Patch-wise Inference using Pre-trained Vision Transformers: NEUON Submission to PlantCLEF 2024

Notebook for the LifeCLEF Lab at CLEF 2024

Sophia Chulif[1,2,*], Hamza Ahmed Ishrat[1], Yang Loong Chang[2] and Sue Han Lee[1]

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

Abstract

This paper describes our submissions to the PlantCLEF 2024 challenge, where the task involved a multi-label plant classification on vegetation plot images. Vegetation plots describe the plant species composition within a fixed area, typically ranging from a few square centimetres to hectares. It is vital for ecological and environmental research and remains a challenging effort where over 300,000 plant species exist globally. The training data for the challenge includes over 1 million images of 7,806 South Western European species, while the test data comprises 1,695 high-resolution plot images in different floristic contexts. The difficulty of the challenge is the data distribution shift between the training images of individual plants and test images of multi-species vegetation plots. Noting the size of the test images, we opted for a patch-wise inference method that tried to reduce the problem from multi to single or few-class identification. We implemented patch-wise inference on the test set using an Inception-ResNet-v2-based convolutional neural network (CNN) and DINOv2-based vision transformers (ViT) with multilayer perceptrons (MLP) of 3 layers. In particular, we trained our vision transformer-based MLP classifier on custom-generated patched training images from the provided training dataset. However, our MLP classifier, the method above, did not outperform the base pre-trained DINOv2 ViT. The performance of these approaches ranked highest from the base ViT, ViT-based MLP, and finally, the CNN. Our best run, an ensemble of the base ViT and ViT-based MLP, achieved a 23.01 public macro F1 score and a 21.31 private macro F1 score per vegetation plot, making our submission the second best.

Keywords

vegetation classification, multi-species identification, vision transformer, multilayer perceptron, convolutional neural network

1. Introduction

Vegetation plots are plant observations at a particular site within a distinct area that records the plant species composition. The plots can be defined in varying sizes from a few centimetres to a few hundred metres or even hectares square. Generally, the plot sizes depend on the vegetation type, i.e., smaller plots are used for small, low-grown herbaceous vegetation, and larger plots ranging over a hundred square meters are used for woodlands [1]. Since the late 19th century, researchers have been studying the ideal plot size for species sampling [2], and various standards have been introduced [3, 4, 5]. Consequently, the documentation of species diversity enables habitat management [6], ecological research [7], conservation planning [8], and vegetation change monitoring [9]. With the advancement of technology and digitisation, the number of electronic databases of vegetation plots is increasing, dominantly in Europe. According to the Global Index of Vegetation-Plot Databases (GIVD) [10, 11], which stores the metadata of vegetation plots, over 300 databases and 3 million vegetation plots are recorded worldwide. The challenge lies in analysing and interpreting these data. Nevertheless, continuous description of vegetation composition is essential for environmental research.

Over the years, various methods, from manual annotations requiring human experts to machine learning, have been adopted for vegetation classification. Customarily, extensive field surveys are conducted involving experts and classification systems [12]. Subsequently, statistical modelling systems

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France
*Corresponding author.
schulif@swinburne.edu.my (S. Chulif); hamza.ishrat@yahoo.com (H. A. Ishrat); yangloong@neuon.ai (Y. L. Chang); shlee@swinburne.edu.my (S. H. Lee)

involving environmental data like elevation and terrain roughness are employed [13]. With the help of remote sensing technologies like aerial imagery and Light Detection and Ranging (LiDAR) systems, vegetation characteristics can not only be identified [14] but also quantified [15]. More recently, the integration of deep learning, such as convolutional neural networks (CNN) and vision transformers (ViT) are used in various vegetation tasks like weed and crop classification [16, 17] through remote-sensing imagery at high spatial resolution. Deep learning has enabled automatic vegetation feature abstraction, like leaf texture and fruit colours, that formerly required manual description. As a result, end-to-end training is achieved whereby the training stages are reduced, and a classification result can be directly obtained.

Overall, vegetation classification requires careful examination and often involves a lengthy process. The PlantCLEF 2024 challenge [18, 19] aims to tackle the multi-species image classification task from vegetation plots with the help of AI. The test set comprises Pyrenean and Mediterranean species, while the training data is focused on over seven thousand South Western European species. The main difficulty of the challenge is the shift in data distribution between the training data (single individual plant images) and test data (multi-species vegetation plot images). This paper describes our approach and results based on our submissions to PlantCLEF 2024. In the previous PlantCLEF 2022 and 2023 challenges involving large-scale plant identification of 80,000 species, we have witnessed the superiority of ViTs against CNNs. This year, two pre-trained ViTs based on the self-supervising DINOv2 architecture [20] were provided by the organisers [21]. These pre-trained models were finetuned on the PlantCLEF 2024 dataset and can be used directly by the participants. Therefore, we employed the given pre-trained ViT for this task.

Our approach mainly comprises a multilayer perceptron (MLP) based on the pre-trained ViT feature embeddings mentioned above. It was trained on custom-generated patched training images from the provided training dataset to improve its robustness in handling multi-class predictions. The custom training image would consist of a maximum of 4 species. We also experimented with an Inception-ResNet-v2 CNN (not trained on custom-generated patched training images) and the base pre-trained ViT as minimum benchmarks. In addition, we adopted patch-wise inference on the test set with the models. In this context, we split the test images into non-overlapping patches instead of feeding the photos as a whole to the networks. The patch sizes experimented on were 1 (whole image), 4, 16, and 64. Our best submission, which is an ensemble of aggregated predictions from the pre-trained ViT and MLP with inference patch sizes of 64 + 16, obtained an average public macro F1 score per plot of 23.01 and an average private macro F1 score per plot of 21.31, making it the second-best submitted run of the challenge.

