *CAFormer-S18* was fine-tuned on a different train-validation split than the two *Metaformer* models. *Metaformer-0* and *Metaformer-2* differ only in the number of channels in the convolutional and transformer blocks, with *Metaformer-2* having more channels in every block. The *S18* variant of *CAFormer* refers to a specific combination of convolutional and self-attention token mixers. *CAFormer-S18* utilizes a total of 18 blocks: 3 convolution blocks with 64 channels, 3 convolution blocks with 128 channels, 9 attention blocks with 320 channels, and 3 attention blocks with 512 channels.

3.6. *OpenGAN* and *OpenWGAN-GP*

To my knowledge, this is the first time that *OpenGAN* has been utilized for open set recognition of fine-grained images beyond digit recognition. In order to improve training stability, I incorporated the Wasserstein *GAN* loss and gradient penalty (*WGAN-GP*) [26] into the training of *OpenGAN* [13] to create *OpenWGAN-GP*. In addition to incorporating *WGAN-GP*, batch normalization layers were replaced with layer normalization layers for the discriminator as suggested in [26].

OpenGAN proposes selection of the discriminator against a validation set of open- and closed-set examples, selecting the discriminator with the best validation *ROC-AUC*. However, since the *ROC-AUC* is calculated using a range of classification thresholds, the best classification threshold would also need to be determined for each *OpenGAN* discriminator. As such, the macro-averaged *F1* was used as the selection metric instead of *ROC-AUC* for *OpenWGAN-GP*. Additionally, if the *OpenWGAN-GP* discriminator is selected based on macro-*F1*, an ensemble can be averaged without the need to calibrate the classification threshold.

Another difference from the original *OpenGAN* paper is that models were selected based on their macro-*F1* performance rather than the proposed *ROC-AUC*. Since models would be used in an ensemble and the *OpenWGAN-GP* probabilities would be averaged, it was important to assume that the classification threshold for all of the individual *OpenWGAN-GP* models would be the same. An alternative strategy would have been voting, which would have allowed for different classification thresholds per model, but this was not explored in this study.

OpenGAN is a methodology for training a lightweight discriminator that utilizes the intermediate representation of an image to generate a binary classification for open-set recognition. Several related methods were proposed, but the one that performed best in their experiments and the one that I focus on in this work is *OpenGAN*$^{\text{fea}}$ with the inclusion of open-set training data, which I will simply refer to as *OpenGAN*. This paradigm allows for training a classification without initial consideration of the open set data. The discriminator is a multilayer perceptron consisting of fully connected layers with sizes $D \rightarrow H \times 8 \rightarrow H \times 4 \rightarrow H \times 2 \rightarrow H \rightarrow 1$, where D represents the dimension of the intermediate representation from the closed-set classifier and H is a hidden dimension multiplier. $H = 64$ unless otherwise specified. The output layer uses a sigmoid activation function. Batch normalization [27] and LeakyRELU [28] are used between each dense layer. During training, the generator generates a feature vector of length D from a 100-dimensional input vector with each value sampled independently from a standard normal distribution (mean 0, variance 1). The generator is also a multilayer perceptron with batch normalization and LeakyRELU. It has a similar architecture, but there are some critical differences. The output dimension of the generator must match the input dimension D

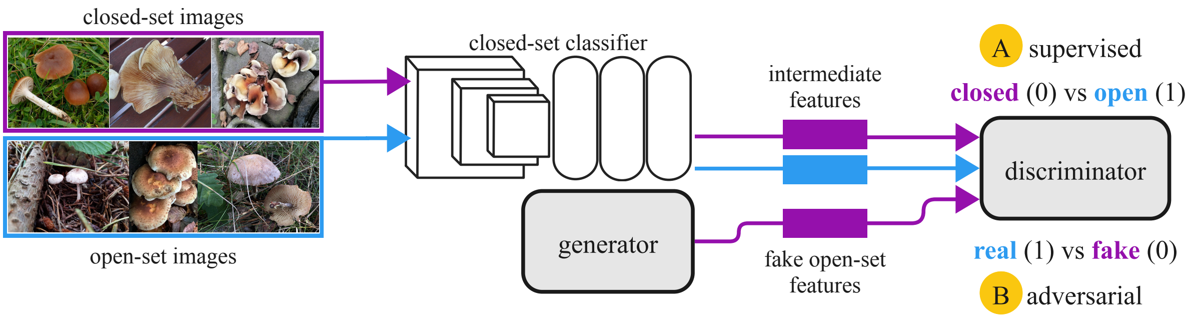


Figure 1: OpenGAN training paradigm. After a closed-set classifier has been trained, an open-set discriminator is trained through both A) supervised learning and B) adversarial training using the penultimate layer feature representation of closed- and open-set observations. A generator is trained adversarially, with a loss that is minimized when the discriminator cannot distinguish between real features and features generated by the generator. This generator is used to supplement the open set training data for the discriminator. Figure reproduced and modified from [13] with permission. OpenWGAN-GP utilizes this same fundamental training paradigm. In contrast to the implementation of OpenGAN, OpenWGAN-GP was trained with the open-set features rather than the closed-set features sharing the real label. To simplify the figure, metadata is not considered, but in the case of Metaformer closed-set classifier models, the metadata would be included along with the images as an additional input.

of the discriminator, both of which correspond to the dimension of the intermediate representation of the closed-set classifier. As such, the generator is a multilayer perceptron with fully connected layers of sizes $G \rightarrow H \times 8 \rightarrow H \times 4 \rightarrow H \times 2 \rightarrow H \times 4 \rightarrow D$, where $G = 100$. The input dimension G is arbitrary, but this work utilizes a 100-dimensional input vector. As with the discriminator, $H = 64$ unless otherwise specified. Additionally, the output activation used by the generator is Tanh instead of sigmoid activation. During training, the discriminator is trained to classify the closed and open set data in a supervised manner with binary labels using binary cross entropy. The generator is used to generate additional open set data and both the generator and discriminator are trained using the standard *GAN* training paradigm [29]. As such, the discriminator is updated twice per update of the generator since the discriminator is trained adversarially against the generator (real vs fake) as well as supervised (open- vs closed-set). This training paradigm is illustrated in Figure 1. The *Adam* optimizer [30] was used with a learning rate of 1e-4 for the discriminator and 2e-4 for the generator. The higher learning rate is used for the generator to account for the 2:1 updates of the discriminator vs the generator.

