Preprocessing Is All You Need: TheHeartOfNoise Submission to PlantCLEF 2025

Notebook for the LifeCLEF Lab at CLEF 2025

Vincent Espitalier

Abstract

The PlantCLEF 2025 competition promotes scientific research in biodiversity monitoring. Participants are invited to analyze high-resolution images, known as quadrats, to identify the plant species present. This task poses several challenges. For example, plants may occupy only a small proportion of the total image area, or they may have a morphology that makes them difficult to identify due to seasonal variations. Every detail counts, and the entire processing chain must be optimized, from loading the high-definition image to compiling the list of predicted species. Preprocessing, which transforms the source image into a tensor for the machine learning model, is the focus of this article's study and leads to the best classification performance in the PlantCLEF 2025 competition. As part of this participation, three approaches to analyzing the quadrat images were explored. The first method, used as a reference, simply reduces the images to the expected model resolution, allowing the research to focus on optimizing preprocessing. The research results were then used to tile the images before analyzing them with a single-shot high-resolution inference. The Rust computer language was primarily used. The associated source code is available on a public repository.[1].

Keywords

multilabel classification, high-resolution image, image preprocessing, plant identification, vision transformer, computer vision, tiling method, high-resolution inference, Rust

1. Introduction

Environmental issues and biodiversity conservation are becoming increasingly important, particularly for monitoring the decline of native species and the emergence of invasive ones [1]. In botany, the quadrat method [2] is used to assess biodiversity indices. This method involves counting plant species in standardized areas, typically 50 cm by 50 cm, at regular intervals to estimate changes in plant diversity. The PlantCLEF 2025 competition [3], part of the LifeCLEF lab [4] under the Conference and Labs of the Evaluation Forum (CLEF) follows on from PlantCLEF 2024 [5] and aims to facilitate this monitoring by simplifying the work of botanists through artificial intelligence.

The Pl@ntNet team, which has been active for fifteen years in research related to the visual identification of plant species, has developed a free application [6] using computer vision models to quickly identify plant species. The team is also working to improve the efficiency of quadrat analysis [7], which is a complex task due to the massive amount of data that must be processed. Often, several hundred VisionTransformer inferences are required for a single high definition image. This makes it essential to optimize computing power and the data processing pipeline.

New automation capabilities for image analysis have been made possible by artificial intelligence, which is gradually being integrated into various aspects of everyday life, such as medicine and autonomous vehicles. In 2012, the ImageNet competition, which involved classifying images into 1,000 categories, was won by the AlexNet model [8], a convolutional neural network with 60 million parameters, trained using the backpropagation algorithm proposed by Yann LeCun in 1989 [9]. This breakthrough was made possible by increased computing power and the availability of large amounts of data.

[1]Public repository: https://github.com/v-espitalier/PlantCLEF2025

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

v.espitalier@laposte.net (V. Espitalier)
https://github.com/v-espitalier/ (V. Espitalier)
0009-0008-8795-6043 (V. Espitalier)

In 2017, convolutional neural network technology was surpassed by Transformers [10], initially designed for natural language processing and adapted in 2020 for image analysis as VisionTransformers [11]. These models, often containing hundreds of millions or even billions of parameters and require large amounts of annotated data to avoid overfitting. This has popularized self-supervised learning methods ([12, 13, 14]). These methods enable the construction of representation latent spaces with strong generalization capabilities and pre-trained models that can be easily adapted to various use cases. Examples include VisionTransformer DINOv2 [15] and its improved version with registers, DINOv2Reg4 [16], which were used in this competition.

Certain use cases of artificial intelligence require specific reliability and robustness criteria, regarding the quality of predictions and proper software functioning. Research is underway to develop explainable AI to increase confidence in the technology [17]. In the short term, risks of cyber and IT failures must also be considered, especially for critical applications such as autonomous cars and space technologies (e.g., satellites). In this context, the Rust programming language [18] stands out by offering performance comparable to hardware-close languages, such as C/C++ [19], while ensuring greater safety, particularly in memory management [20, 21].

Participation in PlantCLEF 2025 aims to advance scientific research in image analysis and demonstrate the potential of the Rust language, through the Candle library [22], for computer vision research. Although the Python/PyTorch stack [23, 24] is widely adopted, Rust offers unique benefits that enhance both reliability and efficiency in computer vision tasks. These advantages include improved performance, enhanced safety, concurrency support, and seamless integration with other languages such as C and CUDA.

Implementing image analysis methods in Rust required reproducing the official code of the reference method (Python script "Official Starter Notebook | Inference on Test Data") and processing the image directly at the expected 518x518 resolution. This process highlighted the essential role of preprocessing, which was then investigated further. Next, two approaches to high-resolution image processing were explored, each leading to different technical challenges: the tiling method, used in the previous edition of the competition [25, 26], consists of extracting square samples from the original image at different scales, potentially with overlap. Then, these tens or hundreds of tiles are inferred individually before the predictions are aggregated.

A second approach, VaMIS (Variable Model Input Size) [7], is based on high-resolution inference. This approach allows the model to adapt to analyzing large images with a single inference, reducing the computational footprint compared to tiling. All tokens in the image can interact with each other via global attention. To further reduce computational complexity, the image can be segmented into windows, allowing tokens to interact solely within their respective windows [27]. Finally, a hybrid attention method alternates between the previous two techniques to combine their advantages [28], as does the SegmentAnything image encoder [29].

The goal of this competition is to improve the performance of plant classification, i.e., to identify the species visible within each quadrat based primarily on visual data rather than image metadata. Therefore, these metadata have been utilized to a limited extent.

2. Overview

2.1. The PlantCLEF2025 competition

The annual PlantCLEF competition aims to promote scientific research in plant classification. Since last year [5], the objective has been to encourage quadrat research, or the multi-label classification of high-definition images. To this end, various single-label and multi-label datasets are provided for training and evaluation purposes. Two deep learning models trained for plant classification are made available to participants to avoid the need for significant computing resources.

2.2. Research avenues explored

A significant part of the research was devoted to data preprocessing. Various avenues of research have made it possible to optimize technical choices, such as the interpolation used during resizing or the choice of a subsampling scheme for certain color channels and the JPEG compression quality, performed after resizing, which can surprisingly be seen as a regularization factor.

During the five-week participation in the PlantCLEF 2025 competition, two approaches were explored for analyzing high-definition images with a VisionTransformer implemented in Rust: the classic tiling method and high-resolution inference methods known as VaMIS, for VAriable Model Input Size. VaMIS uses either global or window attention, or both, within the same model, referred to as hybrid attention. Since VaMIS-adapted models are highly memory-intensive in terms of GPU, it was necessary to reimplement an attention module calculation in CUDA, which is more VRAM-efficient.