2. Datasets

2.1. Training set

The training set comprises 1,408,033 training images focused on 7,806 individual plant species from Southwestern Europe. In contrast with the previous editions of the PlantCLEF challenges, the training set provided in PlantCLEF 2024 is relatively higher in resolution to cater for the use of models with larger resolution input and, at the same time, to reduce the difficulty of classifying small-sized plants in the vegetation plots. The participants are able to train with two types of training sets. The first type contains images with a minimum of 800 pixels on its biggest side (width or height), denoted as TrainMinSide, while the second type includes images with a maximum of 800 pixels on its biggest side, denoted as TrainMaxSide. Both datasets contain the same 1,408,033 training images and differ only in their image resolution. It is also pre-organised into three subsets: train, validation, and test, however, participants can choose to use their own train/validation/test dataset splits. Likewise, in the past editions of PlantCLEF, the training dataset is imbalanced; some species have hundreds of image samples, while some have only one image sample. Figure 1 shows several training images from the dataset. It illustrates the challenges in identifying small-sized plant species, particularly those sharing the same genus and similar visual characteristics. In our experiments, we used the TrainMaxSide dataset

Figure 1: Observations of four individual species sharing the same genus, *Carex*: *Carex capillaris* L., *Carex demissa* Hornem., *Carex depressa* Link, and *Carex extensa* Gooden from the PlantCLEF 2024 dataset. The photos show visual similarities between the species.

Table 1
Details of our datasets used in our CNN model training.

Dataset	Subset folder	No. of images	No. of family	No. of genus	No. of species
Train	train + test	1,356,839	181	1,446	7,806
Validation	val	51,194	181	1,415	5,912

Table 2
Details of our datasets used in our ViT model training.

Dataset	Subset folder	No. of images	No. of species
Train	train	1,356,839	7,806
Validation	test	47,940	6,670

and partitioned our training/validation splits for our CNN and ViT-based MLP models as detailed in Tables 1 and 2.

2.2. Test set

The test set includes 1,695 high-resolution (over 2000 pixels in width and height) vegetation plots comprising Mediterranean and Pyrenean species. Experts produced the plots, and the shooting protocol varied depending on the floristic contexts. Some plots are delimited by wooden frames, while others with measuring tape. It also varies in angles of view, more or less perpendicular to the ground, and quality resulting from the different environmental conditions, which produced shadows and blurry areas. The main difficulty of the challenge is the domain shift between the training and test datasets. Unlike the training images, which consist of individual plant observations, the test images comprise multiple species within an image. Figure 2 shows some samples of the vegetation plot images from the test set. Other than the difficulty in identifying multiple species within the plot, the challenge also lies in the different variations of the images, as shown in Figure 3. The shadows and blurry areas of some photos reduce the species' visibility. In addition, the dried plants, due to weather or growth stages, result in further domain shifts. From healthy/green in the training set to withering/brown in the test set.

Figure 2: Samples of the vegetation plot images from the PlantCLEF 2024 test dataset. The photos show multiple species in each plot.

Figure 3: Examples of challenges in the PlantCLEF 2024 test dataset. The shadows, blurry areas, and shift in domain (healthy/green to withering/brown) of the plant images.

3. Methodology

Based on the results from PlantCLEF 2022 [22] and PlantCLEF 2023 [23], we have seen the superiority of ViTs over standard CNNs in large-scale plant classification concerning 80,000 species. In this year's challenge, the organisers provided two pre-trained vision transformer models based on the self-supervised learning DINOv2 method [20]. Both models were finetuned on the PlantCLEF 2024 dataset with an image input of patch size 14. The first pre-trained model was frozen in its backbone layers,

Table 3
The differences between the trained Inception-ResNet-v2 CNN and DINOv2 ViT (ViT-L/14) models.

	Inception-ResNet-v2 CNN	DINOv2 ViT
Learning method	Supervised	Self-superivised
Number of parameters	56 M	80 M

Table 4
Details of our CNN model hyperparameters.

Hyperparameter	Values
Image input size	$299 \times 299 \times 3$
Batch size	128
Initial learning rate	0.0001
Weight decay	0.00004
Learning dropout rate	0.2
Optimiser	Adam
Loss function	Softmax cross entropy

and only the classifier layer was finetuned. In contrast, the second pre-trained model was finetuned on all the model layers. We have experimented with the standard convolutional neural networks and the latter vision transformer for this task. We employed an Inception-ResNet-v2-based CNN model and a DINOv2-based ViT model, which differs as described in Table 3. Essentially, the self-supervised ViT model does not require manually labelled data like the CNN model and comprises 80 million parameters, making it more resource-intensive.

3.1. Convolutional Neural Network (CNN)

Our CNN model was trained with the TF-Slim [24] library under TensorFlow 1.12 based on the Inception-ResNet-v2 [25] architecture using the hyperparameters described in Table 4. From the weights initialised on the ImageNet dataset, it was first finetuned on the PlantCLEF 2017 [26] dataset, then on the PlantCLEF 2024 dataset. We also trained another CNN model without first finetuning on the PlantCLEF 2017 dataset; however, since it performed lower in our validation set, we employed the model that was finetuned on the PlantCLEF 2017 dataset. The model is trained as a multi-task classifier relating to three taxonomy levels: Species, Genus, and Family. In addition, we trained our CNN model with increased data augmentation (random bi-cubic resizing, hue, contrast, horizontal, and up and down flipping) and balanced batching by limiting the number of training images to 50 samples per epoch.

3.1.1. Inference methods

We employed two types of inference methods to obtain our CNN model predictions. The first method is the traditional softmax function in which the class probabilities are computed from the final output layer of the model. The second method is the feature embedding similarity comparison method, whereby the model's layer before the final fully connected layer is used as a feature extractor to represent the embeddings of the trained species. Each species embedding comprises a vector size of 1536 features. An embedding dictionary of the training dataset is then constructed to compare the distances with the test set embeddings using cosine similarity to obtain the closest species prediction. Considering that our CNN model is trained on single-label images and that the high-resolution test images are provided, we split each test image into 4, 16, and 64 patches for our model's single-class predictions. The patches are divided equally in size and do not overlap one another. During inference, we also applied image cropping on each patch. Two types of cropping methods were adopted: firstly, whole cropping (which involves the entire patch), and secondly, ten croppings (which includes the centre, top-left, top-right,

bottom-left, and bottom-right crops of the patch, and all of these five croppings horizontally flipped, resulting in ten cropppings).

