Post-Hoc Aggregation as a Competitive Alternative to Model-Centric Pipelines: NEUON Submission to PlantCLEF 2025

Notebook for the LifeCLEF Lab at CLEF 2025

Hamza Ahmed Ishrat[1,2,*], Sue Han Lee[1], Yang Loong Chang[2] and Chai Kok Chin[2]

[1]Swinburne University of Technology Sarawak Campus, 93350, Sarawak, Malaysia
[2]Department of Artificial Intelligence, NEUON AI, 93350, Sarawak, Malaysia

Abstract

This paper discusses our submission to the PlantCLEF 2025 challenge, identical to PlantCLEF 2024, where the objective is to predict multiple plant species within vegetation plot images. The difficulty of this challenge stems from multi-class classification, domain shifts, and predictions on high resolution, evidenced by the macro-averaged F1 score not exceeding 0.37. The training data made available consist of more than 1.4 million single-class images, while the test plot set contains 2,105 high-resolution vegetation plot images taken from above. The plots exhibit multiple domain shifts including blurs, plant life cycle, occlusions (via organic or inorganic matter), and seasonal changes. Given the success of using patch-wise inference from last year's challenge, we opted to continue with this method while also exploring domain-aware pretext tasks to finetune the provided DinoV2 vision transformer to address the domain shifts but yielded limited performance, possibly due to using a limited subset of the training data. In contrast, we performed ablation studies using only the base model, focusing on post-hoc techniques including aggregation and filtering. Surprisingly, we found that we were able to post challenge score of 0.35 macro-average F1; surpassing all our model-centric attempts including all but first place. Relevant code and runs will be made available on GitHub

Keywords

multi-species identification, vegetation plots, vision transformers, bayesian model averaging,

1. Introduction

The PlantCLEF 2025 [1] challenge, part of the larger LifeCLEF 2025 initiative [2], is a continuation of PlantCLEF 2024 [3]; identifying all plant species visible in high-resolution vegetation plot images. Unlike traditional single-species classification, this challenge presents unique difficulties, including multi-label prediction, domain shift between training and test data, and the processing of large, high-resolution images. The difficulty is underscored by the 2024 winning submission achieving a macro F1 score of just 28.73 on average per plot.

The primary difference between the 2025 and 2024 challenge lies in the expanded evaluation set, which now includes 2,105 vegetation plots (up from 1,695). The difficulty, however, remains the same. The most widely adopted and effective strategy has been to divide each high-resolution image into non-overlapping patches, typically 64 or 16 per image although some teams opted for 4,9 and 25 [4], and run inference patch-wise. These predictions are then aggregated to infer the species composition of the whole plot. Some teams have also explored segmentation-based approaches using tools such as Segment Anything Models (SAMs) [5], which help isolate vegetation from background noise and non-organic material

Our own approach [6] combined convolutional neural networks (CNNs) and vision transformers (ViTs) [7], specifically leveraging the DINOv2-based models provided by the challenge organisers [8].

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain
*Corresponding author.
hamza.ishrat@yahoo.com (H. A. Ishrat); shlee@swinburne.edu.my (S. H. Lee); yangloong@neuon.ai (Y. L. Chang); kc@neuon.ai (C. K. Chin)

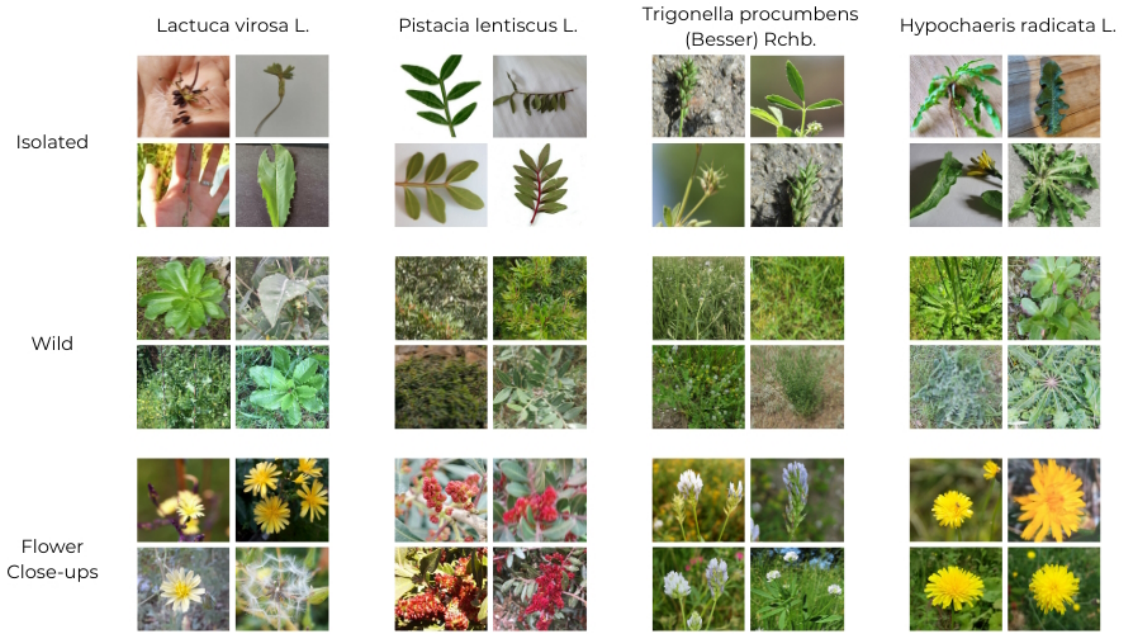

Figure 1: Observations of four different species, *Lactuca virosa* L., *Pistacia lentiscus* L., *Trigonella procumbens* (Besser) Rchb., and *Hypochaeris radicata* L. , showcasing the varying sample types within classes

We employed an aggregation method similar to Bayesian Model Averaging (BMA) [9] to aggregate patch-level predictions, achieving second place in the 2024 leaderboard.

For 2025, we extended our investigation by exploring Self-Supervised Learning (SSL) with pretext training tasks aimed at improving the species-level representation learned by DINOv2. However, none of these modifications surpassed the base model's performance due to a limited subset of data used for fine-tuning the model. This result shifted our focus: rather than changing the model itself, we studied the post-hoc aggregation pipeline in depth. Our ablation experiments demonstrated that, with no model fine-tuning, we could surpass our official submission and achieve a post challenge macro F1 score of 0.3518; a score that would have placed second overall.

2. Dataset

The dataset is reused from the PlantCLEF 2024 [3] challenge, with the exception of a larger vegetation plot set and an additional unlabelled training set, which we have not used.

2.1. Training set

The training set is a subset of the Pl@ntNet training data [10], consisting of 1,408,033 images across 7,806 species predominantly found in South Western Europe. The dataset has a large class disparity, as found in previous PlantCLEF challenges, with some classes exceeding 500 images, while others have less than 10. The images vary from isolated samples to wild samples among other species, including different organs of the plants such as leaves or flowers, as shown in Figure 1. Table 1 explains the training set in more detail. The train dataset provided including both training and testing, amounting to more than 1.3 million images, with an additional 51,194 images separated for validation.

2.2. Test set

The test set consists of 2,105 high-resolution vegetation plot images (from 2000 to 4000 pixels per side), significantly more as compared to PlantCLEF 2024 [3] which only had 1,695, taken by experts in multiple ecological contexts from Pyrenean and Mediterranean floras. The plot images are taken

Table 1
Single class training set

Dataset	Subset folder	No. of images	No. of species
Train	train + test	1,356,839	7,806
Validation	val	51,194	5,912

Figure 2: Observations of two plots, CBN-Pla-F4 and CBN-Pla-E6, displaying temporal domain shifts as plants wither and regain vitality

from above, though inconsistent and may vary slightly in elevation as opposed to being strictly taken overhead. The vegetation plots exhibit a variety of domain shifts, such as seasonal where plants maybe withered as shown in Figure **??**. Other domain shifts include occlusions by measuring tools, rocks, or other plants, visual blurs due to motion or shadows. An important distinction is that a single vegetation plot may have from 1 to many species within them, such is the nature of the challenge as it differs from the training set. The individual species may themselves be in different stages of their life-cycle adding another layer of complexity to this challenge.

3. Methodology

Following our work on PlantCLEF 2024, we chose to continue our work using the vision transformers provided by the organizers [8]. Two models were provided, both based on the DinoV2 [11] architecture and pretrained on the PlantClef 2025 dataset, the difference being one only had the classifier heads trained while the other continued training of the entire backbone as well. We opted to use the fully trained model as the backbone for all our models and attempts. Due to the competitive nature and exploratory scope of the task, this report presents both official results and an extensive series of post-evaluation experiments. Owing to time constraints, not all results were submitted officially. These will be clearly marked and discussed accordingly.