3.7. Metadata

Metaformer-0 and *Metaformer-2* allow the fusion of metadata information with the vision information. Metadata was utilized to provide the model with information concerning location, local growth conditions, and temporal information by including the country code, substrate, and habitat, and observation date (month and day). Example substrates include “fruits”, “wood”, “cones”, “soil”, and “peat mosses”, while example habitats include “bog”, “dune”, “meadow”, and “roof”. There are 34, 32, and 31 categories for country code, substrate, and habitat, respectively. Metadata was preprocessed for *Metaformer-0* and *Metaformer-2* according to [17]. The month and day were transformed by periodic encoding into $\left[\sin\left(\frac{2\pi\,\text{month}}{12}\right), \cos\left(\frac{2\pi\,\text{month}}{12}\right)\right]$ and $\left[\sin\left(\frac{2\pi\,\text{day}}{31}\right), \cos\left(\frac{2\pi\,\text{day}}{31}\right)\right]$ respectively to preserve temporal relationships. Geographical information in the form of country codes, habitat, and substrate were all one-hot encoded. *Metaformer-0* and *Metaformer-2* utilize trainable embeddings to project this encoded metadata to the same dimensionality as the image features in order to fuse them with the latent representation of the images.

3.8. Training settings

I used the *AdamW* optimizer [31] with an initial learning rate of 1e-3 on only the classification output dense layer with the pretrained model frozen for the first 5 epochs, then reduced to 5e-5 for subsequent epochs. *CAFormer-S18* models were trained with a batch size of 40, while *Metaformer-0* models were trained with a batch size of 32, and *Metaformer-2* models were trained with a batch size of 12. A weight decay of 0.05 was used for all of these models. The learning rate was reduced by a factor of 0.1 if the model did not improve the validation loss for 5 consecutive epochs. Early stopping was also employed when training all models to prevent wasting compute time on models which were no longer improving in their generalization to the validation set as measured by the validation loss.

LogitNorm [32] describes a technique that applies an L2 norm to the logits during training (the norm is not applied during inference). It was included with the hope that it should improve the separation of the classes in the embedding space relative to standard seesaw loss, which might enable *OpenWGAN-GP* to leverage the embedding space for more accurate open-set recognition. Additionally, LogitNorm was shown to act similarly to temperature scaling [33] to create models that generate less overconfident predictions. This would be important for maximum softmax probability or entropy thresholding, which were explored as alternatives to *OpenWGAN-GP*.

3.9. Training data augmentation

Training was performed with a square random crop, TrivialAugment [34], horizontal flip with 50% probability, and GridMask [35] with a probability of 20%, applied in that order.

3.10. Test-time augmentations

At test-time, all images were resized with bicubic interpolation along the shortest dimension to 384 or 576 depending on the model followed by a square center crop of the same size. Test-time augmentations and ensembling were instrumental techniques for the final inference performance. Using a larger image size for inference than training was shown to improve accuracy for multiple datasets in FixRes [36].

The strategies employed were averaging horizontal flips, multi-instance averaging, ensemble averaging, and inference at a higher resolution (576x576) for *CAFormer-S18* relative to training (384x384). The overall inference pipeline is summarized in Figure 2. Fivecrop was also investigated, but could not be incorporated in the allowed compute budget. Since it did not yield as significant an improvement relative to a larger ensemble with horizontal flipping according to local evaluation and public leaderboard scores for *Track 3*, the chosen configuration was preferred. It could be useful in the future to experiment with ensembling techniques that are more sophisticated than simple averaging, but none were attempted in this study.

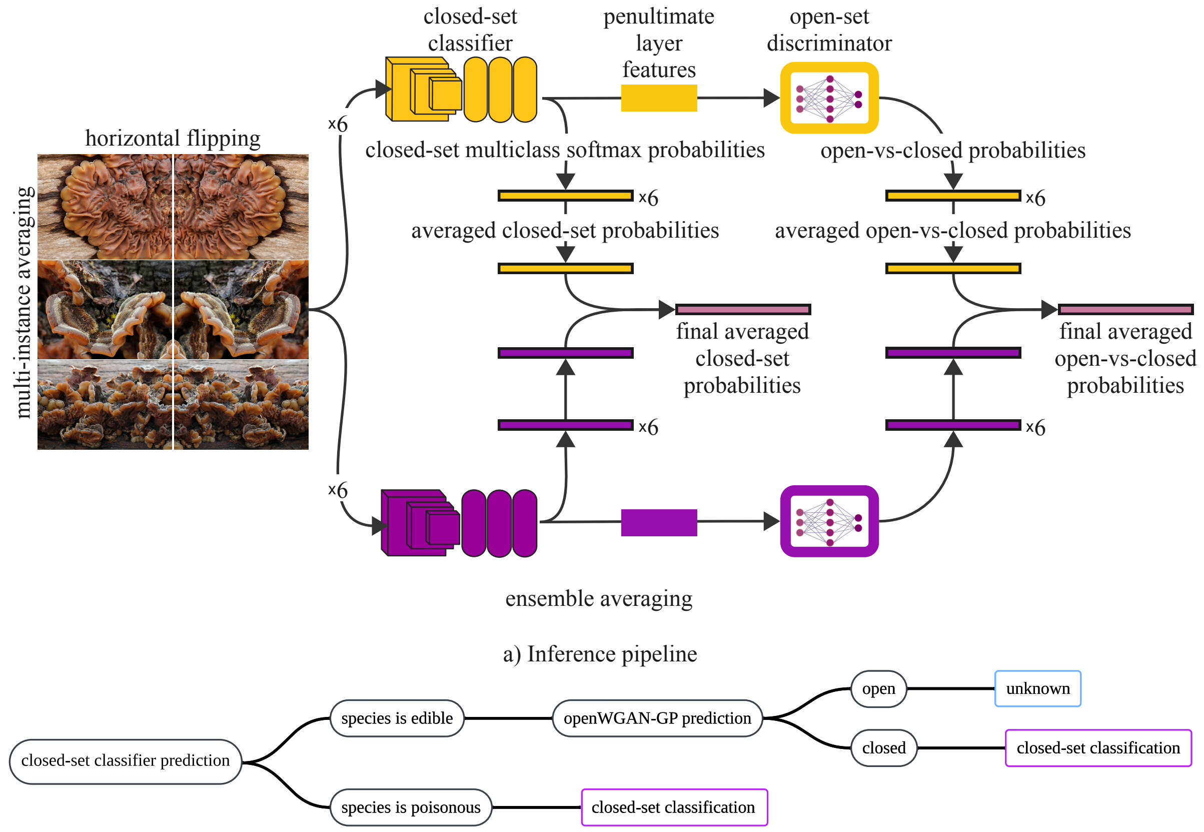
Due to the open set "unknown" class being implicitly edible, the penalty for misclassifying poisonous mushrooms as unknown was greater than the decrease in misclassification loss. To mitigate this in the proposed solution, if the top prediction of the classification network was a poisonous mushroom, the prediction from the *OpenGAN* open set classifier was ignored for that observation.