3.1.2. Final predictions

We applied several computations beforehand to obtain the final predictions from our CNN model. The order of these computations varies between the submission runs (which are further detailed in Section 4.2). However, in general, the processes are described in this section. Firstly, **the computation of a composite score** to assess each prediction. We used the predicted label's occurrence frequency and probability score to rank the predictions better. By accumulating the total predicted class labels and probabilities of the test images (from each image patch), we assigned specific weights to the class occurrences and probabilities. In our case, either 0.3 or 0.7 as the weights of the class occurrences and probability score. The final composite score of each test image plot was also filtered by applying a threshold of 2.5, which was determined through the histogram graph of the entire test plot images (only submission Run 12 was applied with this threshold). In addition, we **removed duplicate species sharing the same genus** from our list of predictions, and only the species with the highest probability score was kept. In submission Runs 6 and 7, we also **incorporated the model's genus label predictions** before obtaining the species predictions.

3.2. Vision transformer (ViT)

Likewise for the vision transformer models, we performed patch-wise inference on the test set then aggregate the results of each patch to get the predicted image labels. Patch-wise classification using CNNs has been proven useful in whole slide tissue images for differentiating cancer subtypes [27]. However, due to resource limitations, we utilised the vision transformer provided by the organisers [21] as our feature extractor, training only a multilayer perceptron (MLP) to form the classification head. Specifically, the "vit_base_patch14_reg4_dinov2_lvd142m_pc24_onlyclassifier_then_all" classifier that involved a 2 steps training. First, its backbone weights were frozen and only the classification head was trained. Then, the whole model was further finetuned from the only-head-trained model.

3.2.1. Data preparation

Using the given training dataset, we used the "vit_base_patch14_reg4_dinov2_lvd142m_pc24_onlyc lassifier_then_all" model as a feature extractor by removing the last classifer linear layer to extract embeddings of size 768 from the training set to create an appropriate dataset for the MLP . Each class was converted, with at most 100 embeddings per class being extracted to reduce the disparity in the number of samples per class, as it was noted during data exploration that samples per class ranged from less than 10 to more than 400.

We also employed a method similar to RICAP [28] to form our image patches to simulate the test images and train our MLP on multi-class images to improve its robustness in handling multiple classes and single-class predictions. Although where RICAP has its grid having only one intersection point (i.e., the shape of a cross with its centre along different regions of the image), we opted to have a staggered grid with varying sizes (1 to 4 patches per image) to simulate the random distribution we observed in the test images provided. The tiles were determined on a selection step; if the first split would be horizontal or vertical, then selecting a small section at the midpoint of the image to ensure the image is not split in halves. The orientation of the splits alternate: if the first split was vertical, the subpatches are split horizontally, then each of them is split vertical and so on. These steps could be extending to as many as 8 subpatches, however 4 was chosen as the maximum as it yielded the best looking patched images. Blending [29] was applied to the edges of the patches to smoothen the transition from patch to patch as a method of data augmentation that has proven effective in histopathological images [30]. Examples of our generated training images for our MLP classifier are shown in Figure 4.

Figure 4: Examples of our generated training images for our MLP classifier. Training images comprising varying patches of 1 to 4 (as shown from left to right) were created. Note that these generated multi-species training images are only used for our vision transformers models with an MLP classifier head.

3.2.2. Model setup

The model developed was a classification head consisting of 3 layers of sizes 768, 1024, and 7806. The large final layer was due to the labels being one-hot encodings of the image to allow for multi-class classification and to show what other classes were being activated when an image was presented. Batch normalisation was also used after the 1024 layer, along with the ReLU activation function. The final layer had either no activation function or a sigmoid activation function as part of experimentation, along with the type of training data, either being embeddings of single-class or multi-class images generated from the training set. The tunable hyperparameters were the learning rate, weight decay, and the number of epochs the model ran for. Cross entropy was used as our loss function and SGD for the optimiser. We also tried experimenting with binary cross entropy and sigmoid cross entropy as the loss function; however, only the cross entropy model was submitted due to the unfavourable results obtained from the other two (we had limited time to rectify the issue so they were scratched). The dataset per epoch ensured that every class was present once, giving a total dataset size of 7806 (relating to the 7806 classes available in the training set) per epoch, with a batch size of 128.

3.2.3. Bayesian Model Averaging

Bayesian Model Averaging (BMA) is a technique used in ensemble learning to best aggregate predictions produced by multiple models weighted by the model's individual performance on any given data [31]. Despite our method not using ensemble learning, we adapted its averaging method to aggregate our scores as an alternative to other methods, such as pooling or voting. BMA is traditionally calculated using two major components: likelihood and the priori of the model, M_k, given the data, D.

The likelihood is of a model M_k explains the 'correctness' of the observed data D given model M_k. Typically, this is computed via the integrals of the weights of model M_k using the Markov Chain Monte Carlo approximation method to approximate the likelihood.

The priori encapsulates the pre-existing knowledge and assumptions of M_k before observing any data. This can be initialised either uniformly (all priors across all models are the same to let the data dictate the posterior probability) or through other statistical means based on the data.

The posterior probability of M_k given D is calculated as such:

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

Where $P(D \mid M_k)$ is the likelihood of M_k given the data D, and $P(M_k)$ is the prior probability of M_k, and the denominator is the sum of all model's product of their respective likelihoods and priors.

3.2.4. Inference method

Using the patch-wise classification approach, we split each test image down to 4, 16 and 64 patches each by mathematically dividing the length and width by 2, 4 and 8, respectively, with no overlap between patches. Below are the steps with worked examples of how we arrived at our results.