A brief collaboration with the company Quandela should also be mentioned. The goal was to conduct preliminary tests to evaluate and enhance the performance of multilabel classification using quantum photonics and demonstrate the potential of this technology. The Python libraries Perceval, for designing quantum circuits, and MerLin [30], for easily integrating these circuits into a PyTorch model, enable such experiments. This research took place over the course of three days in the final week of the competition. As the preliminary results were inconclusive, no submission was made during the competition. However, these methods could be further developed with more reasonable research deadlines. More details can be found in the appendix.

3. Methodology

3.1. The 2025 PlantCLEF challenge

The PlantCLEF 2025 competition promotes innovation in ecology, focusing on biodiversity monitoring and plant species evolutionary dynamics. Based on the standardized quadrat protocol sampling method, the challenge motivates participants to develop automated approaches for analyzing high-resolution images of plant quadrats. The main objective is to identify various plant species among over 7,800.

3.1.1. Single-label dataset

The PlantCLEF 2025 monolabel training dataset remains the same as in the previous edition. It consists of observations of individual plants in southwestern Europe and covers 7,806 species. The dataset includes approximately 1.4 million images, which have been supplemented with additional images from the GBIF platform to include less represented species. The images are pre-organized into subfolders by species and divided into training, validation, and test sets to facilitate model training.

3.1.2. Multi-label datasets

The PlantCLEF 2025 test dataset is a compilation of quadrat image datasets in various floristic contexts. It contains a total of 2,105 high-resolution images. The shooting protocols vary, including different angles and weather conditions. These images, produced by experts, allow to evaluate the ability of models to accurately identify plant species under various conditions, thereby testing their robustness.

A complementary dataset is also available [31]. It contains over 200,000 images from the LUCAS Cover Photos 2006-2018 collection, including a large number of unannotated pseudo-quadrat images. These additional data are intended to adapt the models to process images of multi-species vegetation quadrats. These data were not used in this study.

3.1.3. Technical characteristics

The images are provided in JPEG format, which can introduce compression artifacts that modify the represented data by reducing its entropy, according to the principle of lossy compression. Two main

Table 1
JPEG compression quality of training images. These data were obtained by analyzing the 1,408,033 images in the training dataset with a maximum image size of 800px.

JPEG compression quality	Number of images
75 percent	7
76 percent	1102476
95 percent	305550
Total	1408033

Table 2
YCbCr JPEG subsampling schemes for training images. These data were obtained by analyzing the 1,408,033 images in the training dataset with a maximum image size of 800px. The 4:4:4 ratio corresponds to the absence of chroma subsampling. The 4:2:2 ratio signifies horizontal chrominance subsampling by a factor of 2.

YCbCr JPEG subsampling scheme	Number of images
4:4:4	25
4:2:2	1408008
Total	1408033

Table 3
JPEG compression quality of test images. These data were obtained by analyzing the 2,105 images in the quadrats test dataset.

JPEG compression quality	Number of images
98 percent	1397
80 percent	664
78-95 percent	44
Total	2105

Table 4
YCbCr JPEG subsampling schemes for test images. These data were obtained by analyzing the 2,105 images in the quadrats test dataset. The 4:2:2 ratio signifies horizontal chrominance subsampling by a factor of 2. The 4:2:2v ratio signifies vertical chrominance subsampling by a factor of 2. The 4:2:0 ratio signifies horizontal and vertical chrominance subsampling by a factor of 2 respectively.

YCbCr JPEG subsampling scheme	Number of images
4:2:2	1403
4:2:0	662
4:2:2v	40
Total	2105

parameters control compression: JPEG compression quality and the YCbCr subsampling scheme. The latter is the underlying color space used to decompose the luminance of the chroma channels. These compression parameters are discussed in a dedicated section below.

Tables 1 and 2 summarize the JPEG technical characteristics of the training dataset, which contains images with a maximum size of 800px. Similarly, tables 3 and 4 report the JPEG compression parameters of the quadrat test set.

3.1.4. Models

To facilitate access to the competition, two VisionTransformer [11] models were made available to participants. These models were pre-trained using the DINOv2 self-supervised learning (SSL) method [15]. The final DINOv2Reg4 architecture uses registers [16] as temporary memory to aggregate information at the image level. The chosen architecture size is "ViT-Base" which consists of 12 successive VisionTransformer blocks with 12 attention heads and a latent space size of 768. The models take an RGB image as input. The image is partitioned into 14x14 pixel patches, leading to 37x37=1369 local tokens. Including the CLS token and the four registers, there are 1369 + 1 + 4 = 1374 tokens (i.e. vectors) of size 768. These tokens represent the informational state space in which the VisionTransformer operates.

The first model uses public DINOv2 weights pre-trained by SSL for the image encoder. Only the classification head was trained in a supervised manner. The second model was trained entirely in a supervised manner using the first model as initial checkpoint. Both models were trained on a server with A100 GPUs using the Timm library with Torch. The first model was trained for approximately 17 hours over 92 epochs, with a batch size of 1,280 images per GPU, and a learning rate of 0.01. The second model was trained for about 36 hours over 92 epochs, with a batch size of 144 images per GPU and a learning rate of 0.00008.

For the PlantCLEF 2025 competition, only the second model was used. Since all of its parameters were fine-tuned on the plant training dataset, it can be expected to perform better than the first model.

3.1.5. F1 metric

The metric chosen to rank participants' submissions is the macro-averaged F1 score per sample, which strikes a good balance between statistical recall and precision. The 2,105 images are grouped into transects, which represent samples from specific areas within selected sites. To mitigate biases related to oversampled areas, the score is first calculated for each transect in the test set and then averaged across transects to obtain the final score.

3.1.6. Challenge posed by the competition

Several machine learning challenges must be solved to enable effective classification. The training data consists of images of individual plants or parts of plants with a single label, while the test data consists of images of vegetation quadrats with multiple labels. Therefore, the monolabel model must be adapted to perform multi-label classification. Additionally, the test images have fairly high resolutions, frequently ranging from eight to ten million pixels, compared to the model's input size of approximately 250,000 pixels. The next section details different approaches to address this discrepancy.

3.2. Research avenues explored

Quadrat image analysis involves processing a substantial amount of information, including images containing nearly 10 million pixels and a tiling process that can multiply the number of inferences by 100 compared to the initial dataset. Additionally, the analysis predicts between one and 15 species out of 7,806 possibilities, generating a large number of potential combinations. In accordance with the official public Python script, the decision was made to limit the maximum number of predicted species to 15 in order to avoid substantially reducing statistical accuracy.