A tiling approach, similar to multiple teams last year, was also implemented to infer species on the entire vegetation plot by splitting the image into 64 or 16 patches and inferring independently, and

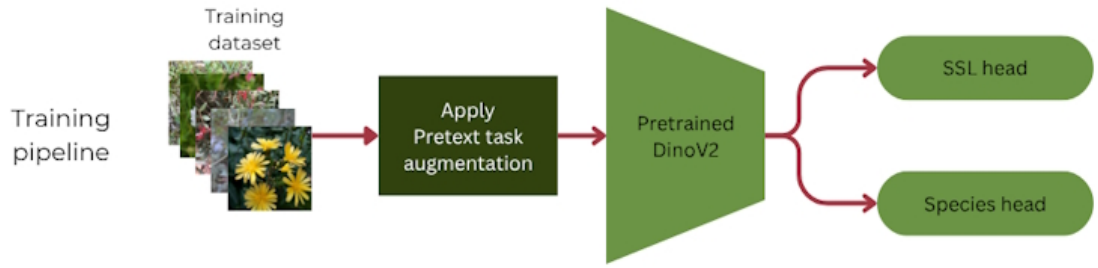

Figure 3: Our training pipeline which involves applying a pretext task and a dual head classifier, one for species and the other to classify which variant of the pretext task was applied

Table 2
Details of our training pipeline

Hyperparameter	Values
Image input size	$244 \times 244 \times 3$
Epochs	50
Unfreeze layer	20
Batch size	128
Initial heads learning rate	0.05
Post un-freezing heads learning rate	0.005
Backbone learning rate	0.0001
Weight decay	0.00001
Momentum	0.9
Optimiser	SGD
Loss function	Cross Entropy

then aggregating them in post to get the overall results for the entire plot, as will be discussed in the Inference method section.

The dataset we used was the training set provided, however we chose to limit each class to a maximum of 50 images per class as to combat the class imbalance, as well as computational constraints. Although 50 is low for amount of samples, we employed a higher than normal learning rate to accelerate training, as will be discussed in our Architecture section.

3.1. Architecture and training

Our base architecture consists of two parts; the backbone, which is the aforementioned pretrained DinoV2 model, as well as classifier heads using the hyperparameters described in Table 2. We opted to use pretext tasks as a method to finetune our models with two heads, one for species classification and the other for pretext labels, as shown in Figure 3. Both heads were just a simple linear layer. The classifier heads were trained first for 20 epochs while the backbone was frozen, then for the remaining 30 the full model was trained with the last two layers of the backbone unfrozen at a lower learning rate.

3.2. Pretext tasks

Pretext tasks, also known as auxiliary tasks, are intermediate tasks a model performs in tandem with the main objective task . Typically employed in Self-Supervised Learning (SSL), it serves as to reinforce the model's internal representation of classes by evaluating different versions of an image which undergo transformations, such as rotation [12], jigsaw [13] or colourisation [14].

SSL tasks have been used in the vegetation domain [15] as well as plant-disease identification using translational augmentations [16], contrastive learning [17] and auto-encoders [18]. However in our case, spatial augmentations such as rotation or flipping were deemed less beneficial as vegetation plot

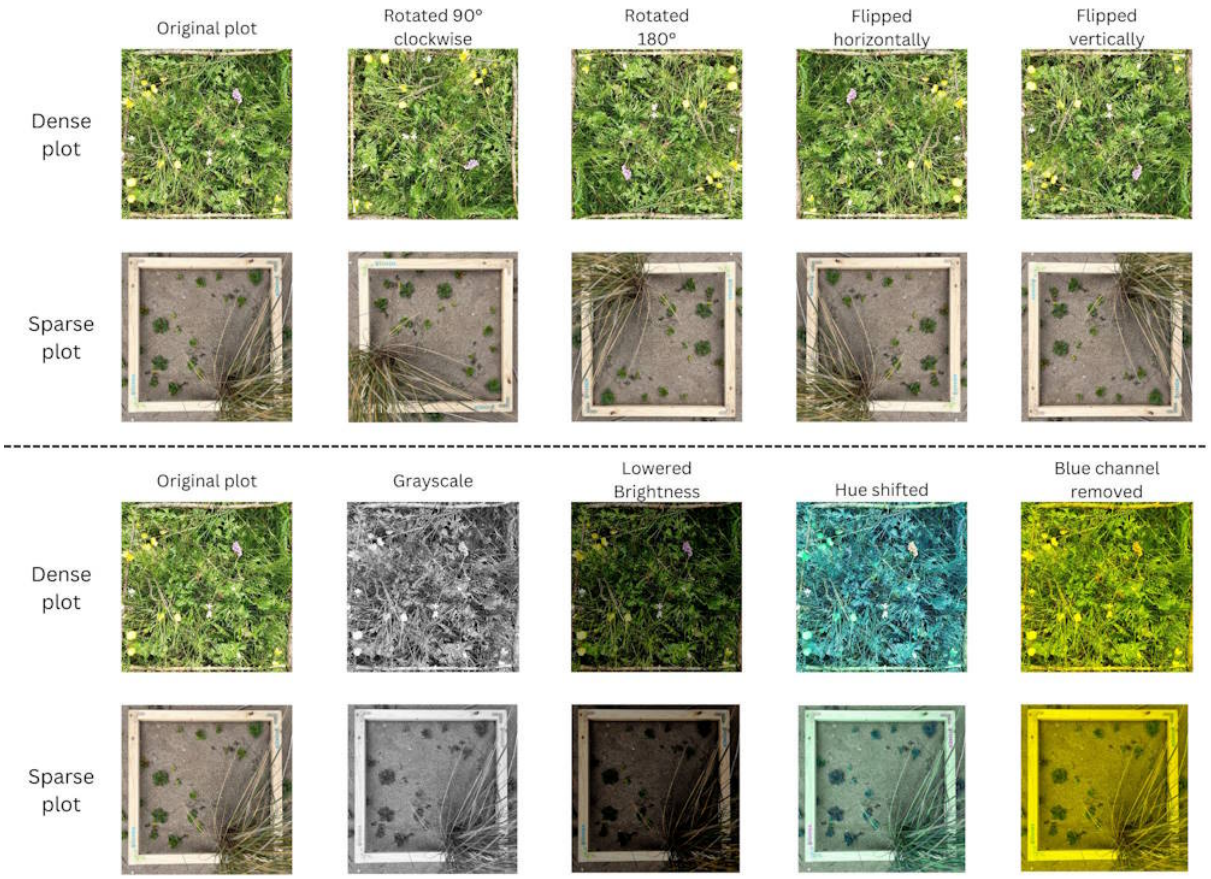

Figure 4: Top: Different spatial transformations applied to both dense and sparse plots, showing that such augmentations do not effect on a larger, plot level scale. Bottom: Different chromatic transformations, showcasing the broader range and capacity of utilising colour as an augmentation

images are largely invariant to such transformations. Instead, we focused on colour-based pretext tasks, motivated by the assumption that robust species recognition requires sensitivity to chromatic variation across different plant life stages. As shown in Figure 4, spatial transformations do little to add any variety as the overall semantic relationship between species is unchanged. Regardless of flipping or rotation and regardless of sparse or dense plots, plant density is preserved, as well as inter-spatial relationships between the plants themselves. These transformations do not meaningfully change the species-level semantic content, which limits their usefulness as supervision signals. Chromatically however, contains strong species-level cues. Removing or distorting colour degrades visual distinctions between species, unlike geometric transforms. Thus, colour-based augmentations or tasks may better guide self-supervised feature learning. Previous attempts of exploiting the chromatic space as opposed to the translational space have proven fruitful [19], although to our knowledge none have been applied specifically to the plant domain. We hypothesize that a model that learns to associate a species with its colour variants, such as discoloured leaves due to ageing or shading, may generalize better under natural variations found in test plots. Examples of the pretext tasks used are shown in Figure 5. Each pretext task was trained with its own model; the training pipeline does not incorporate more than one pretext task at a time. We note that the pretext tasks were applied only to the training set as shown in Figure 5 and not on the vegetation plots.

3.2.1. RGB elimination

In this pretext task, we exploit the three primary colour channels: red, green, and blue, by selectively removing one of them from an image and training the model to predict which channel was eliminated.