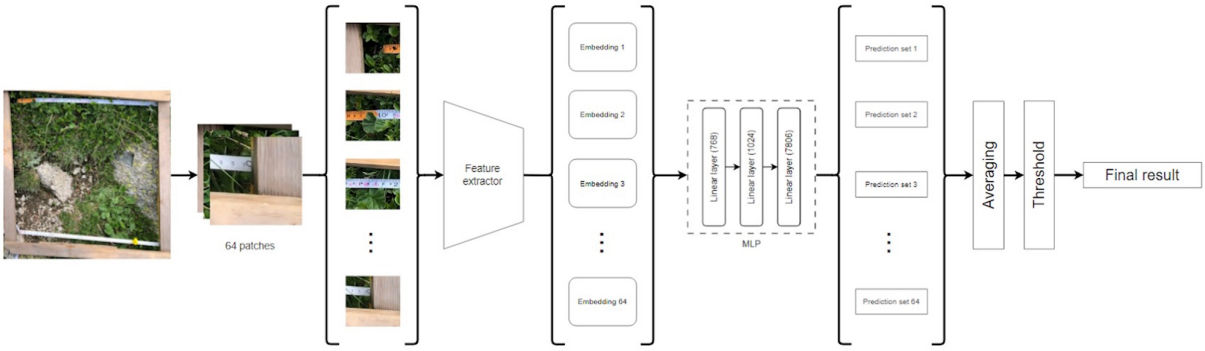

Figure 5: The overall inference method for an image.

Firstly, we split the image down to 64 individual, non-overlapping patches. For example, if the original size of the image was 3024 by 3024 pixels. With each one now being 378 by 378 pixels. The embedding is extracted for each patch and put through the trained MLP model to get an array of predictions, one for each class. Figure 5 illustrates the overall inference method for an image.

BMA Likelihood Given our case of only one model giving 64 predictions from 64 patches of one image, we adapted this equation to our needs. The easiest to figure out was the priori. Since only one model is used, we based the priori on the number of patches, N. Thus, our priori for each 'model' would then be $\frac{1}{N}$. This was later removed during the calculation of each posterior probability as it would cancel out.

The likelihood was decided upon the variance of the model's predictions. An alternative to this was the performance of the model used on a validation set. However, this would cause all the likelihoods to be the same across all predictions, and so was scraped. We chose variance as a measure of how 'confident' the model was in its prediction.

Variance calculates the spread of given set based on its mean and standard deviation, where s^2 is the sample variance, N is the number of samples and $\bar{x}$ is the mean of all the samples. The higher the variance, the bigger the spread of values. If a patch has a particularly high variance, we will assume that the top scores are significantly higher than the rest of the results and so the model is more 'confident' that it sees something in the patch. The variance is computed with the formula described in Equation 1.

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

We had initially planned to train a secondary model to discriminate against patches of plants and non-plants, as we would want to give the patches with plants within them more of a say towards the average. After training a simple MobileNet classifier on these two sets, with data sourced from the patches and sorted manually, the classifier would be too extreme in its predictions; it was 'too correct', through no fault of its own, but rather demonstrating its ability to be an excellent classifier.

To calculate the variance of each prediction, we first take the top 1000 probabilities and applied the SoftMax activation function to each prediction set to limit the range from 0 – 1, then multiplied by 100 to shift the range to 0 to 100. The variance alone, however, was not enough; it was simply too small of a value in too broad of a range, usually in the range of 10^{-1} to 10^{-6}. To overcome this, we take the absolute log of the variance, giving us a better comparison of variances:

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

From this new range of values, we needed a way to convert them into a scale of 0 to 1. We considered two functions, where a is a predetermined constant to shift the x intercept:

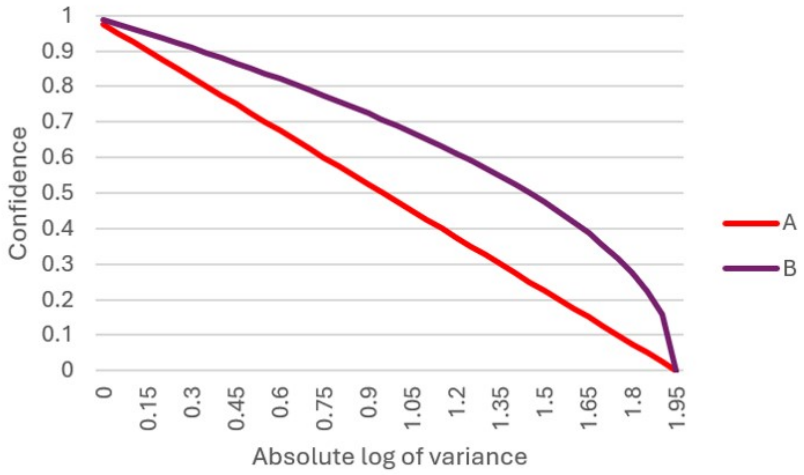

Figure 6: Graphs of functions A and B.

$$(A) \quad y = 1 - \frac{x}{a - 0.5} \qquad (B) \quad y = \sqrt{\frac{a + 0.5 - x}{a + 0.5}}$$

The graphs of these functions are illustrated in Figure 6. Ultimately, we decided on function (B) due to its bias towards higher confidence scores. Our confidence score for patch i would now be calculated as:

$$confidence_i = \sqrt{\frac{var_{max} + 0.5 - var_i}{var_{max} + 0.5}}$$

Where var_{max} is the highest recorded log of variance from the pool of 64 patches. 0.5 is added as a buffer to prevent the smallest log of variance to equal 0.

The confidence scores therefore became a means of adding weight to the prediction based on how varied the prediction set was, from the assumption that a higher variance means a more certain prediction.

Admittedly not a perfect method as evidenced by the darker, hence less confident, patches on what should obviously be identifiable as illustrated in Figure 7. We attribute this to misclassification due to high probabilities across a range of classes. This method does discriminate against the edges and some non-plant structures as intended, however some patches, despite the clear absence of plants, show up clear, i.e., higher confidence.

Lastly, assuming likelihood is represented by $confidence_i$ and priori as $\frac{1}{N}$, the posterior probability of patch i would be:

$$P(i|D) = \frac{confidence_i \times \frac{1}{N}}{\sum_{k=1}^{N} confidence_k \times \frac{1}{N}}$$

Where N is the number of patches. This can be simplified to:

$$P(i|D) = \frac{\frac{confidence_i}{N}}{\sum_{k=1}^{N} \frac{confidence_k}{N}}$$

$$P(i|D) = \frac{\frac{confidence_i}{N}}{\frac{\sum_{k=1}^{N} confidence_k}{N}}$$

$$P(i|D) = \frac{confidence_i}{\sum_{k=1}^{N} confidence_k}$$

Figure 7: Overlay of confidence heatmap on original test image. The image is split into 64 patches for inference. Darker patches indicate lower confidence, clearer patches indicate higher confidence.