A clear pipeline for processing this data must be established. This pipeline can be described in four successive steps:

1. Preprocessing: conversion of a 3000x3000x3 quadrat image stored on hard disk in JPEG format to 518x518x3 tensor(s) in RAM (or VRAM).
   First, the high-definition image file is loaded into RAM. Then, the JPEG format is decoded. Next, the image is resized and cropped. JPEG decoding can take into account the final position of the pixels. In other words, it is not necessary to decode all 10 million pixels of the initial quadrat

image because this process is relatively intensive. One or more tiles (i.e., square areas within the original image) are extracted and resized to the model's input resolution of 518x518 pixels or a higher resolution for the high-resolution inference approach.

2. The image encoder of the DINOv2 plant model: transformation of the 518x518x3 image tensor(s) into a vector(s) of size 768.
   In a standard deep learning approach, the reduced image is provided as input to the model in the form of a tensor. The VisionTransformer image encoder calculates "deep features," which are floating-point vectors that summarize all the plant information contained in the original image. Then, it performs linear classification. It was decided that these deep features (more precisely, only the CLS token, which is used for classification) would be stored on a hard disk.
   The classification head is usually calculated immediately after the image encoder. However, it is preferable to divide the model inference into two steps. This is because calculating deep features is more resource-intensive than calculating the classification head. The latter can be recalculated quickly after loading the deep features from the hard disk without significant delay. Additionally, storing predictions by tile and species would require much more space than storing the respective deep features. Finally, testing other classification heads or fine-tuning only the head may be desirable, if necessary.
3. Linear classification and prediction aggregation: transformation of the 768-size vector(s) into a 7806-size score vector.
   Each deep feature vector (CLS token) is input into a linear classifier to obtain a vector of logits, and then the SoftMax function is applied to get probabilities by species. Predictions are aggregated from the tile level to the high-definition image level using the chosen method: maximum pooling by species, average pooling, a given quantile [32], or any method summarizing information at the quadrat image level.
   To limit confusion and improve statistical precision, it is also possible to retain only one or two species per tile, as is done in the official code provided for the tiling method.
4. Submission calculation: transformation of the 7806-size score vector into a list of species.
   Only the 15 most probable species at the high-definition image scale are retained (e.g., Top15), and only those with a score above the detection threshold are selected. The final list comprises the species chosen by the model as predictions for the given quadrat image. Other methods of calculating submissions are possible.

As part of the PlantCLEF 2025 competition, the research primarily focused on the initial stages of the pipeline: image preprocessing and selecting an image encoder process to analyze high-definition images. Two approaches were implemented for the latter: the classic tiling technique and the VaMIS approach, which analyzes each image with a single high-resolution inference. Due to insufficient computing power, these approaches were only tested in inference mode, i.e., without fine-tuning the deep learning model. Finally, prediction aggregation and submission calculation were briefly explored without specific analysis.

To implement these models and calculate inferences and submissions, Python and Rust code were produced.

3.3. Preprocessing

The development of the Rust pipeline started with implementing the reference approach, which solves the problem of analyzing high-definition images by simply reducing them to the resolution expected by the VisionTransformer model (518 x 518 pixels). The initial goal was to reproduce the results of the official script provided for the "Official Starter Notebook | Inference on Test Data" competition. However, multiple tests and the launch of the code revealed a bottleneck caused by image preprocessing. Loading and decoding JPEG images with nearly 10 million pixels and reducing them to 518x518 pixels requires as much computational power as, if not more than, the VisionTransformer inference that follows image loading.

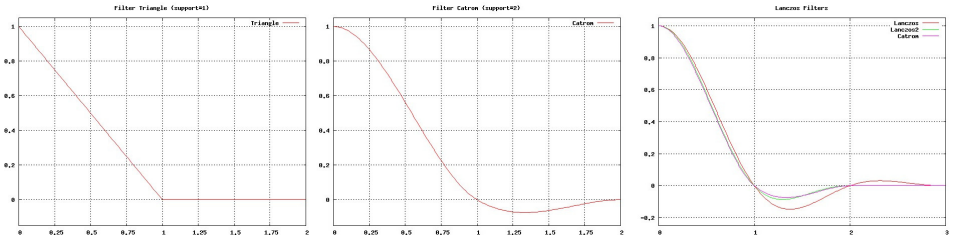

Figure 1: Graphs of the filters used. In order: Bilinear, Bicubic, and Lanczos

Therefore, the initial step involved performing external preprocessing of the images in the PlantCLEF 2025 test dataset to reduce their size upstream. This allowed to use the reduced images as input and test the Rust solution. The list of plant species predicted by quadrat varied significantly depending on the chosen preprocessing. In particular, preprocessing involving JPEG compression after resizing the high-definition image showed significant variations in performance.

Therefore, research on preprocessing was conducted along several lines: the interpolation used for resizing, and in the case of post-resizing JPEG compression, the YCbCr subsampling scheme and compression quality factor used by JPEG.

Tests were conducted using the reference approach, which reduces images to the expected dimensions of 518x518 pixels at the model input. This technical choice allows for multiple tests at reasonable computational cost. It also allows for better evaluation and comparison of the different preprocessing methods. Listing the plant species present in a quadrat image with a reduced resolution of 518x518 pixels is difficult; every pixel counts for identifying plants. This differs from a tiling process, in which tiles are extracted at the original resolution of the image by cropping it. In the latter case, the importance of technical preprocessing choices is reduced. The detection threshold was set to a one percent probability for all the experiments, which is the same value as the official 518x518 reference method.

Finally, the impact of preprocessing, including post-resizing JPEG compression, was measured on the two approaches implemented for PlantCLEF 2025: tiling and the high-resolution VaMIS approach.

3.3.1. Interpolations

Interpolation necessarily plays a role in the final performance of the classifier. Reducing the image size requires estimating the color of pixels in a grid that may not overlap the original geometry. This process makes assumptions when choosing a surface model. This introduces an additional source of noise that may affect the downstream deep learning model, thereby penalizing classification quality. The following interpolations were tested: nearest neighbor, bilinear, bicubic, and Lanczos. Many other interpolation methods exist and could have been evaluated as well. However, the decision was made to limit the evaluation to the most commonly used methods.

Nearest neighbor interpolation is unique because it simply involves selecting the pixel in the source grid that is closest to the target pixel. In contrast, bilinear, bicubic, and Lanczos interpolations are convolution products, i.e., weighted averages of the pixels in the neighborhood of the target pixel projected onto the source grid. Named after Hungarian mathematician and physicist Cornelius Lanczos, the latter interpolation is calculated using the cardinal sine function and is known to reduce visual artifacts such as blurring and aliasing. Figure 1 illustrates the filters associated with these convolution products. [33] provides additional information.

The classification performance of preprocessing was evaluated with each of the four interpolations on grayscale test data images to avoid dependence on a particular color space and to focus on the geometric calculations involved in reducing high-definition quadrat images to 518x518 pixels.