4. Experimental results

My best-performing ensemble and *OpenWGAN-GP* combinations achieved 1st place on *Track 1*, *F1*, and *Accuracy* on the private leaderboard. All results reported are for the public leaderboard test set unless explicitly stated otherwise.

4.1. Open-set recognition

Different open set detection methods were evaluated against the public leaderboard and these results can be seen in Table 1. Experiments with a local validation set showed that temperature scaling improved the performance of maximum softmax probability (MSP) thresholding as well as softmax


horizontal flipping
multi-instance averaging
×6
×6
closed-set classifier
penultimate layer features
open-set discriminator
closed-set multiclass softmax probabilities
open-vs-closed probabilities
×6
×6
averaged closed-set probabilities
averaged open-vs-closed probabilities
final averaged closed-set probabilities
final averaged open-vs-closed probabilities
×6
×6
ensemble averaging
closed-set classifier prediction
species is edible
openWGAN-GP prediction
open
unknown
closed
closed-set classification
species is poisonous
closed-set classification

a) Inference pipeline

b) Classification decision

Figure 2: Inference pipeline. (a) Averaged closed-set classification probabilities and open-set recognition probabilities are both generated for each observation. Multi-instance averaging, horizontal flipping, and ensemble averaging. To generate predictions for each observation, all instances belonging to that observation are considered. The images for each instance are horizontally flipped and both flips of every instance are used as input to each closed-set classifier in the ensemble. In the example shown, the observation has 3 instances, thus 6 images would be used as input into each model. The number of instances in the dataset is variable. Each closed-set classifier has a corresponding open-set discriminator model that has been trained on its penultimate layer feature representations of open and closed set images as illustrated in Figure 1. To simplify the figure, metadata was omitted, but for Metaformer models it is included as an additional input along with each image. The simplest case of a two model ensemble is illustrated, with each closed-set classifier and open-set discriminator pair shown in a different color. The ensemble used for the final leaderboard evaluation was composed of more than two models. (b) The final prediction is determined based on both the prediction from the closed-set classifier ensemble average probabilities as well as the prediction from the open-set discriminator ensemble average probabilities. If the predicted fungus species is poisonous, then the open-set recognition classification is ignored. If the predicted fungus species is edible, then the open-set recognition classification is used to determine whether the species is unknown or belongs to the closed-set classifier.

entropy thresholding (experiments not shown). *OpenWGAN-GP* consistently performed better on the leaderboard than softmax thresholding or entropy thresholding, even after temperature scaling [33] the probabilities. As can be seen in Table 1, the performance of either of these methods depends on optimizing a threshold. The optimal entropy threshold for local validation was 6, which did not appear to be optimal for the public leaderboard. This suggests that this method may not generalize well between test sets. *OpenWGAN-GP* is a binary classifier that is selected using the macro-*F1* score with a classification threshold of 0.5, which means that no additional thresholding should be needed to generalize between test sets. Table 1 shows that despite not tuning the classification threshold, *OpenWGAN-GP* shows the best *Track 1* and *Track 3* performance while maintaining a similar *Track 2* performance to MSP and entropy thresholding after avoiding poisonous ⟶ unknown misclassification as explained below.

Table 1
Thresholding experiments. All experiments are performed with an ensemble of three *CAFormer-S18* (A, B, and C data splits) using optimized training for *OpenWGAN-GP*, horizontal flips, and multi-instance averaging. *OpenWGAN-GP* has comparable or better performance across *Track 1*, *Track 2*, and *Track 3* metrics without the need to tune a threshold per test dataset.

Open-set recognition	Ignore Poison Pred	Temperature scaling	Track1↓	Track2↓	Track3↓	F1↑
Softmax T=0.25	✓	✓	0.3654	0.1751	0.5405	52.04
Softmax T=0.25	-	✓	0.3646	0.1847	0.5494	52.08
Softmax T=0.2	-	✓	0.3676	0.1798	0.5473	51.51
Entropy T=2	✓	✓	0.3752	0.1808	0.5559	52.52
Entropy T=2	-	✓	0.3738	0.3163	0.6901	**52.61**
Entropy T=3	✓	✓	0.3705	0.1751	0.5456	51
Entropy T=3	-	✓	0.3699	0.2296	0.5994	51.48
Entropy T=4	-	✓	0.3726	**0.1746**	0.5472	50.49
Entropy T=5	-	✓	0.3756	0.175	0.5506	50.01
Entropy T=6	-	✓	0.377	0.1751	0.5521	49.54
Entropy T=7	-	✓	0.3773	0.1751	0.5524	49.4
OpenWGAN-GP	-	-	**0.2287**	0.4438	0.6725	49.06
OpenWGAN-GP	✓	-	0.2458	0.1756	**0.4213**	49.22
None	-	-	0.3789	0.1811	0.56	48.95

Table 2
Ensemble experiments. *CAFormer-S18* shows strong performance relative to *Metaformers* despite not utilizing metadata information. Multiple splits of the data allow a simple method for ensembling and can improve performance beyond what can be achieved solely through employing different architectures in an ensemble. All experiments are performed using optimized training for *OpenWGAN-GP*, horizontal flips, and multi-instance averaging. *Metaformer-0* and *Metaformer-2* are always trained using train-val data split D. The data split each model in the ensemble was trained on is shown in parentheses (e.g. CAFormer-S18 (A) represents a CAFormer-S18 model trained on data split A).