Once obtained, a patch's posterior probability is multiplied to each one of its predictions, followed by the total class-wise summation across all classes to obtain the average prediction. Note that only the top 100 classes for each patch are considered for the average prediction.

Threshold As the final averages still contain a number of classes, we opted to use z-scores to determine our threshold in our predictions. Z-score as represented in Equation 2 is a measure of how many standard deviations a point is from the mean of a set.

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

Where x is the probability in question, μ is the mean and σ is the standard deviation of the set. We implemented this calculation to find the outliers in the prediction, hence considered as a more probable prediction in the set. Our threshold is therefore any class that has a z-score of 2 or more, i.e., 2 standard deviations or more from the mean. Figure 9 shows the z-scores of several test images.

4. Submissions

4.1. Evaluation metric

The evaluation metric used for the PlantCLEF 2024 challenge is the macro averaged F1 score per sample or plot. It is the average of the F1 scores calculated individually for each vegetation plot. The F1 score considers both precision and recall, in which precision measures the fraction of true positive labels among the instances predicted as positive by the model. In contrast, recall measures the fraction of true positive instances the model correctly predicted. The F1 score can be represented as $F1 = 2 \times \frac{precision \times recall}{precision + recall}$.

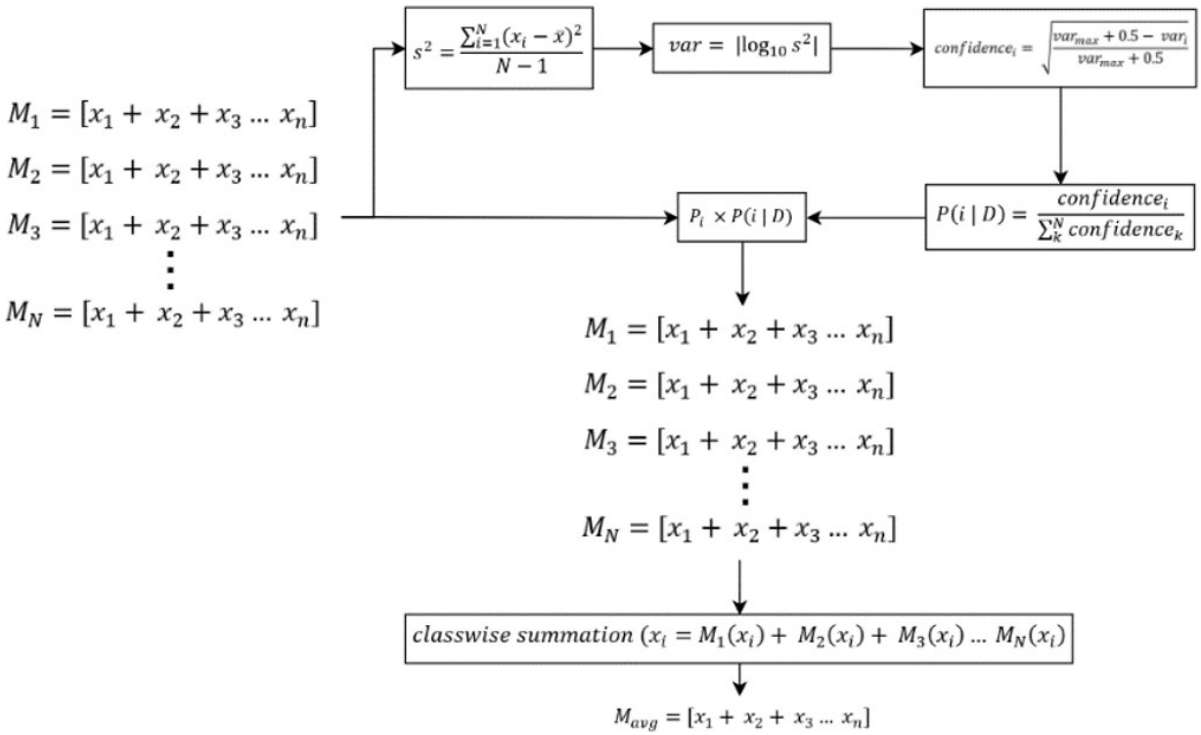

Figure 8: Mathematical workflow of the averaging method.