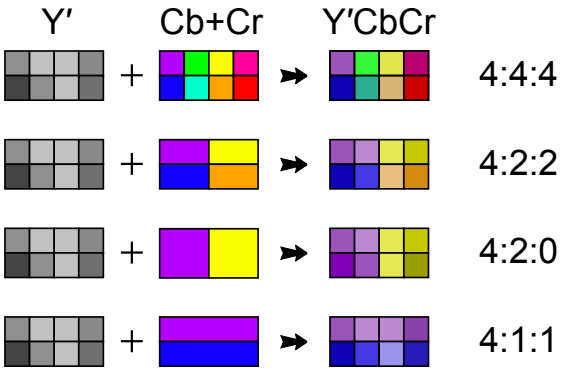

Figure 2: YCbCr subsampling scheme.

3.3.2. JPEG compression: YCbCr scheme and quality factor

YCbCr subsampling scheme Various preprocessing methods were considered, including recompressing the resized images to JPEG format. The initial aim was to reduce disk space and subsequently optimize classification performance. This required selecting the YCbCr subsampling scheme and compression quality used by JPEG.

The YCbCr color space, obtained via a linear transformation from RGB, distinguishes the Y luminance channel from the Cb and Cr chroma channels. To reduce data size while accounting for human visual perception, JPEG allows the chroma channels to be subsampled relative to the luminance channel according to standardized schemes:

- **4:4:4** : No subsampling. The chroma and luminance components have the same resolution.
- **4:2:2** : chroma is subsampled horizontally by a factor of 2.
- **4:2:0** : chroma is subsampled horizontally and vertically by a factor of 2.
- **4:1:1** : chroma is subsampled horizontally by a factor of 4.

These schemes are summarized in Figure 2.

A fifth, less common scheme appears in the test data, in which chroma is subsampled vertically by a factor of two. It is denoted **4:2:2v** in this article.

JPEG compression quality In JPEG compression, the quality factor is a number between 0 and 100 that determines how high frequencies are filtered in the discrete cosine transform (DCT, [34]) and how the remaining coefficients are quantized. For example, a higher quality factor preserves more high frequencies, provides more accurate quantization, and results in less compression of the image.

The prediction performance of the single inference reference method was assessed by reducing high-definition quadrat images to a resolution of 518x518 pixels. The tested preprocessing methods involve post-resizing JPEG compression, including a comparison of five subsampling schemes and JPEG compression qualities ranging from 75 to 100 percent.

3.4. High-resolution processing approaches

3.4.1. Tiling method

Principle This paper explored the tiling method used in previous editions [25, 26] at different scales. First, the source image is resized to a multiple of 518 by cropping the long edge. Then, it is partitioned without overlap into square samples of size 518 x 518, which corresponds to the input expected by the model. The dimension multiplier corresponds to the scale. For example, at a scale of three, nine adjacent tiles are produced without overlap, as illustrated in Figure 3.

Figure 3: Diagram illustrating the tiling method. In the case of scale 3 tiling, the pixels of the high-definition image are partitioned into 3x3=9 square surfaces referred to as tiles. Each tile is then resized to the model's input dimension (518x518 pixels) for inference. This method was explored during the previous edition of the PlantCLEF competition[25, 26].

Each tile is fed into the VisionTransformer image encoder, which returns a vector of deep features stored on the hard drive. During the competition, different scales were calculated and aggregated (from scale one to scale ten). The preprocessing implements JPEG compression after resizing the high-definition image.

Prediction calculation and aggregation The model's linear classification head is applied to the deep features of each tile to obtain the logits. The probabilities for each species are obtained by applying the SoftMax function. Probabilities at the image scale are obtained by applying maximum or average pooling per species. Initially used for convolutional neural networks [35], the logic of maximum pooling is as follows: If a species is detected in a tile, its probability in that tile is high, and the max() operator "brings up" this high score to the quadrat image scale. This corresponds well to the detection of presence in the image, regardless of the tile position or its occupied surface within the image.

3.4.2. High-resolution inference (VaMIS)

VaMIS (Variable Model Input Size) methods analyze high-resolution quadrat images using a single inference and adapt the machine learning model. This reduces computational costs compared to the tiling method, which performs multiple inferences at different scales.

In the VisionTransformer [11] inference process, the image is split into 14x14 pixel patches. Each patch is linearly projected, and its position in the image is encoded by adding a vector of learned parameters. According to Transformer [10] terminology, this becomes a "token." These tokens interact in pairs via the attention module, much like words in a sentence. The second VisionTransformer module is the feed-forward module, which processes the tokens individually. There is a distinction between local tokens, which come from image patches according to the aforementioned method, and global tokens (CLS and registers), which have an initial value learned during model training.

Global attention In the case of global attention, all tokens are considered in the calculation simultaneously. This standard calculation is the most natural method of image size extension. The model analyzes the entire square image and can refine its predictions by taking co-occurrences between plant species into account. The standard VaMIS method is available in Candle [22].

Window attention This method, known as Window Shifted Attention (WSA) [27], partitions tokens based on their initial spatial positions within the image. The attention module only interacts with tokens belonging to the same window (see Figure 4). In the 3x3 case, for example, this corresponds

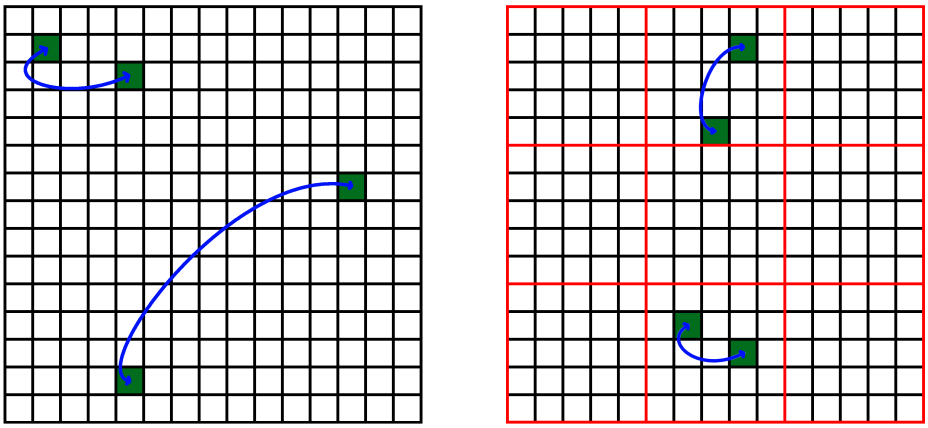

Figure 4: Diagram illustrating global attention (left) and window attention (right). The high-definition image is divided into patches, which are then projected into tokens. Global attention involves interactions between all pairs of tokens within the image by calculating their similarities. From the first Transformer block, neurons have access to the entire image, which is thus analyzed as a whole according to the Vision Transformer principle. Window attention (Window Sliding Attention) partitions the set of tokens based on the spatial position of the corresponding patch within the image (red grid). Tokens interact only within the same window. In the absence of learning species co-occurrences, it is possible that distinct areas of a quadrat image can be processed partly independently, similar to tiling, without losing classification performance, significantly reducing the amount of computation. However, window attention ensures aggregation of the global image information after each Transformer block, thanks to global tokens (CLS and 4 registers), aggregated by averaging over all windows. The diagram illustrates a simplified example with 15x15=225 tokens. Global attention requires 225x225=50,625 similarity calculations, whereas window attention, with 3x3=9 windows, requires only 9x(5x5)x(5x5)=5,625, dividing the computational complexity of the attention module by 9 in this example.