Model(s) (data split)	Track1↓	Track2↓	Track3↓	F1↑
Metaformer-0 (D)	0.2988	0.1864	0.4852	45.14
Metaformer-2 (D)	0.2944	0.2082	0.5026	45.07
CAFormer-S18 (A)	0.2945	0.2042	0.4987	46.47
CAFormer-S18 (B)	0.2819	0.1766	0.4585	46.53
CAFormer-S18 (C)	0.2796	0.1759	0.4555	44.67
CAFormer-S18 (A), CAFormer-S18 (B), CAFormer-S18 (C)	0.2458	0.1756	0.4213	49.22
Metaformer-0 (D), CAFormer-S18 (B), CAFormer-S18 (C)	0.2424	0.1857	0.4281	49.71
Metaformer-0 (D), Metaformer-2 (D), CAFormer-S18 (A)	0.2436	0.1737	0.4174	**49.89**
Metaformer-0 (D), Metaformer-2 (D), CAFormer-S18 (C)	**0.2394**	**0.1681**	**0.4075**	49.81

It can be seen from the results in Table 1 that ignoring *OpenWGAN-GP* predictions for open-set recognition in the cases when the highest predicted probability belongs to a poisonous species ("ignore poison pred") is critical to preventing the open-set recognition from degrading performance on *Track 3*. *OpenWGAN-GP* without ignore poison pred achieves a better *Track 1* score than *OpenWGAN-GP* with ignore poison pred, but a much higher *Track 2* score. This suggests that in many cases *OpenWGAN-GP* is correctly identifying unknowns that the classification network is predicting to be poisonous, but that the poisonous → edible cost for the poisonous closed → open misclassifications overwhelms the improvement in classification loss. This reinforces how challenging it is to simultaneously optimize classification performance, identification of unknown species, and avoidance of misclassifying poisonous species as edible.

Following [17], I explored fine-tuning the models through outlier exposure after first training the models without the inclusion of unknowns, but validation loss failed to improve after the first epoch upon inclusion of unknowns (results not shown). It appears that unknowns were included for the entire duration of training in their work. Unfortunately, I was unable to complete this experiment before the competition concluded.

4.2. Model architectures and ensemble selection

Multiple architectures were evaluated for this study. *Efficientnet-B0* [37] and *Efficientnetv2-S* [38] were both experimented with but their results were not as promising as *Metaformer* and *CAFormer* against a local validation set in early experiments. Results are not shown for these experiments since they are not directly comparable to the experiments reported. *CAFormer-S18* has better performance than *Metaformer-0* when the image resolution is increased at inference time despite belonging to the same family of models as *Metaformer-0* and *Metaformer-2*, which leverage metadata information. *CAFormers* performed almost as well as *Metaformer* despite not utilizing information from the metadata. Future work could evaluate merging the two architectures into a *CAFormer* with a head for the metadata information. An ensemble of three *CAFormer-S18* models that vary only by their training and validation data split performs nearly as well as ensembles of *Metaformer-0*, *Metaformer-2*, and *CAFormer-S18*. The best performing ensemble was *Metaformer-0*, *Metaformer-2*, *CAFormer-S18* split C. Split C performs better than the other two *CAFormer-S18* data splits, which provides a likely explanation as to why this ensemble outperformed *Metaformer-0*, *Metaformer-2*, and *CAFormer-S18* split A.

4.3. Optimization of *OpenWGAN-GP* training

OpenWGAN-GP training was optimized with respect to the hidden dimension size, ratio of closed and open set samples, and whether training augmentations were applied. Table 3 shows that the best *Track 3* performance is achieved when using training augmentations and oversampling the open-set data to roughly the same number of samples as the closed-set data. This case is represented as weighted undersampling (w.u.) for closed-set sampling and “3x all” sampling for open-set sampling, which represents the case of oversampling the open-set dataset completely 3 times with training augmentations to increase the diversity of representations of the limited open-set data. These settings are used for all results shown in Table 2 and the *OpenWGAN-GP* results in Table 1. For sampling the closed-set data, weighted undersampling outperforms random undersampling and balanced undersampling. In cases where there is a >5% disparity between the number of samples in the open and closed sets, training is performed with balanced sampling between the open and closed sets. This pertains to all combinations except the 3x oversampling of the open set and weighted undersampling of the closed set. Local experimentation suggested that using the entire closed set dataset could yield a slight increase in *Track 3* performance, but this dramatically increases the training time for the *OpenWGAN-GP* classifier (experiments not shown). While the *Track 3* performance does not appear to be particularly sensitive to the hidden dimension size, the trend suggests that a smaller hidden dimension may have slightly improved performance, as shown in Table 4. More exhaustive combinations could not be performed due to competition submission limitations.

4.4. Test-time augmentations

Several test-time augmentations were evaluated. The performance of each of these augmentations is shown in Table 5. Multi-instance averaging, averaging of horizontal flips of the same image, averaging of multiple crops of the same image, and averaging multiple image sizes are explored with Metaformer-0 trained on split D. Since Metaformer-0 does not support inference at a different resolution than the training resolution (in this case 384x384), the image size in Table 5 refers to the image resolution of the shorter dimension before a square crop of 384. For example, if the image size is 441, then the image is resized to 441 along the shorter dimension (assuming it is a rectangular image) and then a square center crop of 384 is taken. As such, image size must be at least 384 for Metaformer-0. Each augmentation improves performance individually and in combination. Of the test-time augmentations that were experimented with, multi-instance averaging has the greatest impact of any individual transformation.