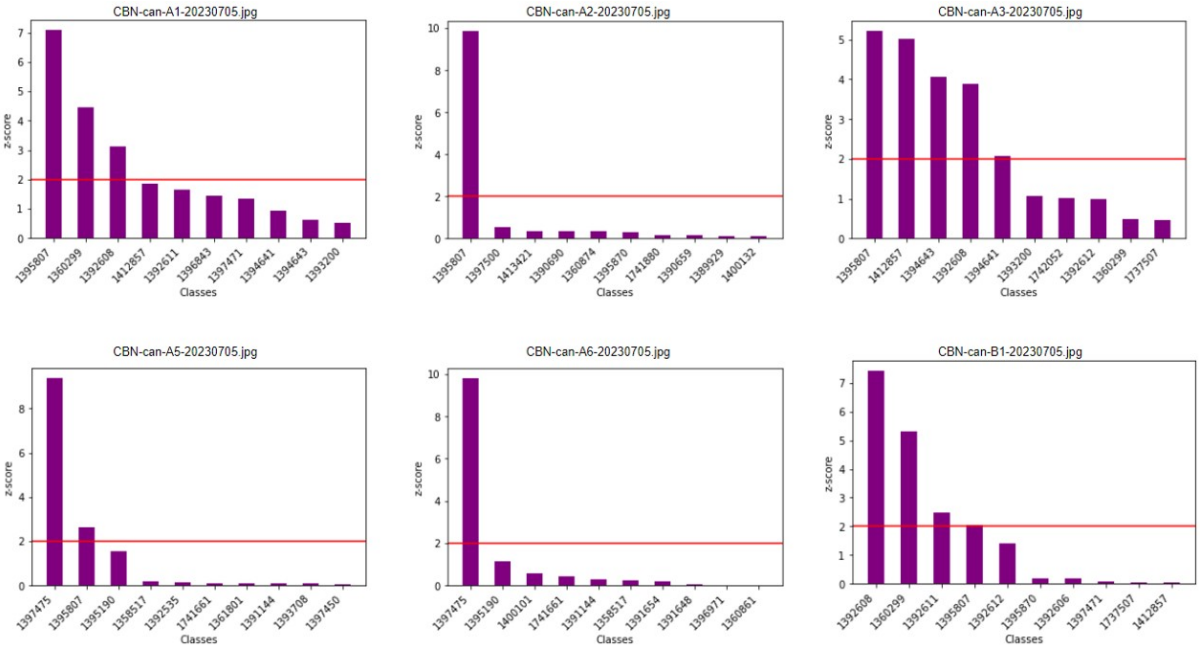

Figure 9: Graphs showing z-scores of the first 6 testing images. The red lines indicate a threshold of z-score = 2. Any class exceeding the limit is considered in the final result.

4.2. Models and runs

We submitted thirteen runs to PlantCLEF 2024; 5 consisted of the CNN models, while 8 consisted of the ViT models. The 5 CNN runs involved the same Inception-ResNet-v2 model and only differ in inference methods. Meanwhile, among the 8 ViT models, 4 included a trained MLP classifier head to get the initial predictions, 3 used the pre-trained vision transformer provided by PlantCLEF, and the last was an aggregation of the ViT and MLP models. We experimented with several patch sizes: 1 (the

whole image), 4, 16, 64, and 64 + 16 patches, where the predictions across the 64 sub-patches and 16 sub-patches were combined for the calculations. The overall submitted models are detailed in Table 5 and summarised as follow. The number (in parentheses) in the run names indicates the patch size on which it was inferred. For example, in Run 1, the “ViT (1) model" refers to the patch-wise inference on a single patch, which is the entire image.

4.2.1. Run 1: ViT (1) model

This run directly used the pre-trained ViT model provided by PlantCLEF. Inference on the test image was made on the whole image without splitting the test image to patches, with the z-score of more than 2 being used as the threshold.

4.2.2. Run 2: MLP v1 (1) model

This is similar to the Run 1, except the model used was the ViT and MLP v1 model with the threshold being the same.

4.2.3. Run 3: CNN (16) model

This run used the trained CNN model. Inference on the test image was made from 16 patches. We applied ten croppings on each patch, obtained the top-10 species predictions of each image crop, aggregated the results for the composite score calculations, and finally removed the species sharing the same genus.

4.2.4. Run 4: CNN (4) model

This run is the same as Run 3, the only difference was instead of 16 patches, 4 patches was used during inference.

4.2.5. Run 5: MLP v1 (64) model

This run used the trained MLP classifier along with the ViT as the feature extractor. Inference was made on 64 sub patches and the predictions were aggregated using the averaging method discussed previously.

4.2.6. Run 6: CNN genus (16) model

This run used the trained CNN model. Inference on the test image was made from 16 patches. We applied ten croppings on each patch, obtained the top-10 species and top-1 genus predictions of each image crop, and aggregated the results for the composite score for both the species and genus predictions. From the total species prediction list, we removed the species not under the genus prediction list. Then, we removed the species sharing the same genus. Finally, we picked the top-10 species from the final prediction list.

4.2.7. Run 7: CNN genus (64) model

This run used the trained CNN model. Inference on the test image was made from 64 patches. We applied ten croppings on each patch, obtained the top-5 species and top-1 genus predictions of each image crop, and aggregated the results for the composite score for both the species and genus predictions. From the total species prediction list, we removed the species not under the genus prediction list. Then, we removed the species sharing the same genus. Additionally, we applied a threshold of 2.5 on the species composite score before selecting the top-10 species from the final prediction list.

Table 5
Details of our submitted models.

Run	Model	Architecture	Learning rate	Epochs trained	Patch size used
1	ViT (1)	Pre-trained ViT	-	-	1
2	MLP v1 (1)	Pre-trained ViT + 3-layer MLP	0.05	100	1
3	CNN (16)	Inception-ResNet-v2	0.0001	91	16
4	CNN (4)	Inception-ResNet-v2	0.0001	91	4
5	MLP v1 (64)	Pre-trained ViT + 3-layer MLP	0.05	100	64
6	CNN genus (16)	Inception-ResNet-v2	0.0001	91	16
7	CNN genus (64)	Inception-ResNet-v2	0.0001	91	64
8	ViT ten crops (64)	Pre-Trained ViT	-	-	64
9	ViT (64)	Pre-trained ViT	-	-	64
10	MLP v2 (64 + 16)	Pre-trained ViT + 3-layer MLP	0.07	300	64 + 16
11	MLP v2 (64)	Pre-trained ViT + 3-layer MLP	0.07	300	64
12	CNN emb (64 + 16)	Inception-ResNet-v2	0.0001	91	64+16
13	Ensemble (64 + 16)	-	-	-	64 + 16

4.2.8. Run 8: ViT ten crops (64) model

This run is similar to Run 1 except that inference on the test image was made on 64 patches. In addition, the cropping on each patch was applied with ten croppings but no z-score threshold was applied. We selected the top-10 species as the final species prediction list.

4.2.9. Run 9: ViT (64) model

This run is similar to Run 5, except the base ViT is used for the inference on 64 patches.

4.2.10. Run 10: MLP v2 (64 + 16) model

This run is similar to Run 5, except the MLP v2 is used as the classifier for the inference on 64 and 16 patches.

4.2.11. Run 11: MLP v2 (64) model

This run is similar to Run 10, except the MLP v2 is used as the classifier for the inference on only 64 patches

4.2.12. Run 12: CNN emb (64 + 16) model

Unlike the previous CNN runs, this run applied the feature embedding similarity comparison method by comparing the embedding dictionary of the training dataset with the patches of test images. Inference of the test image was made from the ensembled embeddings of 64 + 16 patches. Additionally, instead of ten croppings, whole cropping of each patch was used. From each patch, we obtained the top-5 species prediction using cosine similarity then compute the species composite score. From the list of species predictions, we removed the species sharing the same genus and applied a threshold of 2.5 on the species composite score before picking the top-10 species from the prediction list.