to replacing standard VaMIS attention with nine attentions on adjacent windows of the image. After the window attention calculation is complete, the global tokens (CLS and registers, respectively) are averaged over the nine windows to obtain the global tokens for the entire image. This technical choice is natural but would certainly benefit from specific fine-tuning of the adapted model.

Windows smaller than 518x518 can be chosen, provided the resized image size is a multiple of the window size. For example, a window VaMIS with a size of 1680x1680 and 5x5 windows of size 336x336 or 8x8 windows of size 210x210 pixels is possible.

Hybrid attention One limitation of window attention is that local tokens from different windows never interact with each other. Hybrid attention, which alternates between global and window attention, allows information to diffuse better across the whole image by enabling all tokens to interact within given Transformer blocks.

The Segment Anything image encoder [29] uses this approach, spacing global attention evenly. For instance, in a VisionTransformer with a "ViT-Base" dimension [11], such as the one used in the competition with 12 blocks, global attention is applied to blocks 3, 6, 9, and 12. The other blocks use window attention, as detailed in Table 5.

The three VaMIS methods described above were tested at a single resolution of 1554x1554 (equivalent to scale three of the tiling). Due to a lack of computing resources, the deep learning models were not fine-tuned for these methods and were only used for inference. The preprocessing implements JPEG compression after resizing the high-definition image.

Implementation of attention calculation in CUDA The attention module calculates the similarity between all pairs of tokens in the image, resulting in an attention matrix. The attention matrix has n^2 elements, where n is the number of tokens. VisionTransformers divide an input image into patches of a fixed size (e.g., 14 x 14 pixels for DINOv2) and then convert the patches into tokens. The number of

Table 5
Attentions of the 12 Transformer blocks for the VaMIS method with hybrid attention. Hybrid attention [28] offers a compromise between image encoders using either global attention or window attention. In the case of a Vision Transformer with hybrid attention, global attention is alternated with window attention, reducing computational complexity while retaining some diffusion of global information, i.e., at the image scale, within each local token, as implemented in the image encoder of SegmentAnything [29].

Index of Transformer bloc	Attention type
1	Window
2	Window
3	Global
4	Window
5	Window
6	Global
7	Window
8	Window
9	Global
10	Window
11	Window
12	Global

tokens increases linearly with the image's surface area and therefore quadratically with its dimensions (width and length) while maintaining its aspect ratio. Thus, doubling the dimensions of an image multiplies the size of the attention matrix by 16.

VaMIS methods rely on increasing the model input size, but the limits of the graphics card are quickly reached using the explicit method to calculate the attention module because the attention matrix must be stored in GPU memory. FlashAttention [36] optimizes this calculation and limits GPU memory usage. However, this feature was unavailable on the graphics card used during the competition because it was too old. Therefore, a new implementation of the attention module in the CUDA language was necessary.

3.5. Submission calculation: thresholding and species selection

The above methods provide a vector of presence scores for each species at the image level. The list of species predicted by the model is obtained by thresholding; that is, by listing the species whose score (e.g. probability) exceeds a chosen detection threshold.

For the tiling method, the threshold was slightly optimized by testing different levels and observing the public F1 score of the respective submissions. However, this method is costly in terms of the number of submissions required.

The detection thresholds for the VaMIS methods were chosen roughly, considering the average number of species detected. One run was performed for each VaMIS method to allow for initial validation of the Rust implementation.

Semi-automatic methods were tested occasionally to adjust the detection thresholds per transect because some transects are more difficult to analyze and require different thresholds. These thresholds were determined using a semi-automatic procedure with dichotomy to maintain an average number of species similar to a reference classification submission. A multiplicative parameter adjusts this number globally to optimize performance. These methods were primarily applied to tiling techniques.

Finally, manual optimizations were briefly explored through a detailed analysis of the species prediction probabilities generated by the model and subsequent submissions. For example, if a species frequently appeared within a transect, it was occasionally generalized across all images of that transect. Conversely, species appearing only once (or only a few times) within a transect were deemed outliers and subsequently excluded from the predictions. Such considerations were applied across various scales:

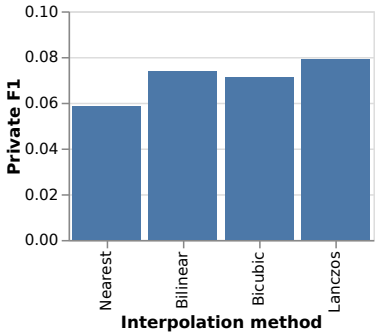

Figure 5: Private F1 for different interpolations used during resizing. The reference pipeline was used, in which the model performs a single inference at a resolution of 518x518 pixels. The 2,105 test images were converted to grayscale beforehand to avoid depending on a particular color space and to focus attention on the geometric calculations. Lanczos interpolation yields the best results.

Table 6
Main sets of JPEG compression parameters after resizing that give the best performance. The reference pipeline was used, and the model performed a single inference at a resolution of 518x518 pixels. Both subsampling schemes, 4:2:2 and 4:1:1, show a maximum value of around 0.23752 private F1.

YCbCr sub-sampling scheme	Quality factor	Private F1	Public F1
4:2:2	85	0.23752	0.20085
4:1:1	94	0.23694	0.19507

the entire dataset, individual transects, and images from the same location within a transect. Given that these methods incorporate metadata rather than relying solely on visual data, performance metrics with and without these optimizations are provided to illustrate the quality and generalizability of the approaches outlined in this technical note.

4. Results

Most of the test data (89 percent) is used to calculate the private F1 score, making it more robust than the public F1 score from a statistical perspective. To perform a relevant analysis of the results, this section mainly lists the private F1 statistics according to different methods and parameter sets.

4.1. Preprocessing

4.1.1. Interpolations

Figure 5 illustrates the classification performance of different interpolations. Nearest neighbor interpolation is the least effective, while Lanczos outperforms the others. It is notable that F1 score of 0.07906 is achieved with grayscale images (Lanczos interpolation), compared to a score of 0.12077 with the reference approach of the official script, which uses the three RGB channels.