Table 3
OpenWGAN-GP training optimization. "n.o." refers to the number of open set samples in thousands. "n.c." refers to the number of closed set samples in thousands. The best performance for *Track 3* was achieved by ignoring poison predictions ("i.p.p.") and balancing the datasets using oversampling for the open-set data and weighted undersampling of the closed-set data. Training augmentations were performed for both datasets. In all cases, the closed set is undersampled. For closed-set sampling ("closed sample"), "w.u." refers to weighted undersampling and "bal" refers to class-balanced undersampling. For open-set sampling ("open sample"), "all" refers to the use of all open-set (unknown) samples in the training fraction of the training-validation split, while "3x all" refers to using all open-set samples 3 times each with training augmentations. For both open- and closed-set sampling, "random" refers to random undersampling. "aug" refers to the inclusion of training augmentations for *OpenWGAN-GP* training. "dim" refers to the hidden dimension multiplier for the open set discriminator. Row color is used to visually differentiate between data splits. Baselines without any open-set recognition are included for reference. The best results for each data split are in bold.

Data split	open WGAN -GP	i.p.p.	closed sample	open sample	n closed	n open	aug	Track1↓	Track2↓	Track3↓	F1↑
A	-	-	-	-	-	-	-	0.3860	**0.2042**	0.5902	46.36
A	✓	-	random	all	50	19	-	0.2911	0.3189	0.6100	46.34
A	✓	-	random	all	50	19	✓	**0.2788**	0.5917	0.8705	44.70
A	✓	-	random	random	50	2	-	0.3720	0.2237	0.5957	46.39
A	✓	-	random	random	50	2	✓	0.3724	0.2535	0.6259	46.16
A	✓	✓	w.u.	3x all	60	57	✓	0.2805	**0.2042**	**0.4846**	45.82
A	✓	✓	w.u.	all	60	19	-	0.3135	**0.2042**	0.5177	**47.26**
A	✓	✓	w.u.	all	60	19	✓	0.2945	**0.2042**	0.4987	46.47
B	✓	✓	w.u.	3x all	60	57	✓	**0.2819**	**0.1766**	**0.4585**	46.53
B	✓	-	random	random	50	2	-	0.3734	0.1974	0.5708	**46.69**
B	✓	✓	w.u.	all	60	19	-	0.2896	**0.1766**	0.4662	46.27
C	-	-	-	-	-	-	-	0.3836	**0.1759**	0.5595	**47.87**
C	✓	-	random	random	50	2	-	0.3614	0.2109	0.5723	47.56
C	✓	-	bal	all	34	19	-	0.2833	0.3809	0.6641	46.56
C	✓	✓	w.u.	3x all	60	57	✓	**0.2796**	**0.1759**	**0.4555**	44.67
C	✓	✓	w.u.	all	60	19	-	0.2918	**0.1759**	0.4677	46.04

Track 3 performance is best for multi-instance averaging in combination with horizontal flip averaging of the combinations shown in Table 5.

Despite improved performance on local validation, increasing the image resolution to 576 relative to the training resolution of 384 appears to have mixed results on the leaderboard performance as shown in Table 6. For data split A, the *Track 1* score is improved with a higher resolution, but *Track 2*, *Track 3*, and *F1* performance are better with an image resolution of 384. Image resolution 384 seems favored overall.

Table 7 shows that the removal of poison loss degrades the performance of the model across all metrics. *Track 2* has a larger percent change than *Track 1* or *F1*, which is sensible given that *Track 2* corresponds to the edible ⟶ poisonous confusion loss.

Removal of LogitNorm from the training decreased performance for *Track 1* and *F1* as shown in Table 8. Presumably this is due to improving the separation of the classes in the latent space which is used by *OpenWGAN-GP* for unknown classification. Future work could explore whether LogitNorm also improves classification of fine-grained datasets in cases for which open-set recognition is not a consideration. Interestingly, removal of LogitNorm increases performance in *Track 2* as shown in Table 8. The gain in performance in *Track 2* from the removal of LogitNorm is great enough that *Track 3* (the sum of *Track 1* and *Track 2* losses) is improved. This may suggest that LogitNorm is incompatible with the the poison loss used in this work.

Table 4
OpenWGAN-GP discriminator hidden dimension multiple optimization. “n open” refers to the number of open set samples in thousands. “n closed” refers to the number of closed set samples in thousands. In all cases, the closed set is undersampled. For closed-set sampling, “w.u.” refers to weighted undersampling. “dim” refers to the hidden dimension multiplier for the open set discriminator. Row color is used to visually differentiate between experiment configurations beyond hidden dimension multiple. For each configuration, the best results per metric are in bold. Data split C was used in all cases. No training augmentations were used in an experiments.

closed sampling	open sampling	n closed	n open	dim	Track1↓	Track2↓	Track3↓	F1↑
random	random	50	2	64	**0.3614**	0.2109	0.5723	47.56
random	random	50	2	26	0.3622	**0.1907**	**0.5529**	**47.76**
w.u.	all	60	19	26	**0.2771**	0.4534	**0.7306**	45.95
w.u.	all	60	19	32	0.2838	**0.4481**	0.7319	**46.24**

Table 5
Test-time augmentations. All models were trained with seesaw loss and poison loss. Open-set recognition is performed in all cases with optimized *OpenWGAN-GP* ignoring *OpenWGAN-GP* in cases where the closed-set classifier’s top prediction is a poisonous species. Performance of horizontal flip averaging (HFlip), multi-instance averaging, and multiple crop averaging (multicrop) combinations are compared. All experiments are performed with a Metaformer-0 model trained on data split D (“Metaformer-0 (D)”). In the case of multiple image resolutions ([384, 441]), the probabilities output by the model for both image resolutions are averaged before making a classification decision.

Model (data split)	Multi-instance	HFlip	Image size	Multicrop	Track1↓	Track2↓	Track3↓	F1↑
Metaformer-0 (D)	✓	✓	384	-	0.2990	**0.1805**	**0.4795**	**45.14**
Metaformer-0 (D)	✓	-	384	-	**0.2988**	0.1864	0.4852	**45.14**
Metaformer-0 (D)	-	✓	384	-	0.3272	0.2769	0.6041	41.44
Metaformer-0 (D)	-	-	384	✓	0.3245	0.2767	0.6013	42.10
Metaformer-0 (D)	-	-	[384, 441]	-	0.3213	0.2874	0.6088	42.37
Metaformer-0 (D)	-	-	384	-	0.3266	0.2827	0.6093	41.64

Table 6
Image size. Increased image resolution for inference relative to training. 384 to 576. Multi-instance averaging and horizontal flip averaging are used in all cases. Open-set recognition is performed in all cases with optimized *OpenWGAN-GP* ignoring *OpenWGAN-GP* in cases where the closed-set classifier’s top prediction is a poisonous species. Best results are per data split are shown in bold.