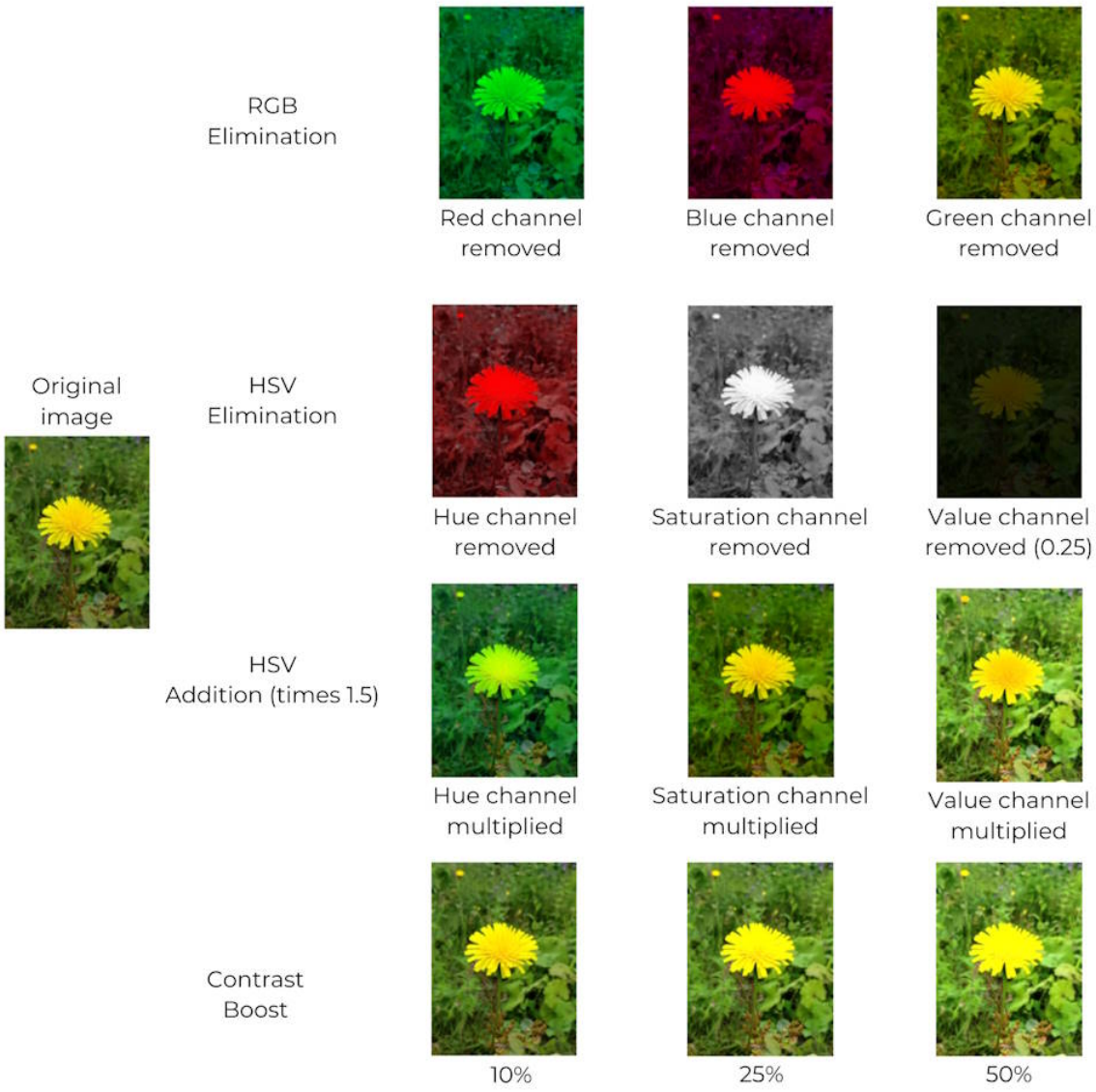

Figure 5: Visualisation of different SSL tasks used in training

This yields four possible classes for the task: red removed, green removed, blue removed, and no modification. Only one channel is removed at a time per image.

The underlying motivation is to simulate chromatic degradation or variation that may occur in real-world vegetation imagery due to environmental conditions such as lighting, seasonal changes, or plant life cycle stages. By forcing the model to distinguish which colour component is missing, we hypothesize that it will learn more robust internal representations that are invariant to certain types of colour-based distortions.

3.2.2. HSV elimination

Building on the RGB elimination task, we also explore a secondary chromatic representation: the HSV colour space: Hue, Saturation, and Value. This colour space is often more aligned with human perception of colour properties. Hue refers to the dominant wavelength of a colour (e.g., red, green, blue), essentially defining its "type." Saturation indicates the intensity or purity of the colour, where higher values correspond to more vivid colours. Value represents brightness, with lower values producing

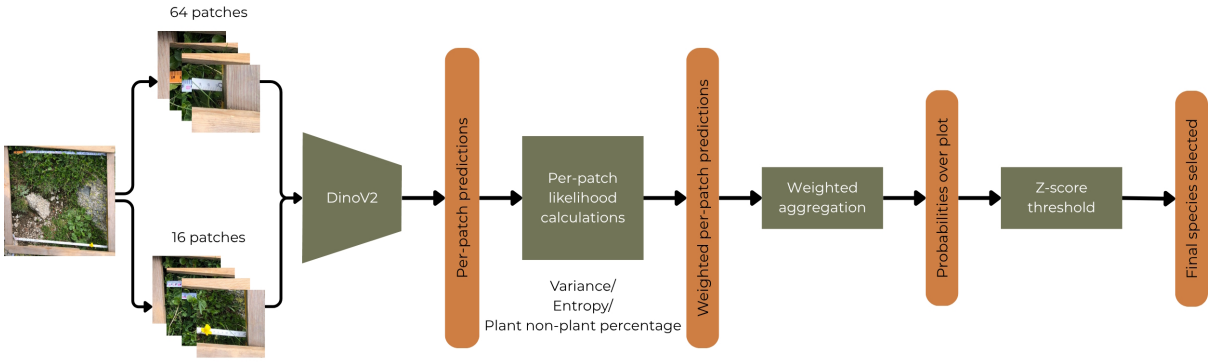

Figure 6: Overall inference method for an image.

darker tones, and zero resulting in black, though for our runs we set a limit of 0.25 to avoid a completely black image.

Here we selectively eliminate one of the three HSV components at a time, setting the corresponding channel to zero and train the model to predict which component has been removed. The objective is not to alter colour directly (as with RGB), but to manipulate properties such as darkness and vibrancy. Our motivation is that by exposing the model to colour variants that simulate environmental effects like shadows (low value) or faded pigmentation (low saturation), it may develop more invariant and generalizable feature representations of plant species under diverse conditions.

3.2.3. HSV addition

The direct inverse of HSV elimination, where the value of the channels are multiplied by 1.5 to increase Hue, Saturation and Value.

3.2.4. Contrast boost

For this pretext task, we apply varying levels of contrast enhancement to the input image and task the model with identifying the applied contrast level. The contrast adjustment is implemented via a simple scaling of pixel intensities, where the image is multiplied by a factor ranging from no change (1.0×) to a significant boost (up to 1.5×, i.e., +50% contrast). The task includes four possible classes: unchanged, +10%, +25%, and +50%.

This task aims to teach the model to become sensitive to intensity-based variations that may occur in real-world scenarios such as overexposure, harsh lighting, or high reflectance from leaves or soil. By learning to recognize plant structures under different contrast levels, the model can develop robustness to varying imaging conditions and enhance its feature extraction across heterogeneous lighting environments.

3.3. Inference method

In line with our previous approach from PlantCLEF 2024, we continue to use patch-wise inference as the primary method for prediction. Each vegetation plot image is first divided into multiple smaller patches, both 64 and 16 per plot. These patches are then passed individually through the model to generate class-wise predictions. The outputs are aggregated using a Bayesian Model Averaging [9] scheme, where predictions from each patch are treated as individual, noisy estimations of the true class distribution. These are then combined to form a more confident and robust final prediction per plot.

3.3.1. Bayesian Model Averaging

Bayesian Model Averaging (BMA) typically combines multiple models by weighting their outputs according to their posterior probabilities given the observed data. It consists of two key components: the likelihood of each model M_k given the data D, and the prior probability of the model M_k before observing the data.

$$P(M_k|D) = \frac{P(D|M_k) \times P(M_k)}{\sum_{l=1}^{K} P(D|M_l) \times P(M_l)} \tag{1}$$

Here $P(D \mid M_k)$ is the likelihood measuring how well M_k explains the data D, $P(M_k)$ is the prior, representing the assumed probability of M_k before seeing any data, and the denominator is the sum of all models K product of their respective likelihoods and priors.