4.1.2. JPEG compression: YCbCr sub-sampling scheme and quality factor

Figure 6 illustrates the classification performance for different YCbCr subsampling schemes. Two "modes" or two maxima, are observed for different sets of parameters, summarized in Table 6. We find that a JPEG quality of 100 percent does not improve multi-label classification performance for the considered model. Performance improves as soon as this quality is slightly reduced. In particular, all schemes perform remarkably well (private F1 score greater than 0.22) with a JPEG quality of 75 percent.

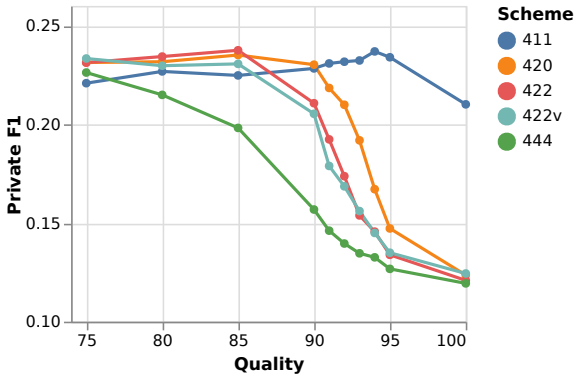

Figure 6: Private F1 for different JPEG compression parameter values: compression quality factor and YCbCr subsampling scheme. The reference pipeline was used, and the model performed a single inference at a resolution of 518x518 pixels. It is noteworthy that reducing the JPEG quality factor of post-resizing compression generally improves classification performance. All schemes provide remarkable performance (private F1 greater than 0.22) for a JPEG quality of 75 percent.

Table 7
Performance of tiling schemes. For each quadrat image, square tiles are extracted at various scales. There is no overlap within each scale. Scale three, for example, yields nine tiles, each measuring 518 x 518. The 46 tiling scheme corresponds to scales one, three, and six, as shown in the table. Only the graphic content of the images (visual data) is analyzed, not the metadata. The preprocessing involves JPEG compression following the resizing of the high-definition image. Scheme 91 achieves the best performance. It uses all scales from one to six and therefore involves 91 VisionTransformer inferences for each high-resolution quadrat image. The private F1 score is 35.013. Schemes involving higher scales do not seem to improve performance.

Scheme (tiles number, geometry)	Private F1	Public F1	Detection threshold
1=1x1	0.23268	0.19348	1%
10=3x3+1	0.30523	0.31209	10%
46=6x6+3x3+1	0.33876	0.34745	33%
91=6x6+5x5+4x4+3x3+2x2+1	0.35013	0.32818	35%
155=8x8+6x6+5x5+4x4+3x3+2x2+1	0.34426	0.33497	40%
210=10x10+8x8+6x6+3x3+1	0.31361	0.33590	52%

The 4:1:1 subsampling scheme, which reduces chroma by a factor of four relative to luminance, performs well for all JPEG qualities. In contrast, the 4:4:4 scheme, which does not subsample, requires significantly reduced JPEG compression quality (75 percent) to perform well.

4.2. High-resolution processing approaches

4.2.1. Tiling method

The tiling schemes are summarized in Table 7. The tiling scheme that provides the best performance is the one with 91 tiles and a private F1 of 0.35013. Schemes with more tiles do not seem to improve performance. Note that the above results do not use metadata and only evaluate visual detection performance.

Based on the results of the 91-tiles scheme, the detection thresholds were subsequently optimized by studying the average number of species per transect. Manual optimization involves quickly analyzing the predictions and species lists visually at different levels of image grouping, such as the entire dataset, a transect, images from the same location within a transect, or an individual quadrat image. For instance, frequent species within a group of test images were generalized and outliers were removed. The performance results are summarized in Table 8.

Table 8
Additional performance gains were achieved through optimizations using metadata. Tiling with scheme 91 was further developed, improving performance using metadata. Preprocessing includes JPEG compression after resizing the high-definition image. Purely visual performance using a global threshold of 35 percent serves as a reference. Images were grouped by transect according to name. A probability detection threshold was optimized for each transect, resulting in an approximate gain of 0.007 in private F1. Further manual optimization involving analysis of lists of predictions and submitted species for different image groupings, as well as removal of detection anomalies, allows for an additional increase in private F1 of approximately 0.0075.

Scheme (tiles number, geometry)	Private F1	Public F1	Detection threshold
91=6x6+5x5+4x4+3x3+2x2+1	0.35013	0.32818	global 35%
91=6x6+5x5+4x4+3x3+2x2+1	0.35691	0.33499	per transect
91=6x6+5x5+4x4+3x3+2x2+1	0.36479	0.3523	manual optimization

Table 9
Performance of high-resolution inference methods. The VisionTransformer model has been adapted to accept larger images as input. In this test, the resolution is 1554 x 1554, which corresponds to a magnification factor of three (3 x 518 = 1554). Global attention is the standard type of attention. It is available in Candle [22] and makes all tokens in the image interact. Two additional types of attention have been implemented in Rust: window and hybrid. Window attention groups tokens according to square areas based on the initial position of 14x14 patches within the image. A 3x3=9 window pattern was chosen. Hybrid attention alternates between global and window attention, similar to the SegmentAnything image encoder. Of the 12 blocks in the VisionTransformer, only blocks 3, 6, 9, and 12 use global attention. These tests were performed without optimizing the probability detection threshold, and only during inference (i.e., without fine-tuning the machine learning model). Only the graphic content of the images (visual data) is analyzed, not the metadata. The preprocessing implements JPEG compression after resizing the high-definition image. Hybrid attention appears to produce better results, with a private F1 score of 0.22711.

Attention	Private F1	Public F1	Detection threshold
Global	0.21950	0.20271	1.5%
Windows	0.22225	0.19684	1.5%
Hybrid	0.22711	0.19993	3%

Optimizing the threshold per transect improves the private F1 score by approximately +0.007. Manual optimization improves the private F1 score by an additional +0.0075 on top of this initial gain.

4.2.2. High-resolution inference (VaMIS)

High-resolution inference methods without fine-tuning achieve the performance shown in Table 9. Among these methods, the hybrid attention method based on the SegmentAnything image encoder attention scheme [29] seems to be the most effective.

4.3. Preprocessing: performance delta on tiling and VaMIS approaches

All of the performance results shown below were obtained using a purely visual model. The detection threshold was the same for all of the images in the test dataset, and no metadata was used.