Model (data split)	Image size	Track1↓	Track2↓	Track3↓	F1↑
CAFormer-S18 (A)	384	0.2977	**0.1813**	**0.479**	**47.22**
CAFormer-S18 (A)	576	**0.2805**	0.2042	0.4846	45.82
CAFormer-S18 (B)	384	0.2963	**0.1761**	0.4724	**47.41**
CAFormer-S18 (B)	576	**0.2819**	0.1766	**0.4585**	46.53
CAFormer-S18 (C)	384	**0.2791**	**0.1694**	**0.4485**	**46.95**
CAFormer-S18 (C)	576	0.2796	0.1759	0.4555	44.67

4.5. *OpenWGAN-GP*

The training stability and overall performance of *OpenWGAN-GP* is demonstrated relative to the original *OpenGAN* in Table 9. The identical performance of *OpenGAN* with and without these training optimizations suggests that the particular failure state observed is *OpenGAN* classifying none of the test set observations as open-set. The data sampling optimizations explored in Table 3 were not sufficient to overcome the failure of *OpenGAN* to learn a meaningful representation of the data for classification. The

Table 7
Poison loss ablation. Both models use image size 576 with no additional augmentations (no multi-instance, horizontal flip, or multicrop averaging). Both models are using the optimized *OpenWGAN-GP* and ignoring the *OpenWGAN-GP* prediction in the case of the top classification prediction belonging to a poisonous species.

Model (data split)	Poison loss	Track1↓	Track2↓	Track3↓	F1↑
CAFormer-S18 (A)	-	0.3332	0.4175	0.7507	40.78
CAFormer-S18 (A)	✓	**0.3105**	**0.3071**	**0.6176**	**41.22**

Table 8
LogitNorm ablation. As in Table 7, both models use image size 576 with no additional augmentations (no multi-instance, horizontal flip, or multicrop averaging). Both models are using the optimized *OpenWGAN-GP* and ignoring the *OpenWGAN-GP* prediction in the case of the top classification prediction belonging to a poisonous species. The baseline model is the same as Table 7.

Model (data split)	LogitNorm	Track1↓	Track2↓	Track3↓	F1↑
CAFormer-S18 (A)	-	0.3370	**0.2405**	**0.5775**	38.47
CAFormer-S18 (A)	✓	**0.3105**	0.3071	0.6176	**41.22**

Table 9
OpenWGAN-GP vs vanilla *OpenGAN*. All results shown use the same model for classification and thus to generate embeddings for *OpenGAN* training and inference. All the models are ignoring the *OpenGAN*/*OpenWGAN-GP* prediction in the case of the top classification prediction belonging to a poisonous species. In all cases, the closed-set classification model is *CAFormer-S18* trained on data split A. The baseline model is the same as Tables 7 and 8.

OpenWGAN-GP data sampling optimizations	OpenGAN variant	Selection metric	Track1↓	Track2↓	Track3↓	F1↑
✓	OpenGAN	F1 macro	0.4143	0.2735	0.6878	**41.78**
-	OpenGAN	ROC-AUC	0.4143	**0.2735**	0.6878	**41.78**
✓	OpenWGAN-GP (ours)	F1 macro	**0.3105**	0.3071	**0.6176**	41.22

switch from *ROC-AUC* to *F1* as the discriminator selection metric against the validation set apparently also did not make a difference in light of the training failure. Since the same *CAFormer-S18* classification model was used to generate the embeddings used by the *OpenGAN* variants shown above, it appears that the improved *Track 1* (and consequently *Track 3*) performance is the result of the addition of the *WGAN-GP* training paradigm to *OpenGAN*.

4.6. Leaderboard performance

Public leaderboard performance is shown in Table 10 for teams that have selected models for private leaderboard evaluation. Private leaderboard performance for the selected models for each team are shown in Table 11. The best performance for each metric is independently reported for each team, which means that results for each team may represent distinct solutions for each metric. My models achieved the best performance for *Track 1* and accuracy in both the public and private leaderboards and the best *F1* for the private leaderboard. My models placed 3rd on the private leaderboard for *Track 2* and 2nd for *Track 3*, indicating that the poisonous → edible misclassification could be improved for the methods presented here.

Table 10
Public leaderboard performance for teams with selected models. The best of each track is independently reported for each team, which means that results for each team may represent distinct solutions for each metric. Accuracy is used for the table ranking.

Rank	Team Name	Track1↓	Track2↓	Track3↓	F1↑	Accuracy↑
1	jack-etheredge (ours)	**0.2394**	0.1357	0.4075	52.08	**76.06**
2	chirmy	0.2641	0.4026	0.6667	46.65	73.59
3	IES	0.2691	**0.0699**	**0.3621**	**56.55**	73.09
4	TingTing1999	0.2734	0.4201	0.6934	44.39	72.66
5	upupup	0.368	0.1348	0.513	54.04	63.2
6	DS@GT	0.395	1.6493	2.0443	27.61	60.5

Table 11
Private leaderboard performance for teams with selected models. The best of each track is independently reported for each team, which means that results for each team may represent distinct solutions for each metric. Accuracy is used for the table ranking.