An important distinction to make is that though BMA is used for ensemble models, we applied its weighting calculations for our case, substituting model performance for prediction confidence.

This aggregation method is particularly useful in our case, as it inherently handles uncertainty across patches. Moreover, due to the multi-label nature of the task, we compute a posterior probability for each class independently across patches, assuming conditional independence between them. Figure 6 illustrates our overall inference for an image in detail.

This method remains unchanged from our 2024 submission [6] due to its proven reliability and performance across unseen domains and its interpretability in post-hoc adjustments, however we do alter how the likelihood is calculated.

Similarly we also incorporated using z-score as the threshold. Equation 2 describes the z-score, which is a measure of how many standard deviations a data point is from the mean of a set.

$$z = \frac{x - \mu}{\sigma} \tag{2}$$

Where x is the data point, μ is the mean and σ is the standard deviation of the set. Typically used to find outliers, we implemented this calculation to find any classes that are significant enough to be considered as present in a plot and as such we set our threshold to 2.

3.3.2. BMA Likelihood

As stated in our previous working notes [6], we note that since we only use one model, the prior would be irrelevant as it is based on the model's probability before seeing any data. The likelihood then would act as the weight to our aggregation as prior would be cancelled out in calculations. Assuming we use the number of patches, N, as the prior, it would then be $\frac{1}{N}$ for each patch. Denoting L_k as the likelihood of a patch k:

$$P(L_k|D) = \frac{L_k \times \frac{1}{N}}{\sum_{n=1}^{N} L_n \times \frac{1}{N}} \tag{3}$$

This can then be simplified to:

$$P(i|D) = \frac{\frac{L_k}{N}}{\frac{\sum_{n=1}^{N} L_n}{N}} \tag{4}$$

$$P(i|D) = \frac{L_k}{\sum_{n=1}^{N} L_n} \tag{5}$$

Leaving the final probability of a patch k as a fraction of the sum of all likelihoods of all patches. We chose 3 metrics to calculate the likelihood of the patches, namely variance, entropy, and plant percentage. Each method was used one at a time, using each method to assign a weight to each patch in a plot.

3.3.3. Variance

Variance is a statistical term that describes the spread of a given data based on the standard deviation and mean of the set. Typically, the higher the variance, the higher the spread and the more skewed the set is.

$$s^2 = \frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N-1} \tag{6}$$

Equation 6 describes variance where s^2 is the sample variance, N is the number of samples and $\bar{x}$ is the mean of all the samples. Hence variance was used as a proxy for how 'confident' the model is; a higher variance suggests a few classes score disproportionately higher than the rest, therefore the variance of the set as a whole would increase, and we can assume the model is confident. Likewise, if the model is confused and produces a probability distribution that is less exaggerated or more spread out, we assume the model is not confident.

Typically variance values are in the range of 0 to 1, and in our case typically less than 0.01. To amplify the score we used the absolute common log ($\log_{10}$) of the variance to get a range that is easier to work with as in Equation 7.

$$var = |log_{10}(s^2)| \tag{7}$$

Another challenge would be mapping the variance to a usable range. while normalizing all the values could be an option, we opted to use a custom curve that would penalize higher values less as shown in Equation 8

$$y = \sqrt{\frac{a + 0.5 - x}{a + 0.5}} \tag{8}$$

Where a is the maximum absolute log of variance across all patches and x is the absolute log of variance of a given patch

This would provide a mapping function from the absolute log of variance to a confidence score, which could then be used as a likelihood for the patch.

3.3.4. Entropy

Entropy, or Shannon Entropy [20], is another confidence metric we use to evaluate prediction uncertainty across image patches. Unlike variance, which captures dispersion across patch predictions, entropy focuses on the internal uncertainty within each prediction itself. The assumption is similar to using variance; a plot that has more distinct species will cause a spike in the probability distribution, thus increasing its entropy. Given a probability distribution $P = \{p_0, p_1 ... p_n\}$, entropy is calculated as:

$$H(P) = -\sum_{n=1}^{N} p_n log(p_n) \tag{9}$$

The entropy of the patch's distribution is then used directly as the likelihood.

3.3.5. Plant percentage

In an attempt to classify a patch as truly plant based, we developed a small regression model that would give a score from 0 to 1. The goal was to have the model distinguish between organic (plants) and inorganic (rulers, wooden planks, rocks, dirt) and produce a score for the patch as a whole. The training set was 3000 images taken randomly from all the patches across all plots, specifically where each plot is split into 64 patches. Using a custom labelling software, shown in Figure 7, a patch's plant percentage was estimated by overlaying an 8 by 8 grid onto the patch, and calculating the percentage of squares contain plants.

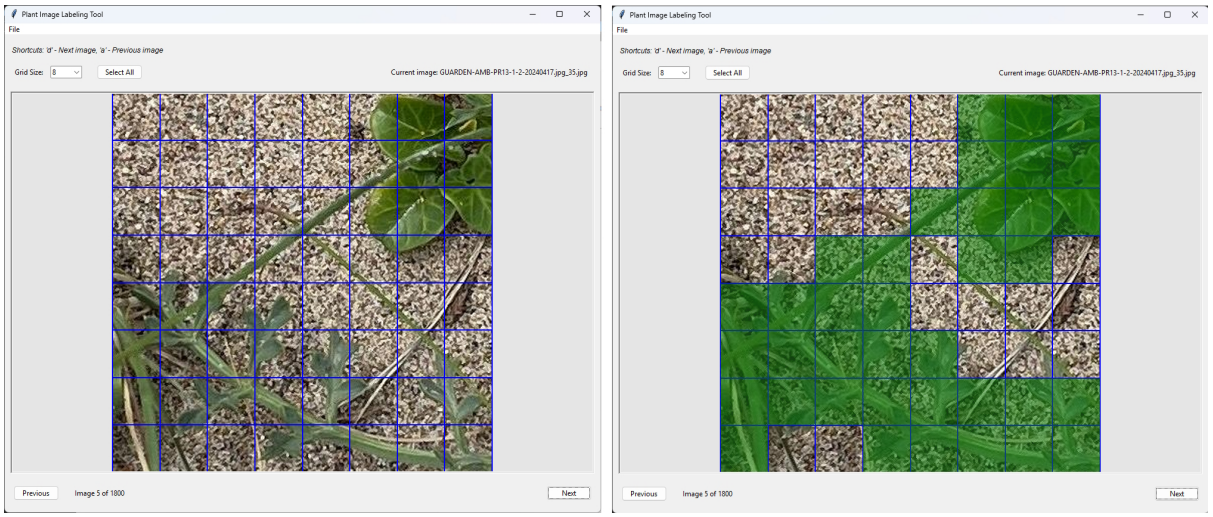

Figure 7: Overview of the labelling software showing the grid for manual annotation, with the image on the right showing the highlighted squares where a plant is visible

Table 3
Details of our regression models

Hyperparameter	Values
Epochs	100
Batch size	32
Learning rate	0.0001
Optimiser	SGD
Loss function	MSELoss
Evaluations	R^2 and MAE

Table 4
Testing results on the various regression models

Model	R^2	MAE
ResNet50	0.8138	8.1912
MobileNet	0.7911	9.4496
MobileViT	0.7004	10.9209

3 models were tested, namely ResNet50 [21], MobileNet [22] and MobileViT [23], all following the same training parameters as shown in Table 3, with their performance in Table 4. Ultimately we decided on using the ResNet50 model as it performed the best overall. Figure 8 shows the results on a patch using this model, where the clearer the square is the higher the plant percentage The output of the regression model was then used as the likelihood.

4. Submissions

Due to overlapping commitments within the limited competition timeline only 5 runs were officially submitted. Originally there were a planned 17 including the official runs, however the remaining 12 were submitted after the deadline and will be included but noted as unofficial scores.