4.2.13. Run 13: Ensemble (64 + 16) method

This run combined the predictions of Runs 5, 9 and 11, where each uses 64 + 16 patches to get the predictions. Once aggregated using the averaging method, the top-n classes from each prediction for each image were combined, removing duplicates.

4.3. Results and discussion

The evaluation metric used by the organisers was the macro averaged F1 score per sample, or more specifically, MacroF1ScoreAveragedPerPlot; however, MacroF1ScoreAveragedPerSpecies and MicroF1Score were also shown for information purposes on the challenge's HuggingFace platform: https://huggingface.co/spaces/BVRA/PlantCLEF2024. The results were divided into public and private scores, and the final rankings were based on the private splits. Since we do not have the complete results of the private split scores, we will only display our public score results in this section. The results from our submitted runs are detailed in Table 6. Overall, our ensemble predictions of the pre-trained ViT and the MLP classifier placed 2nd on the public leaderboards, with the base ViT run as third.

We noticed a boost in performance when using the same model varying the number of patches from 64 to 64 and 16 as in the case of both runs involving MLP v2, with the MacroF1ScoreAveragedPerPlot going from 19.57 to 20.16. However, we did not expect it to perform worse than the MLP v1 model, considering it was trained for longer with a slightly higher learning rate. The MLP v1 model came as a surprise as we had purposely cut its training time short as the deadline was approaching and needed a benchmark, which ultimately turned out to be one of the top methods. The ViT model was also used to provide a benchmark on models using the same averaging method and outperformed both MLP models with a MacroF1ScoreAveragedPerPlot of 22.19. As a final attempt we decided to gather the predictions of all ViT-based models based on 64 + 16 patches and combined them together, giving us our best run of a MacroF1ScoreAveragedPerPlot of 23.01.

Further research In summary, ViT remains superior and outperforms CNN in this multi-species identification task, as expected. Additionally, comparing the sole use of the base pre-trained ViT (64) model (Run 9 - 22.19 F1 score) and the MLP v1 (64) model (Run 5 - 21.65 F1 score), we find that our MLP model did not improve in performance. It could be due to the different data distribution between the test set (vegetation plots) and the generated patched training images, as illustrated in Figure 4. A more realistic generation of patched images could be experimented with to tackle this issue. Data augmentation techniques like ReMix [32] and CutMix [33] can also be implemented and studied.

Additionally, using a higher patch size for inference has helped in the classification of all models. With high-resolution plots, splitting the image into patches is favourable to reduce the problem from a multi to single or few-class identification. The largest patch size we tried was 64. Therefore, more patches can be examined to determine a suitable patch size for the vegetation classification.

Furthermore, the adoption of higher taxonomy level classification (genus) implemented in the CNN run, CNN genus (16) (Run 6 - 10.91 F1 score) showed improvements over the results from the CNN without genus classification, CNN (16) (Run 3 - 9.7 F1 score). Due to constraints, genus classification was not implemented in our ViT models, but it can be employed for further research.

Besides, binary cross entropy or sigmoid cross entropy loss functions can be adopted to create a better or more accurate multi-species classification model.

Lastly, more studies can be carried out to analyse the misclassifications of the models, specifically the ViT-based, as shown in Figure 7. More experiments can be explored to determine and eliminate why the absence of plants produced high confidence scores.

Table 6
Results of our submitted runs based on the public leaderboard.

Run/Model	MacroF1ScoreAveraged PerPlot	MacroF1ScoreAveraged PerSpecies	MicroF1Score
(13) Ensemble (64 + 16)	23.01	46.2	20.84
(9) ViT (64)	22.19	51.57	19.75
(5) MLP v1 (64)	21.65	46.41	18.92
(10) MLP v2 (64 + 16)	20.16	42.89	17.94
(11) MLP v2 (64)	19.57	42.04	17.38
(8) ViT ten crops (64)	16.99	39.16	17.3
(7) CNN genus (64)	12.81	41.32	12.63
(6) CNN genus (16)	10.91	36.02	10.85
(3) CNN (16)	9.7	32.95	9.52
(1) ViT (1)	9.14	38.68	7.63
(12) CNN emb (64 + 16)	6.9	32.18	6.51
(2) MLP v1 (1)	5.16	23.99	4.51
(4) CNN (4)	3.56	34.85	3.57

5. Conclusion

This paper presents our multi-species vegetation plot classification work in conjunction with PlantCLEF 2024. Considering the task of the challenge involving high-resolution test plot images, we implemented patch-wise inference on the image whereby the plot is split into varying sizes, reducing the task from a multi to single or few-class identification. Our best submission, an ensemble of DINO-v2 based ViT and MLP classifiers, scored an average public macro F1 score per plot of 23.01 and a private macro F1 score per plot of 21.31, ranking us second in the challenge. Since our ViT model comprising the MLP classifier head did not perform better than the base pre-trained ViT benchmark, further research can be implemented to improve the classification. ViT has once again shown to be a competitive choice in plant identification in the context of PlantCLEF. Nonetheless, more research is required to tackle the resource constraints and to leverage this powerful architecture in vegetation tasks like plot classification.