Rank	Team Name	Track1↓	Track2↓	Track3↓	F1↑	Accuracy↑
1	jack-etheredge (ours)	**0.2436**	0.1613	0.4075	**56.79**	**75.64**
2	chirmy	0.2693	0.4149	0.6667	51.75	73.07
3	TingTing1999	0.2749	0.4378	0.6934	51.42	72.51
4	IES	0.2958	0.0860	**0.3621**	56.41	70.42
5	upupup	0.3882	**0.0718**	0.513	54.80	61.18
6	DS@GT	0.3907	1.6040	2.0443	30.01	60.93

5. Discussion

The proposed methodology for open-set recognition of fungi species addresses the critical challenge of distinguishing between edible and poisonous mushrooms while effectively identifying unknown species. This study demonstrates the potential of combining *Metaformer* and *CAFormer* models to achieve robust classification performance. The integration of metadata in *Metaformer* models significantly enhances the model's ability to leverage additional contextual information, thereby improving classification accuracy. However, one notable challenge is the current evaluation metrics, which assume unknown species are edible. This assumption may not be ideal if the models are intended to be used for foraging contexts, where new poisonous species of mushrooms are continually discovered. Re-evaluating these metrics to consider unknown species as potentially poisonous could mitigate the tension between open-set classification and the misclassification of poisonous species, thereby enhancing the practical applicability of the models in real-world scenarios. The current structure of the metrics which treats unknown species as edible puts open-set recognition and poisonous species identification in direct opposition with each other since misclassifying a closed-set poisonous species as unknown is heavily penalized, making their joint optimization challenging. If the intention is to build a system which displays both high detection rates for unknown species and high recall for poisonous mushrooms, the high penalty for poisonous → edible misclassification would work in favor of rather than against identification of unknown species if unknowns were assumed poisonous instead of edible. This may ultimately improve the model's performance in both aspects.

During *OpenWGAN-GP* training of the open-set discriminator, the same label was used for real features and open-set features. All results shown utilizing *OpenWGAN-GP* in Section 4 represent cases when the open-set label and real label are shared during training of the open-set discriminator. By assigning the same label to real features and open-set features during the supervised and adversarial phases of each update respectively, the generator is incentivized to generate features that are indistinguishable from features created by the closed-set classifier for open-set observations. This is the opposite of the mapping used in the implementation of *OpenGAN*, which used the same closed-set label as the real label, which would have the effect of generating supplemental closed-set features instead. Initial experiments showed that a higher macro-*F1* score was achieved between open- and closed-set validation examples for the FungiCLEF 2024 dataset when the real label was shared with the open-set label rather than the closed-set label. If the open-set label is the same as the real label, as in this work, the generator generates fake open-set features and the discriminator predicts less realistic features as closed-set features. Intuitively, the open-set should be more diverse and thus sharing the label with the inherently less realistic fake generated data seems the more logical choice in most scenarios when greater diversity is expected to be observed in the open-set data.

6. Future work

Future research should explore the redefinition of evaluation metrics to account for the possibility of unknown species being poisonous. This adjustment could reduce the conflict between optimizing for open-set classification and minimizing poisonous species misclassification. Additionally, investigating more sophisticated ensembling techniques and incorporating advanced data augmentation strategies could further improve model performance. Exploring the use of few-shot learning techniques might address the challenge posed by classes with very few observations. Finally, expanding the application of the proposed *OpenWGAN-GP* framework to other domains with similar classification challenges could validate its versatility and robustness.

7. Conclusions

This paper presents a novel method for open-set recognition of fungi species. The integration of *WGAN-GP* training optimizations into *OpenGAN*, resulting in *OpenWGAN-GP*, enhances training stability and enables lightweight discriminators to effectively identify unknown fungi species. An ensemble of *Metaformer* and *CAFormer* models is leveraged to classify fungi accurately while avoiding the misclassification of poisonous mushrooms as edible. The application of carefully chosen test-time augmentations, such as image resolution adjustments, horizontal flipping, and multi-instance averaging, dramatically improves classification performance. These techniques collectively contributed to achieving 1st place in the FungiCLEF 2024 competition for *Track 1*, *F1*, and Accuracy and 2nd place for the final ranking metric *Track 3*, which combines edible → poisonous confusion loss *Track 2* with standardard misclassification loss including the unknown class *Track 1*.

Acknowledgments

The author would like to thank Jillian Etheredge for constructive criticism of the manuscript.

References

[1] H. Li, Y. Zhang, H. Zhang, J. Zhou, Z. Chen, J. Liang, Y. Yin, Q. He, S. Jiang, Y. Zhang, Y. Yuan, N. Lang, B. Cheng, J. Zhong, Z. Li, C. Sun, Mushroom Poisoning Outbreaks — China, 2023, China CDC Weekly 6 (2024) 64–68. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10832152/. doi:10.46234/ccdcw2024.014.

[2] 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.

[3] A. Joly, L. Picek, S. Kahl, H. Goëau, V. Espitalier, C. Botella, B. Deneu, D. Marcos, J. Estopinan, C. Leblanc, T. Larcher, M. Šulc, M. Hrúz, M. Servajean, et al., Overview of LifeCLEF 2024: Challenges on species distribution prediction and identification, in: International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, 2024.

[4] Q. Diao, Y. Jiang, B. Wen, J. Sun, Z. Yuan, MetaFormer: A Unified Meta Framework for Fine-Grained Recognition, 2022. URL: http://arxiv.org/abs/2203.02751. doi:10.48550/arXiv.2203.02751, arXiv:2203.02751 [cs].

[5] G. Van Horn, O. Mac Aodha, iNat Challenge 2021 - FGVC8. Kaggle. (2021). URL: https://kaggle.com/competitions/inaturalist-2021.

[6] G. Van Horn, S. Branson, R. Farrell, S. Haber, J. Barry, P. Ipeirotis, P. Perona, S. Belongie, Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015) 595–604. URL: http://ieeexplore.ieee.org/document/7298658/. doi:10.1109/CVPR.2015.7298658, conference Name: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) ISBN: 9781467369640 Place: Boston, MA, USA Publisher: IEEE.

[7] C. Wah, S. Branson, P. Welinder, P. Perona, S. J. Belongie, The Caltech-UCSD Birds-200-2011 Dataset, 2011. URL: https://api.semanticscholar.org/CorpusID:16119123.

[8] L. Picek, M. Šulc, R. Chamidullin, J. Matas, Overview of FungiCLEF 2023: Fungi Recognition Beyond 1/0 Cost, in: Working Notes of CLEF 2023 - Conference and Labs of the Evaluation Forum, 2023.

[9] A. Krizhevsky, Learning Multiple Layers of Features from Tiny Images, 2009. URL: https://api.semanticscholar.org/CorpusID:18268744.

[10] Y. Le, X. S. Yang, Tiny ImageNet Visual Recognition Challenge, 2015. URL: https://api.semanticscholar.org/CorpusID:16664790.