4.1. Evaluation metrics

The evaluation metric used was macro-averaged F1 score per sample, to balance both false positives (incorrectly predicting species that are not present) and false negatives (not predicting species that are

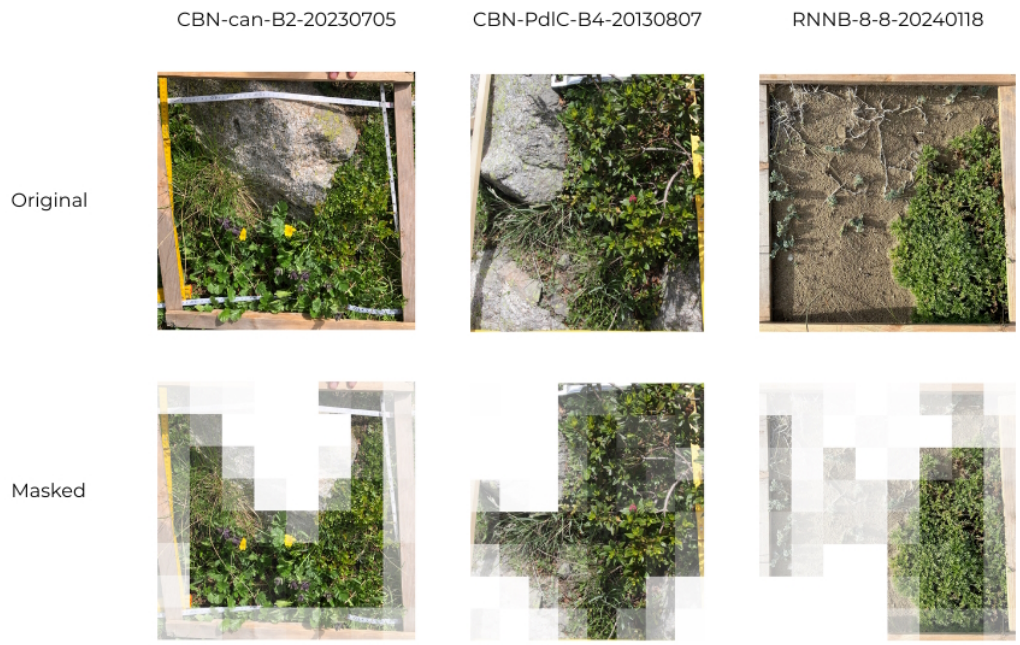

Figure 8: Visualisation of the regression model on different plots, the clearer the square the higher the probability a plant exists within the square

present), using Precision and Recall. Equation 10 describes how macro-averaged $F1$ is calculated

$$Macro\ avg \quad F1 = \frac{1}{N}\sum_{i=1}^{N}(\frac{1}{T_i}\sum_{j=1}^{T_i}F1^j) \tag{10}$$

Where $F1^j$ is calculated by:

$$F1 = \frac{2 \times Precision_j \times Recall_j}{Precision_j + Recall_j} \tag{11}$$

Precision (Equation 12) serves to calculate how many true positives TP were predicted from a full set of true positives TP and false positives FP, while Recall (Equation 13) relates to how many true positives TP were predicted from a full set of true positives TP and false negatives FN

$$Precision = \frac{TP_j}{TP_j + FP_j} \tag{12}$$

$$Recall = \frac{TP_j}{TP_j + FN_j} \tag{13}$$

Two scores were produced: a public score that was calculated on 11% of the test data during the competition, and the private score which was shown for all teams at the end of the competition using the remaining 89%.

4.2. Model and runs

Given the 4 pretext tasks discussed and an additional base configuration where the model was taken as-is, and the 3 methods of calculating likelihood, 15 runs were produced. We also submitted a baseline using the base configuration and only using 64 patches per plot, whereas all the runs used an both 64 and 16 patches per plot, as well as another baseline which utilised of simple aggregation, which added

Table 5
Results of our submitted runs. Official runs are marked with an asterisk (*) and bolded. Note that the suffix p_np refers to the Plant Percentage (plant_non plant) calculations for likelihood

Run	Private Score	Public Score
dinov2_base_variance	0.34318	0.31732
dinov2_simple_aggregation	0.33856	0.31426
dinov2_base_entropy	0.3385	0.31432
dinov2_base_p_np	0.33024	0.32458
baseline*	**0.31457**	**0.30619**
dinov2_hsv_add_variance	0.29286	0.29295
dinov2_hsv_add_entropy	0.29161	0.2901
dinov2_rgb_elim_entropy*	**0.28822**	**0.2918**
dinov2_rgb_elim_variance*	**0.28801**	**0.29349**
dinov2_hsv_elim_variance*	**0.28579**	**0.28062**
dinov2_hsv_elim_p_np	0.28572	0.28547
dinov2_rgb_elim_p_np	0.28517	0.28689
dinov2_hsv_elim_entropy	0.28501	0.28219
dinov2_hsv_add_p_np	0.27909	0.2975
dinov2_contrast_variance*	**0.2408**	**0.28983**
dinov2_contrast_entropy	0.24051	0.28769
dinov2_contrast_p_np	0.23229	0.28444

up all the class probabilities across all plots and applied the threshold. Table 5 shows the results of the 17 runs.

5. Results

Among our official submissions, the highest-scoring model, ironically, was our baseline, placing 7th overall. Our other runs placed 13th through 15th, with the lowest being dinov2_contrast_variance at 18th. Due to limited time and resources, we were unable to submit our strongest runs during the competition window. Frustratingly, these also turned out to be simple base models. One such post-hoc run using straightforward aggregation with the provided ViT model achieved a macro-F1 score of 0.33856, which would have outperformed the official 3rd place. The strongest unofficial result came from dinov2_base_variance, scoring 0.34318, again placing at 3rd had it been submitted.

6. Ablation study

Given the success of our best, albeit unofficial, run of dinov2_base_variance, we opted to explore the parameters that may increase its performance. For this run we would like to emphasise that no model tuning was done, and the only parameters were related to post processing the model's prediction. All results are derived using the pre-trained model provided by the organisers and the variation lies solely in the aggregation of patch-level predictions.

For every plot, both 64 and 16 patch per plot predictions were obtained and aggregated using our Bayesian Model Averaging (BMA) framework. In this setup, we employed variance as the model likelihood, motivated by its consistent performance; it outperformed both entropy and plant-percentage based measures in 4 out of 5 models, with only a marginal exception (dinov2_rgb_elim_entropy, outperforming by 0.00021).

Our motivation was straightforward: since all finetuned models underperformed the baseline, we hypothesized that substantial gains could be made by tuning post-processing alone, treating the model as fixed and optimizing everything around it.

Accordingly, we explored a range of hyperparameters within the aggregation pipeline to better understand their influence on performance.

6.1. Parameters

6.1.1. Logit type

We first identified the inputs to the pipeline. In prior runs we relied on prediction probabilities after softmax has been applied, which normalises all logits, across 7,806 classes, from 0 to 1. However we argued that if the disparity between the probabilities of the highest predicted class and the lowest predicted class is too large, it may exaggerate some predictions, particularly in the lower end of top-n, to be included into the final scores which may be false negatives.

To address this, we experimented with using raw logits directly. Instead of applying softmax across all classes, we first sorted the raw logits per class in descending order and then applied softmax to a truncated subset. This allowed for finer control over the distribution of logits within the top-k classes without distorting the relative magnitude of the raw outputs.

6.1.2. K values

Following the adjustment to logits, we examined the effect of varying the number of top predictions (k) considered per plot. Under the assumption that typical patches may contain fewer than 10 distinct species, we hypothesized that limiting k could reduce prediction noise. All our previous runs used a k-value of 100. We evaluated k values of 100, 500, 1,000, and 5,000

6.1.3. Z-score

Finally, we investigated the z-score threshold used in the final selection step. While a threshold of 2.0 was initially chosen as a conventional outlier cutoff, this value was largely arbitrary. We suspected that adjusting this threshold might allow borderline-relevant classes to be included in the final predictions.

To evaluate this, we varied the threshold from 1.0 to 2.0 in increments of 0.1. This allowed us to assess whether loosening the strictness of the filter could yield improvements without introducing excessive noise.

Another hyperparameter does exist regarding z-score, pertaining to how many values out of the final set are considered. Our previous runs use 100 (similar to k-value) to calculate the z-scores, however for these experiments we opted to keep it at 100 regardless if k is higher than 100.

6.2. Results

To assess the impact of the identified hyperparameters, we conducted a full permutational sweep across all configurations, resulting to 88 distinct runs, with top 10 shown in Table 6.. The following section outlines key performance trends and notable takeaways.

The ablation results were surprising; not only did the best-performing configuration exceed expectations, but it also significantly outperformed the second-ranked configuration by a margin of 0.00639 in macro-F1 score, and was only 0.01351 behind the top-performing entry overall. Notably, these results were achieved without any model fine-tuning or architectural changes; purely through adjustments to post-processing parameters. This highlights the potential of post-hoc methods, particularly when starting from a strong pretrained model.