Acknowledgments

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

References

[1] M. Chytrỳ, Z. Otỳpková, Plot sizes used for phytosociological sampling of european vegetation, Journal of vegetation science 14 (2003) 563–570.
[2] J. Moravec, The determination of the minimal area of phytocenoses, Folia geobotanica & phytotaxonomica 8 (1973) 23–47.
[3] M. Poore, The use of phytosociological methods in ecological investigations: I. the braun-blanquet system, Journal of ecology 43 (1955) 226–244.
[4] J. S. Rodwell, J. N. C. C. (GB), National vegetation classification: Users' handbook, Joint nature conservation committee Peterborough, 2006.
[5] S. K. Wiser, N. Spencer, M. De Caceres, M. Kleikamp, B. Boyle, R. K. Peet, Veg-x–an exchange standard for plot-based vegetation data, Journal of Vegetation Science 22 (2011) 598–609.
[6] C. Y. Manullang, et al., Distribution of plastic debris pollution and it is implications on mangrove vegetation, Marine Pollution Bulletin 160 (2020) 111642.
[7] W. Hansen, J. Wollny, A. Otte, R. L. Eckstein, K. Ludewig, Invasive legume affects species and

functional composition of mountain meadow plant communities, Biological Invasions 23 (2021) 281–296.
[8] Y. R. F. Nunes, C. S. Souza, I. F. P. de Azevedo, O. S. de Oliveira, L. A. Frazão, R. S. Fonseca, R. M. dos Santos, W. V. Neves, Vegetation structure and edaphic factors in veredas reflect different conservation status in these threatened areas, Forest Ecosystems 9 (2022) 100036.
[9] A. H. Halbritter, V. Vandvik, S. H. Cotner, W. Farfan-Rios, B. S. Maitner, S. T. Michaletz, I. Oliveras Menor, R. J. Telford, A. Ccahuana, R. Cruz, et al., Plant trait and vegetation data along a 1314 m elevation gradient with fire history in puna grasslands, perú, Scientific Data 11 (2024) 225.
[10] J. Dengler, F. Jansen, F. Glöckler, R. K. Peet, M. De Cáceres, M. Chytrỳ, J. Ewald, J. Oldeland, G. Lopez-Gonzalez, M. Finckh, et al., The global index of vegetation-plot databases (givd): a new resource for vegetation science, Journal of Vegetation Science 22 (2011) 582–597.
[11] J. Dengler, et al., Global index of vegetation-plot databases (givd), 2011. https://www.givd.info/ [Accessed: 2024-06-06].
[12] I. Noble, The role of expert systems in vegetation science, Vegetatio 69 (1987) 115–121.
[13] A. E. Gelfand, M. Holder, A. Latimer, P. O. Lewis, A. G. Rebelo, J. A. S. Jr., S. Wu, Explaining species distribution patterns through hierarchical modeling, Bayesian Analysis 1 (2006) 41 – 92. URL: https://doi.org/10.1214/06-BA102. doi:10.1214/06-BA102.
[14] C. X. Garzon-Lopez, E. Lasso, Species classification in a tropical alpine ecosystem using uav-borne rgb and hyperspectral imagery, Drones 4 (2020) 69.
[15] L. Huo, E. Lindberg, J. Holmgren, Towards low vegetation identification: A new method for tree crown segmentation from lidar data based on a symmetrical structure detection algorithm (ssd), Remote Sensing of Environment 270 (2022) 112857.
[16] T. Kattenborn, J. Leitloff, F. Schiefer, S. Hinz, Review on convolutional neural networks (cnn) in vegetation remote sensing, ISPRS journal of photogrammetry and remote sensing 173 (2021) 24–49.
[17] R. Reedha, E. Dericquebourg, R. Canals, A. Hafiane, Transformer neural network for weed and crop classification of high resolution uav images, Remote Sensing 14 (2022) 592.
[18] A. Joly, L. Picek, S. Kahl, H. Goëau, V. Espitalier, C. Botella, B. Deneu, D. Marcos, J. Estopinan, C. Leblanc, T. Larcher, M.
vSulc, M. Hrúz, M. Servajean, J. Matas, 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.
[19] H. Goëau, V. Espitalier, P. Bonnet, A. Joly, Overview of PlantCLEF 2024: Multi-species plant identification in vegetation plot images, in: Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, 2024.
[20] T. Darcet, M. Oquab, J. Mairal, P. Bojanowski, Vision transformers need registers, arXiv preprint arXiv:2309.16588 (2023).
[21] 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.
[22] H. Goëau, P. Bonnet, A. Joly, Overview of plantclef 2022: Image-based plant identification at global scale., in: CLEF (Working Notes), 2022, pp. 1916–1928.
[23] H. Goëau, P. Bonnet, A. Joly, Overview of plantclef 2023: image-based plant identification at global scale, in: 24th Working Notes of the Conference and Labs of the Evaluation Forum, CLEF-WN 2023, volume 3497, 2023, pp. 1972–1981.
[24] Sergio Guadarrama, Nathan Silberman, TensorFlow-Slim: A lightweight library for defining, training and evaluating complex models in tensorflow, https://github.com/google-research/tf-slim, 2016. URL: https://github.com/google-research/tf-slim, [Online; accessed 29-June-2019].
[25] C. Szegedy, S. Ioffe, V. Vanhoucke, A. A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[26] A. Joly, H. Goëau, H. Glotin, C. Spampinato, P. Bonnet, W.-P. Vellinga, J.-C. Lombardo, R. Planqué, S. Palazzo, H. Müller, Lifeclef 2017 lab overview: multimedia species identification challenges, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction: 8th International Conference of the CLEF Association, CLEF 2017, Dublin, Ireland, September 11–14, 2017, Proceedings 8, Springer, 2017, pp. 255–274.

[27] L. Hou, D. Samaras, T. M. Kurc, Y. Gao, J. E. Davis, J. H. Saltz, Patch-based convolutional neural network for whole slide tissue image classification, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2424–2433.

[28] R. Takahashi, T. Matsubara, K. Uehara, Ricap: Random image cropping and patching data augmentation for deep cnns, in: Asian conference on machine learning, PMLR, 2018, pp. 786–798.

[29] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, J. M. Ogden, Pyramid methods in image processing, RCA engineer 29 (1984) 33–41.

[30] S. T. M. Ataky, J. De Matos, A. d. S. Britto, L. E. Oliveira, A. L. Koerich, Data augmentation for histopathological images based on gaussian-laplacian pyramid blending, in: 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, 2020, pp. 1–8.

[31] 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.

[32] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, Y. Yoo, Cutmix: Regularization strategy to train strong classifiers with localizable features, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 6023–6032.

[33] J. Cao, L. Hou, M.-H. Yang, R. He, Z. Sun, Remix: Towards image-to-image translation with limited data, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 15018–15027.