Two types of preprocessing were compared: Standard (without JPEG compression) and with JPEG compression after resizing. The Standard approach uses the Rust Image crate, which can be compared to Python/Pillow, while the JPEG compression approach uses the MagickRust crate, which uses Wand/ImageMagick and JPEG compression. Due to preprocessing, the performance delta indicated in Table 10 can be observed for the different tiling schemes.

For the VaMIS models, the performance deltas are listed in Table 11.

Table 10
Performance deltas obtained with post-resizing JPEG compression for the tiling approach. Resizing the high-definition image to the 518x518 model resolution and then applying JPEG compression yields a performance delta of more than +0.11 on the private F1. With 91 tiles, the delta is smaller, yet still significant: +0.03354 on the private F1.

Tiling scheme	Private F1	Public F1
1 tile (518x518)	0.12077	0.11776
1 tile (518x518) with JPEG compression	0.23268	0.19348
Delta	+0.11191	+0.07572
46=6x6+3x3+1	0.31861	0.32102
46=6x6+3x3+1 with JPEG compression	0.33883	0.34745
Delta	+0.02022	+0.02643
91=6x6+5x5+4x4+3x3+2x2+1	0.31662	0.32500
91=6x6+..+1 with JPEG compression	0.35013	0.32818
Delta	+0.03354	+0.00318

Table 11
Performance deltas obtained with post-resizing JPEG compression for the high-resolution inference approach. Using JPEG compression after resizing the high-definition image to a resolution of 1554x1554 yields a performance delta on the private F1 ranging from +0.020 to +0.056, depending on the type of attention. The last two experiments use window attention. For this, the technical choice was made to average global tokens (CLS and registers, respectively).

Attention	Private F1	Public F1
Global	0.19938	0.19286
Global with JPEG compression	0.21950	0.20271
Delta	+0.02012	+0.00985
Window	0.16687	0.16274
Window with JPEG compression	0.22225	0.19684
Delta	+0.05538	+0,0341
Hybrid	0.17200	0.15751
Hybrid with JPEG compression	0.22711	0.19993
Delta	+0.05511	+0.04242

5. Discussion

Based on the results, Lanczos interpolation performed the best out of those tested. This is a common finding in signal theory [33]. It was used to conduct JPEG compression tests after resizing images.

Interestingly, a JPEG quality of 100 percent with no subsampling (4:4:4 scheme) does not yield optimal performance in multi-label classification. The tests revealed two distinct "modes" in which the model performed well: One is the 4:1:1 sampling scheme with 94 percent JPEG compression; the other is the 4:2:2 scheme, which requires 85 percent compression.

These two modes are related to the characteristics of the data used to train the model, which include a quasi-unique 4:2:2 sampling scheme and two main levels of compression for the training images. 76 percent (for more than 3 images over 4) and 95 percent. These uniform JPEG parameters determine the entropy, or variance, and more generally the distribution of the RGB color channels of the input pixels during training. It is possible that the model has learned to process only images with these characteristics.

When a test image is resized from a high resolution, such as 3000x3000 pixels, to the model's input size

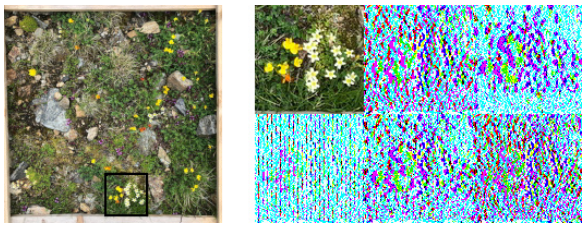

Figure 7: Visualization of the YCbCr color space and chroma channel subsampling for different preprocessing methods. The RGB images are saved right before the Vision Transformer inference process begins. A linear transformation is applied to obtain the YCbCr channels. Then, the differences between neighboring pixels are calculated and normalized between 0 and 255 to visualize the variations on each channel. The top row of the mosaic shows the following: quadrat RGB "Pyr-03", YCbCr channels from the official submission, YCbCr channels from ImageMagick. Bottom row: YCbCr channels from the 4:1:1, 4:2:2, and 4:4:4 submissions. The official submission's visualization is similar to the 4:2:2 and 4:4:4 schemes, which appear noisy but informative. The 4:1:1 visualization shows vertical lines of chroma channel realignment every four pixels. The ImageMagick visualization shows adaptive JPEG preprocessing, likely with smoothing.

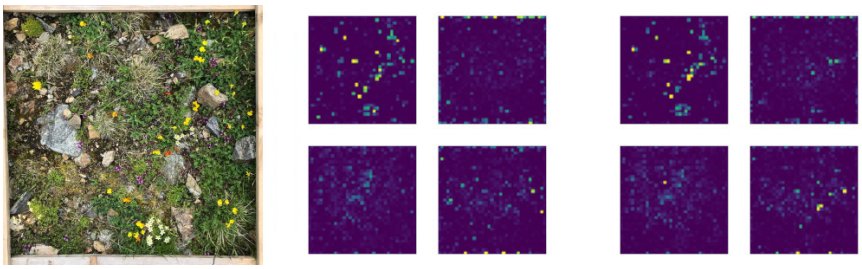

Figure 8: A comparison was conducted between four attention maps (out of twelve) from the final block of a Vision Transformer, derived from two different submissions which resize test images to 518x518 pixels: the public official script submission (left), and a submission involving post-resize JPEG compression with a 4:1:1 sampling scheme and a JPEG quality setting of 94. The additional compression appears to affect the attention maps, with the model focusing its attention more selectively. This suggests enhanced discriminative capabilities and potentially improved species detection due to reduced noise in the maps.

of 518x518 pixels—effectively dividing each side by a factor of approximately six—the resulting image exhibits a significantly higher equivalent JPEG quality compared to its original version. Furthermore, during tile extraction at a scale of six, which involves minimal resizing, two-thirds of the test images retain a compression quality of 98 percent (see Table 3). Overall, all inference methods tend to supply the model with images of higher quality than the one used during its training phase (see Table 1).

Figure 7 illustrates the impact of various compression methods on the YCbCr channels. The model was primarily trained using images with the 4:2:2 scheme (bottom row, middle image). The RGB images are almost indistinguishable to the human eye, following the principle of YCbCr subsampling. However, the model may be more sensitive to these variations than the human eye. Thus, changing the compression scheme between the training and test data constitutes an additional domain shift.

The VisionTransformer appears to treat additional data in the input image, beyond the entropy expected by the model, as noise, which reduces its ability to focus on classification details. Figure 8 compares the 12 attention maps of the final block of the model when it is provided with an uncompressed image and an image compressed with the optimal parameter set, "Scheme 4:1:1 — Quality 94." The first map shows slightly noisier attention, while the second map shows more focused attention on certain points. From the model's point of view, JPEG compression can be seen as a regularizing factor for the image, as the model expects precise entropy on the input pixels. It is noteworthy that the benefits of JPEG compression in a different deep learning context have been previously discussed [37].