Among the hyperparameters, the z-score threshold had the most pronounced effect. The performance curve was smooth and showed a clear optimum around z = 1.6 as shown in Figure 9, validating the idea that threshold tuning alone can yield substantial gains.

In contrast, varying the k-value (i.e., the number of top predictions considered) showed less consistent trends as seen in Figure 10. While higher k-values (e.g., k = 5000) produced some of the top-performing runs, suggesting increased stability, other top-10 runs were found even at k = 100. This lack of a clear

Table 6
Top 10 scores in our ablation study and their configurations, with 0.35128 being the highest score achieved

Run	Logit	K-value	Z-score	Private Score	Public Score
exp_RAW_k5000_z1.6.csv	RAW	5000	1.6	**0.35128**	0.32586
exp_SOFTMAX_k5000_z1.6.csv	SOFTMAX	5000	1.6	0.35126	0.32586
exp_SOFTMAX_k100_z1.6.csv	SOFTMAX	100	1.6	0.35103	0.32174
exp_RAW_k1000_z1.5.csv	RAW	1000	1.5	0.35080	0.32365
exp_SOFTMAX_k1000_z1.6.csv	SOFTMAX	1000	1.6	0.35076	0.32578
exp_SOFTMAX_k500_z1.6.csv	SOFTMAX	500	1.6	0.35055	0.32578
exp_RAW_k100_z1.6.csv	RAW	1000	1.6	0.35051	0.32231
exp_RAW_k500_z1.6.csv	RAW	500	1.6	0.34935	0.32057
exp_RAW_k500_z1.5.csv	RAW	500	1.5	0.34925	0.31851
exp_SOFTMAX_k5000_z1.7.csv	SOFTMAX	5000	1.7	0.34898	0.32177

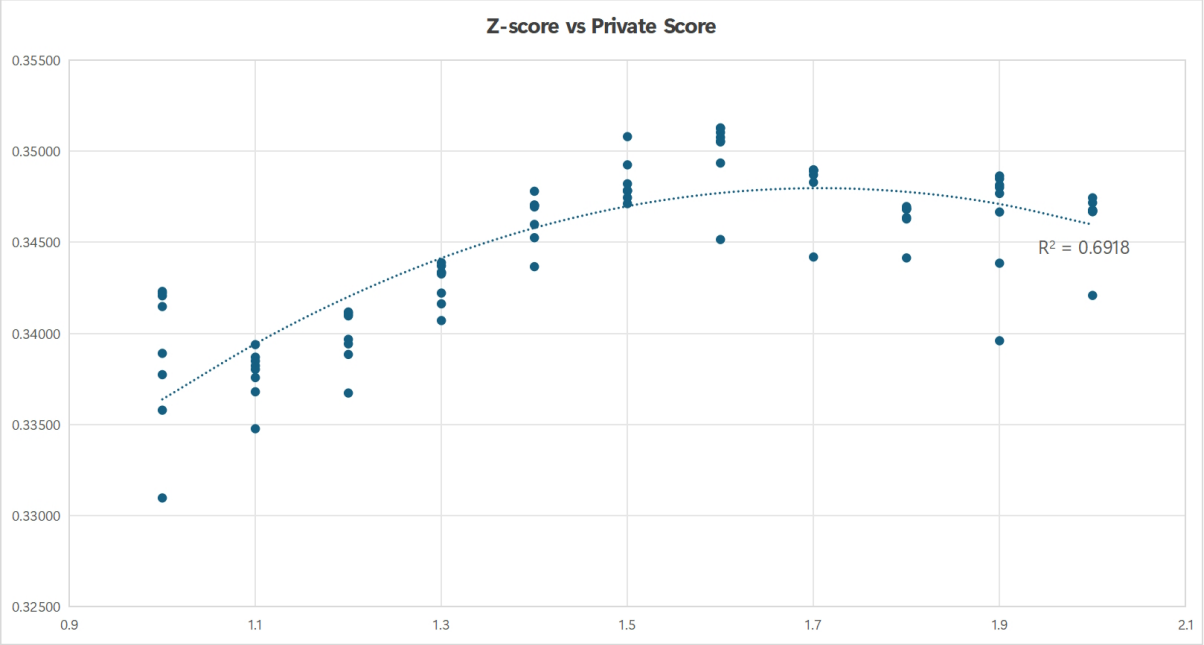

Figure 9: Visualisation of the impact of z_scores on private scores, with 1.6 yielding the highest results

pattern suggests that the optimal k might be more sensitive to interactions with other parameters, rather than being independently influential.

Regarding logit type, there was no conclusive advantage between using raw logits versus softmaxed probabilities. The top 10 runs included a near-even split between the two approaches (5 raw, 5 softmax), and this ratio remained balanced in the top 20 (11 softmax vs. 9 raw), indicating that neither consistently outperformed the other.

All results, including full parameter sweeps, are provided in the Appendix for further inspection.

6.3. Discussion

One of the most surprising outcomes of our study was how often the baseline model outperformed pretext task driven models. Despite applying a range of pretext tasks and transformations, as well as other competitors, it was consistently the base DinoV2 model with variance-based post-processing that yielded the highest scores. This outcome could, however, be due to the foundation itself was simply too limited to benefit meaningfully from added complexity, either due to architecture or the subset of the dataset used for training. Our suspicion leans toward the latter. Without a more robust or discriminative model backbone, the benefits of post-hoc refinements seem capped, regardless of how effective the

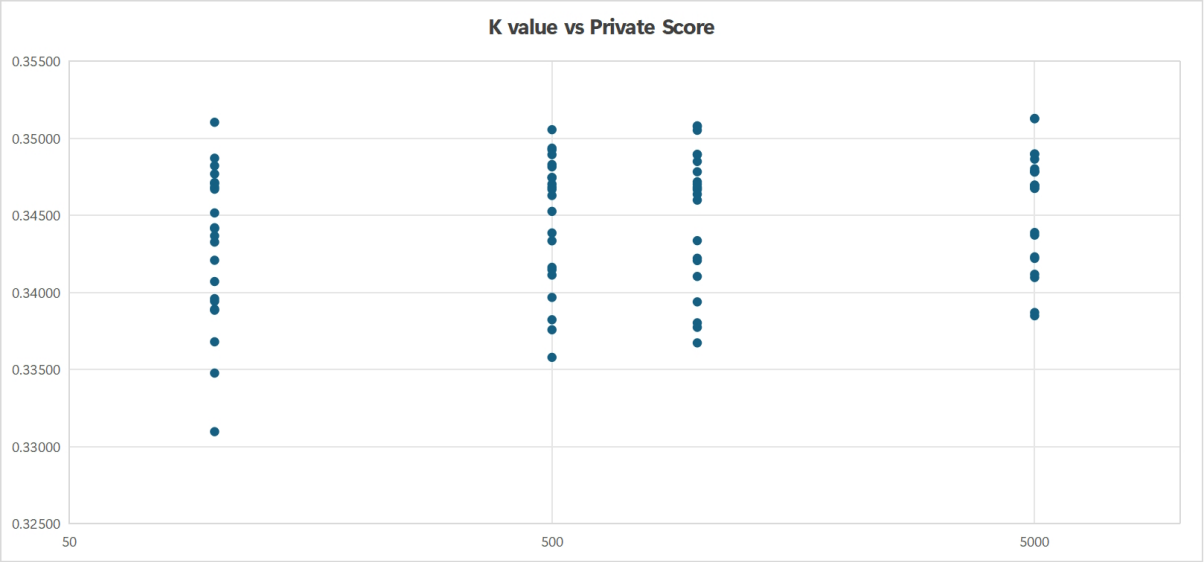

Figure 10: Visualisation of the impact of k_value on private scores

aggregation or filtering strategy may be.

This realization brings us to an important point: post-hoc methods may have their strongest impact when paired with an already competitive model. In our case the organiser-provided model, despite being a provided baseline, performed significantly better once we applied Bayesian model averaging and variance driven confidence filtering. Even so, simple aggregation as in the case with dinov2_simple_aggregation already posted a competitive score with 0.33856. This suggests a compelling argument for future work: using strong, pretrained models as fixed feature extractors and then relying entirely on lightweight inference-stage adaptations. This approach is particularly attractive in resource-constrained or low-data settings, where full fine-tuning is infeasible.