[11] LeCun, Yann, Cortes, Corinna, Burges, CJ, MNIST handwritten digit database, ATT Labs [Online]. 2 (2010). URL: http://yann.lecun.com/exdb/mnist.

[12] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Ng, Reading Digits in Natural Images with Unsupervised Feature Learning, 2011. URL: https://api.semanticscholar.org/CorpusID:16852518.

[13] S. Kong, D. Ramanan, OpenGAN: Open-Set Recognition via Open Data Generation, 2021. URL: http://arxiv.org/abs/2104.02939. doi:10.48550/arXiv.2104.02939, arXiv:2104.02939 [cs].

[14] D. Hendrycks, K. Gimpel, A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks, ArXiv (2016). URL: https://api.semanticscholar.org/CorpusID:13046179.

[15] D. Hendrycks, S. Basart, M. Mazeika, M. Mostajabi, J. Steinhardt, D. Song, Scaling Out-of-Distribution Detection for Real-World Settings, 2022. URL: https://api.semanticscholar.org/CorpusID:227407829.

[16] S. Vaze, K. Han, A. Vedaldi, A. Zisserman, Open-Set Recognition: A Good Closed-Set Classifier is All You Need?, ArXiv abs/2110.06207 (2021). URL: https://api.semanticscholar.org/CorpusID:238634102.

[17] H. Ren, H. Jiang, W. Luo, M. Meng, T. Zhang, Entropy-guided open-set fine-grained fungi recognition, in: Working Notes of CLEF 2023 - Conference and Labs of the Evaluation Forum, 2023. URL: https://api.semanticscholar.org/CorpusID:264441405.

[18] W. J. Scheirer, A. De Rezende Rocha, A. Sapkota, T. E. Boult, Toward Open Set Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (2013) 1757–1772. URL: http://ieeexplore.ieee.org/document/6365193/. doi:10.1109/TPAMI.2012.256.

[19] 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].

[20] J. Wang, W. Zhang, Y. Zang, Y. Cao, J. Pang, T. Gong, K. Chen, Z. Liu, C. C. Loy, D. Lin, Seesaw Loss for Long-Tailed Instance Segmentation, 2021. URL: http://arxiv.org/abs/2008.10032. doi:10.48550/arXiv.2008.10032, arXiv:2008.10032 [cs].

[21] W. Yu, M. Luo, P. Zhou, C. Si, Y. Zhou, X. Wang, J. Feng, S. Yan, MetaFormer is Actually What You Need for Vision, 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 10809–10819. URL: https://ieeexplore.ieee.org/document/9879612/. doi:10.1109/CVPR52688.2022.01055, conference Name: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) ISBN: 9781665469463 Place: New Orleans, LA, USA Publisher: IEEE.

[22] 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.

[23] X. Wang, D. Kondratyuk, E. Christiansen, K. M. Kitani, Y. Alon, E. Eban, Wisdom of Committees: An Overlooked Approach To Faster and More Accurate Models, 2022. URL: http://arxiv.org/abs/2012.01988. doi:10.48550/arXiv.2012.01988, arXiv:2012.01988 [cs].

[24] W. Yu, C. Si, P. Zhou, M. Luo, Y. Zhou, J. Feng, S. Yan, X. Wang, MetaFormer Baselines for Vision, IEEE Transactions on Pattern Analysis and Machine Intelligence 46 (2024) 896–912. URL: http://arxiv.org/abs/2210.13452. doi:10.1109/TPAMI.2023.3329173, arXiv:2210.13452 [cs].

[25] 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.

[26] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. Courville, Improved Training of Wasserstein GANs, 2017. URL: http://arxiv.org/abs/1704.00028. doi:10.48550/arXiv.1704.00028, arXiv:1704.00028 [cs, stat].

[27] S. Ioffe, C. Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, 2015. URL: http://arxiv.org/abs/1502.03167. doi:10.48550/arXiv.1502.03167, arXiv:1502.03167 [cs].

[28] B. Xu, N. Wang, T. Chen, M. Li, Empirical Evaluation of Rectified Activations in Convolutional Network, 2015. URL: http://arxiv.org/abs/1505.00853. doi:10.48550/arXiv.1505.00853, arXiv:1505.00853 [cs, stat].

[29] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative Adversarial Networks, 2014. URL: http://arxiv.org/abs/1406.2661. doi:10.48550/arXiv.1406.2661, arXiv:1406.2661 [cs, stat].

[30] D. P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, CoRR abs/1412.6980 (2014). URL: https://api.semanticscholar.org/CorpusID:6628106.

[31] 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].

[32] H. Wei, R. Xie, H. Cheng, L. Feng, B. An, Y. Li, Mitigating Neural Network Overconfidence with Logit Normalization, 2022. URL: http://arxiv.org/abs/2205.09310. doi:10.48550/arXiv.2205.09310, arXiv:2205.09310 [cs].

[33] C. Guo, G. Pleiss, Y. Sun, K. Q. Weinberger, On Calibration of Modern Neural Networks, 2017. URL: http://arxiv.org/abs/1706.04599, arXiv:1706.04599 [cs].

[34] S. G. Müller, F. Hutter, TrivialAugment: Tuning-free Yet State-of-the-Art Data Augmentation, 2021. URL: http://arxiv.org/abs/2103.10158. doi:10.48550/arXiv.2103.10158, arXiv:2103.10158 [cs].

[35] P. Chen, S. Liu, H. Zhao, X. Wang, J. Jia, GridMask Data Augmentation, 2024. URL: http://arxiv.org/abs/2001.04086. doi:10.48550/arXiv.2001.04086, arXiv:2001.04086 [cs].

[36] H. Touvron, A. Vedaldi, M. Douze, H. Jégou, Fixing the train-test resolution discrepancy, 2022. URL: http://arxiv.org/abs/1906.06423. doi:10.48550/arXiv.1906.06423, arXiv:1906.06423 [cs].

[37] M. Tan, Q. V. Le, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, ArXiv (2019). URL: https://api.semanticscholar.org/CorpusID:167217261.

[38] M. Tan, Q. V. Le, EfficientNetV2: Smaller Models and Faster Training, in: International Conference on Machine Learning, 2021. URL: https://api.semanticscholar.org/CorpusID:232478903.