A particularly illustrative failure case was the plant/non-plant binary classifier model. Despite being trained on a dedicated, hand-labeled dataset of over 3,000 examples, the model failed to surpass even the baseline in final leaderboard scores. One possible reason is that the classifier became too certain; overconfident in rejecting ambiguous or noisy predictions. While this may reduce false positives, it also risks discarding correct, albeit uncertain, labels. Another reason we suspect is that the regression model assigns scores of '0', which results in the patch's predictions being nullified completely where as a lesser penalty might still include the predictions at a lower weight.

In contrast, our post-hoc confidence methods (e.g., variance-based z-score filtering) allowed for just enough uncertainty to admit potentially correct outliers. This demonstrates a kind of useful noise tolerance, where not everything that is low-confidence should be discarded outright. Also evidenced by higher k-values where predictions, while intuitively would have been discarded due to perceived irrelevance, may still contribute indirectly by amplifying classes that are certainly present.

Finally, the broader implication of these findings is a shift in emphasis. Rather than focusing solely on making the model better through training, it may be equally (if not more) productive to improve what we do after the model makes its predictions. Given that state-of-the-art vision transformers are increasingly capable out of the box, enhancing inference strategies might yield significant gains with far less computational cost. Especially in large-class multi-label problems like this one, confidence-aware methods provide a valuable mechanism for navigating uncertainty without retraining from scratch.

7. Conclusion

This work explored the potential of post-hoc methods to enhance prediction performance in the PlantCLEF 2025 challenge. Without altering or fine-tuning any model architecture, we investigated

whether strategic filtering, ranking, and aggregation could push standard models beyond their baseline performance. The answer, surprisingly, was yes, and by a notable margin.

Although regrettably due to constraints we were not able to fully test the extent of fine tuning models or exploring SSL tasks more in depth, we opted to leave them in due to our motivations of using crafted pretext tasks for ecological applications, and provide a starting line for future work exploring these methods.

Using the organiser-provided DINOv2 ViT backbone, we implemented a lightweight post-processing pipeline based on Bayesian model averaging, z-score filtering, and class ranking by variance-derived confidence. This method alone was sufficient to outperform more complex pretext task-based models, including those that incorporated domain-specific augmentations or external classifiers. Most notably, our best-performing unofficial run would have placed 2nd overall on the leaderboard, beating out fine-tuned models despite relying solely on inference-stage modifications.

These results affirm a broader insight: post-hoc methods are an underappreciated tool in large-scale, large-set classification. While it is common to attribute performance gains to deeper networks or larger datasets, we show that thoughtful post-processing of predictions, particularly when incorporating uncertainty measures, can yield comparable if not superior improvements. Crucially, this comes with minimal computational burden and no retraining costs.

Looking ahead, future work may explore this further in two directions. First, by pairing post-hoc strategies with stronger or more targeted pretrained backbones, we may unlock even more performance. Second, by expanding the space of inference-time adaptations—perhaps incorporating learned priors, spatial or temporal context, species co-existence or adaptive thresholds, we might approach state-of-the-art results with surprisingly simple setups.

8. Declaration on Generative AI

During the preparation of this paper, the author(s) used ChatGPT in order to paraphrase, reword and to check grammar. After using this GenAI service, the author(s) reviewed the contents of the generated content and take(s) full responsibility for the publication's content.

Acknowledgments

The resources of this project is supported by NEUON AI SDN. BHD., Malaysia.

References

[1] G. Martellucci, H. Goëau, P. Bonnet, F. Vinatier, A. Joly, Overview of PlantCLEF 2025: Multi-species plant identification in vegetation quadrat images, in: Working Notes of CLEF 2025 - Conference and Labs of the Evaluation Forum, 2025.

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

[3] A. Joly, L. Picek, S. Kahl, H. Goëau, V. Espitalier, C. Botella, D. Marcos, J. Estopinan, C. Leblanc, T. Larcher, 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, pp. 183–207.

[4] M. Gustineli, A. Miyaguchi, I. Stalter, Multi-Label Plant Species Classification with Self-Supervised Vision Transformers, 2024. URL: http://arxiv.org/abs/2407.06298. doi:10.48550/arXiv.2407.06298, arXiv:2407.06298 [cs].

[5] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al., Segment anything, in: Proceedings of the IEEE/CVF international conference on computer vision, 2023, pp. 4015–4026.
[6] S. Chulif, H. A. Ishrat, Y. L. Chang, S. H. Lee, Notebook for the LifeCLEF Lab at CLEF 2024, CEUR-WS (2024).
[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).
[8] H. Goëau, J.-C. Lombardo, A. Affouard, V. Espitalier, P. Bonnet, A. Joly, PlantCLEF 2024 pretrained models on the flora of the south western Europe based on a subset of Pl@ntNet collaborative images and a ViT base patch 14 dinoV2, 2024. URL: https://doi.org/10.5281/zenodo.10848263. doi:10.5281/zenodo.10848263.
[9] J. A. Hoeting, D. Madigan, A. E. Raftery, C. T. Volinsky, Bayesian model averaging: a tutorial (with comments by m. clyde, david draper and ei george, and a rejoinder by the authors, Statistical science 14 (1999) 382–417.
[10] A. Affouard, H. Goëau, P. Bonnet, J.-C. Lombardo, A. Joly, Pl@ ntnet app in the era of deep learning, in: ICLR: International Conference on Learning Representations, 2017.
[11] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al., Dinov2: Learning robust visual features without supervision, arXiv preprint arXiv:2304.07193 (2023).
[12] S. Gidaris, P. Singh, N. Komodakis, Unsupervised representation learning by predicting image rotations, arXiv preprint arXiv:1803.07728 (2018).
[13] I. Misra, L. v. d. Maaten, Self-supervised learning of pretext-invariant representations, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 6707–6717.
[14] R. Zhang, P. Isola, A. A. Efros, Colorful image colorization, in: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14, Springer, 2016, pp. 649–666.
[15] R. C. Sharma, K. Hara, Self-supervised learning of satellite-derived vegetation indices for clustering and visualization of vegetation types, Journal of Imaging 7 (2021) 30.
[16] A. Y. H. Chai, S. H. Lee, F. S. Tay, P. Bonnet, A. Joly, Beyond supervision: Harnessing self-supervised learning in unseen plant disease recognition, Neurocomputing 610 (2024) 128608.
[17] A. A. Mamun, M. Zhang, D. Ahmedt-Aristizabal, Z. Hayder, M. Awrangjeb, Conmamba: Contrastive vision mamba for plant disease detection, arXiv preprint arXiv:2506.03213 (2025).
[18] Y. Wang, Y. Yin, Y. Li, T. Qu, Z. Guo, M. Peng, S. Jia, Q. Wang, W. Zhang, F. Li, Classification of plant leaf disease recognition based on self-supervised learning, Agronomy 14 (2024) 500.
[19] K. Wang, Self-supervised learning for the distinction between computer-graphics images and natural images, Applied Sciences 13 (2023) 1887.
[20] C. E. Shannon, A mathematical theory of communication, The Bell system technical journal 27 (1948) 379–423.
[21] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[22] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, et al., Searching for mobilenetv3, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 1314–1324.
[23] S. Mehta, M. Rastegari, Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer, arXiv preprint arXiv:2110.02178 (2021).

A. Appendix: Ablation study results

No.	Run	Logit	K-value	Z-score	Private Score	Public Score
1	**exp_RAW_k5000_z1.6.csv**	RAW	5000	1.6	**0.35128**	0.32586
2	exp_SOFTMAX_k5000_z1.6.csv	SOFTMAX	5000	1.6	0.35126	0.32586
3	exp_SOFTMAX_k100_z1.6.csv	SOFTMAX	100	1.6	0.35103	0.32174
4	exp_RAW_k1000_z1.5.csv	RAW	1000	1.5	0.35080	0.32365
5	exp_SOFTMAX_k1000_z1.6.csv	SOFTMAX	1000	1.6	0.35076	0.32578
6	exp_SOFTMAX_k500_z1.6.csv	SOFTMAX	500	1.6	0.35055	0.32578
7	exp_RAW_k1000_z1.6.csv	RAW	1000	1.6	0.35051	0.32231
8	exp_RAW_k500_z1.6.csv	RAW	500	1.6	0.34935	0.32057
9	exp_RAW_k500_z1.5.csv	RAW	500	1.5	0.34925	0.31851
10	exp_SOFTMAX_k5000_z1.7.csv	SOFTMAX	5000	1.7	0.34898	0.32177
11	exp_RAW_k5000_z1.7.csv	RAW	5000	1.7	0.34896	0.32131
12	exp_RAW_k1000_z1.7.csv	RAW	1000	1.7	0.34895	0.32102
13	exp_SOFTMAX_k500_z1.7.csv	SOFTMAX	500	1.7	0.34893	0.32128
14	exp_SOFTMAX_k1000_z1.7.csv	SOFTMAX	1000	1.7	0.34893	0.32128
15	exp_SOFTMAX_k100_z1.7.csv	SOFTMAX	100	1.7	0.34870	0.32071
16	exp_SOFTMAX_k5000_z1.9.csv	SOFTMAX	5000	1.9	0.34864	0.31942
17	exp_SOFTMAX_k1000_z1.9.csv	SOFTMAX	1000	1.9	0.34849	0.31897
18	exp_RAW_k500_z1.7.csv	RAW	500	1.7	0.34829	0.32170
19	exp_RAW_k100_z1.5.csv	RAW	100	1.5	0.34821	0.32402
20	exp_SOFTMAX_k500_z1.9.csv	SOFTMAX	500	1.9	0.34814	0.31824
21	exp_RAW_k5000_z1.9.csv	RAW	5000	1.9	0.34801	0.31853
22	exp_SOFTMAX_k5000_z1.5.csv	SOFTMAX	5000	1.5	0.34784	0.32819
23	exp_RAW_k5000_z1.5.csv	RAW	5000	1.5	0.34783	0.32819
24	exp_SOFTMAX_k1000_z1.5.csv	SOFTMAX	1000	1.5	0.34782	0.32817
25	exp_SOFTMAX_k5000_z1.4.csv	SOFTMAX	5000	1.4	0.3478	0.33158
26	exp_SOFTMAX_k100_z1.9.csv	SOFTMAX	100	1.9	0.34768	0.31734
27	exp_SOFTMAX_k500_z1.5.csv	SOFTMAX	500	1.5	0.34745	0.32817
28	exp_RAW_k500_z2.0.csv	RAW	500	2	0.34744	0.31821
29	exp_RAW_k1000_z2.0.csv	RAW	1000	2	0.34717	0.31868
30	exp_SOFTMAX_k100_z1.5.csv	SOFTMAX	100	1.5	0.34712	0.32814
31	exp_SOFTMAX_k100_z1.4.csv	SOFTMAX	100	1.4	0.34705	0.33152
32	exp_SOFTMAX_k500_z1.4.csv	SOFTMAX	500	1.4	0.34702	0.33156
33	exp_SOFTMAX_k1000_z1.4.csv	SOFTMAX	1000	1.4	0.34702	0.33158
34	exp_SOFTMAX_k5000_z1.8.csv	SOFTMAX	5000	1.8	0.34696	0.32188
35	exp_RAW_k5000_z1.4.csv	RAW	5000	1.4	0.34694	0.33124
36	exp_RAW_k5000_z1.8.csv	RAW	5000	1.8	0.34690	0.32170
37	exp_SOFTMAX_k1000_z1.8.csv	SOFTMAX	1000	1.8	0.34683	0.32170
38	exp_SOFTMAX_k500_z1.8.csv	SOFTMAX	500	1.8	0.34682	0.32170
39	exp_SOFTMAX_k100_z1.8.csv	SOFTMAX	100	1.8	0.34682	0.32181
40	exp_SOFTMAX_k5000_z2.0.csv	SOFTMAX	5000	2	0.34676	0.32122

No.	Run	Logit	K-value	Z-score	Private Score	Public Score
41	exp_SOFTMAX_k1000_z2.0.csv	SOFTMAX	1000	2	0.34674	0.32135
42	exp_RAW_k5000_z2.0.csv	RAW	5000	2	0.34674	0.32114
43	exp_SOFTMAX_k100_z2.0.csv	SOFTMAX	100	2	0.34669	0.32114
44	exp_SOFTMAX_k500_z2.0.csv	SOFTMAX	500	2	0.34668	0.32135
45	exp_RAW_k1000_z1.9.csv	RAW	1000	1.9	0.34666	0.32172
46	exp_RAW_k1000_z1.8.csv	RAW	1000	1.8	0.34636	0.32110
47	exp_RAW_k500_z1.8.csv	RAW	500	1.8	0.34628	0.32039
48	exp_RAW_k1000_z1.4.csv	RAW	1000	1.4	0.34598	0.33154
49	exp_RAW_k500_z1.4.csv	RAW	500	1.4	0.34525	0.33124
50	exp_RAW_k100_z1.6.csv	RAW	100	1.6	0.34515	0.31821
51	exp_RAW_k100_z1.7.csv	RAW	100	1.7	0.34419	0.31937
52	exp_RAW_k100_z1.8.csv	RAW	100	1.8	0.34414	0.31820
53	exp_RAW_k5000_z1.3.csv	RAW	5000	1.3	0.34388	0.33619
54	exp_RAW_k500_z1.9.csv	RAW	500	1.9	0.34385	0.32117
55	exp_SOFTMAX_k5000_z1.3.csv	SOFTMAX	5000	1.3	0.34371	0.33609
56	exp_RAW_k100_z1.4.csv	RAW	100	1.4	0.34366	0.32979
57	exp_SOFTMAX_k1000_z1.3.csv	SOFTMAX	1000	1.3	0.34335	0.33609
58	exp_SOFTMAX_k500_z1.3.csv	SOFTMAX	500	1.3	0.34334	0.33494
59	exp_SOFTMAX_k100_z1.3.csv	SOFTMAX	100	1.3	0.34326	0.33502
60	exp_SOFTMAX_k5000_z1.0.csv	SOFTMAX	5000	1	0.3423	0.33991
61	exp_RAW_k1000_z1.3.csv	RAW	1000	1.3	0.34221	0.33658
62	exp_RAW_k5000_z1.0.csv	RAW	5000	1	0.3422	0.33978
63	exp_RAW_k100_z2.0.csv	RAW	100	2	0.34208	0.31438
64	exp_SOFTMAX_k1000_z1.0.csv	SOFTMAX	1000	1	0.34206	0.33978
65	exp_RAW_k500_z1.3.csv	RAW	500	1.3	0.34162	0.34101
66	exp_SOFTMAX_k500_z1.0.csv	SOFTMAX	500	1	0.34147	0.33971
67	exp_RAW_k5000_z1.2.csv	RAW	5000	1.2	0.34117	0.34054
68	exp_SOFTMAX_k500_z1.2.csv	SOFTMAX	500	1.2	0.34112	0.34032
69	exp_SOFTMAX_k1000_z1.2.csv	SOFTMAX	1000	1.2	0.34103	0.34032
70	exp_SOFTMAX_k5000_z1.2.csv	SOFTMAX	5000	1.2	0.34096	0.34055
71	exp_RAW_k100_z1.3.csv	RAW	100	1.3	0.3407	0.33985
72	exp_RAW_k500_z1.2.csv	RAW	500	1.2	0.33967	0.34052
73	exp_RAW_k100_z1.9.csv	RAW	100	1.9	0.33959	0.31902
74	exp_SOFTMAX_k100_z1.2.csv	SOFTMAX	100	1.2	0.33942	0.34027
75	exp_RAW_k1000_z1.1.csv	RAW	1000	1.1	0.33938	0.34056
76	exp_SOFTMAX_k100_z1.0.csv	SOFTMAX	100	1	0.3389	0.33984
77	exp_RAW_k100_z1.2.csv	RAW	100	1.2	0.33884	0.33602
78	exp_RAW_k5000_z1.1.csv	RAW	5000	1.1	0.33869	0.34137
79	exp_SOFTMAX_k5000_z1.1.csv	SOFTMAX	5000	1.1	0.33848	0.34155
80	exp_RAW_k500_z1.1.csv	RAW	500	1.1	0.33822	0.33892
81	exp_SOFTMAX_k1000_z1.1.csv	SOFTMAX	1000	1.1	0.33802	0.34154
82	exp_RAW_k1000_z1.0.csv	RAW	1000	1	0.33773	0.34189
83	exp_SOFTMAX_k500_z1.1.csv	SOFTMAX	500	1.1	0.33757	0.34154
84	exp_SOFTMAX_k100_z1.1.csv	SOFTMAX	100	1.1	0.33679	0.34081
85	exp_RAW_k1000_z1.2.csv	RAW	1000	1.2	0.33672	0.34277
86	exp_RAW_k500_z1.0.csv	RAW	500	1	0.33578	0.34008
87	exp_RAW_k100_z1.1.csv	RAW	100	1.1	0.33476	0.33775
88	exp_RAW_k100_z1.0.csv	RAW	100	1	0.33096	0.34046