From the point of view of the final classification task, this lost data during compression after resizing is certainly informative, particularly in reducing confusion between species thanks to the details of neighboring pixels contained in the high frequencies of the JPEG discrete cosine transform (DCT, [34]).

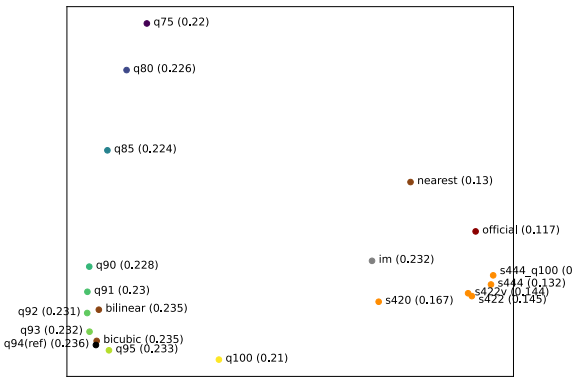

Figure 9: The first two components of the principal component analysis (PCA) of the similarity matrix of the submissions use the F1 score per pair of submissions to measure their correlation. The PlantCLEF2025 private F1 score is indicated in parentheses. The black reference corresponds to the maximum of the YCbCr 4:1:1 scheme (JPEG quality 94, YCbCr 4:1:1 scheme, and Lanczos interpolation). Only variations from this reference are labeled: JPEG quality (q75–q100), scheme (s420, s422, s422v, s444), and interpolation (bilinear, bicubic, and nearest neighbors). The official submission and the ImageMagick submission (default preprocessing) are also shown. It is evident that preprocessing based on JPEG qualities ranging from 91 to 95 or bilinear and bicubic interpolations yields similar submissions and performance. The ImageMagick submission is somewhat isolated, indicating specific preprocessing. The official submission is close to the 4:2:2, 4:2:2v, and 4:4:4 preprocessing.

Training the model on images with various compression schemes and also without any compression could make the model less constrained on the expected input entropy and improve the performance of quadrat images in multi-label classification and confirm the present analysis.

Figure 9 allows for a visual comparison of different submissions resulting from different preprocessing methods. Submissions with JPEG quality ranging from 91 to 95, as well as those with bilinear or bicubic interpolation, are similar. The ImageMagick submission is somewhat isolated, indicating specific preprocessing. The official submission is similar to preprocessing with the JPEG 4:2:2, 4:2:2v, and 4:4:4 schemes.

The tiling approach achieves the best results for high-resolution analysis. It gives a private F1 of 0.35013 with a scheme involving six scales. The high-resolution inference approach (VaMIS) achieves a private F1 of 0.22711 with the hybrid attention method, which alternates between global and window attention. These results improved with preprocessing and post-resizing JPEG compression, explaining a performance delta of +0.03354 and +0.05511, respectively.

Further research concerning the color space used during image resizing could be pursued to more exhaustively study preprocessing. Some color spaces use nonlinear transformations, such as LAB and LUV, and interpolation can produce different images and performance.

6. Conclusion

Preprocessing plays a significant role in the classification performance of computer vision models, especially when analyzing high-definition quadrat images. In these images, plants may occupy only a small area, making each pixel of the reduced image of great importance.

The model is conditioned by the training data. Classic training on the ImageNet dataset requires image normalization of the same name during the testing phase. It focuses on the first two moments of the input image tensor. More generally, the model is affected by the entropy of the input tensors. Therefore, special focus should be given to the format of the training images and their preprocessing.

For instance, having a monolabel training dataset with original, high-resolution images from the sensors before any resizing would allow to use various preprocessing techniques during training such

as interpolation method or JPEG compression parameters. This would teach the model to consider the image details in the high frequencies of the JPEG DCT, improving visual classification performance and making the model more robust to preprocessing.

Additional tests could be conducted for high-resolution image analysis. The tiling approach with scales beyond the native resolution of the quadrat images and/or with overlap should be explored further. Indeed, tiles that framed the plants more precisely facilitated identification. Similarly, single-shot high-resolution inference approaches (VaMIS) could benefit from fine-tuning the model's input size. As previously indicated for tiling, smaller windows for methods involving window attention would refine the spatial framing of each plant within an image.

Additionally, contest participants most likely focused on tiling schemes and methods for aggregating predictions from these tilings. Some may have even tested and fine-tuned other deep learning models. However, preprocessing optimization is not a research avenue usually prioritized to improve the classification performance of a computer vision model. Therefore, we can expect the solutions proposed by other participants to be combined with the preprocessing optimization proposed here to achieve cumulative gains in classification performance.

Finally, since preprocessing is relatively independent of the final multi-label classification task for these high-definition images, the study presented in this paper may be applicable to other competitions and computer vision use cases beyond the scope of the PlantCLEF 2025 competition.

7. Acknowledgments

I would like to thank the Pl@ntNet team for organizing this competition, which enabled us to test new methods and avenues of research. Some of these methods were unexpected and daring, but they were all relevant and instructive.

Thanks are also extended to Jean Senellart and Grégoire Leboucher from Quandela [30] for their collaboration and for conducting experiments to improve the performance of the machine learning model using quantum photonics.

Gratitude is expressed to Rosine Choupe for facilitating the preprocessing experiment in Pl@ntNet by assisting with field photography and sharing her equipment to diversify the sensors used.

Finally, thanks are due to Guillaume Gomez for reviewing the source code and for providing accessible, comprehensive, and up-to-date Rust language documentation [38], which greatly facilitates learning.

8. Declaration on Generative AI

During the preparation of this work, the author used Mistral AI's LeChat in order to: Grammar and spelling check. The author also used DeepL Translation and DeepL Write to facilitate the translation of this document. After using these tools, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] H. E. Roy, A. Pauchard, P. Stoett, T. R. Truong, S. Bacher, B. S. Galil, P. E. Hulme, T. Ikeda, K. Sankaran, M. A. McGeoch, et al., Ipbes invasive alien species assessment: summary for policymakers, IPBES (2023).

[2] A. E. Magurran, Measuring biological diversity, John Wiley & Sons, 2013.

[3] 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.

[4] L. Picek, S. Kahl, H. Goëau, L. Adam, T. Larcher, C. Leblanc, M. Servajean, K. Janoušková, J. Matas, V. Čermák, K. Papafitsoros, R. Planqué, W.-P. Vellinga, H. Klinck, T. Denton, J. S. Cañas, G. Martellucci, F. Vinatier, P. Bonnet, A. Joly, Overview of lifeclef 2025: Challenges on species presence

prediction and identification, and individual animal identification, in: International Conference of the Cross-Language Evaluation Forum for European Languages (CLEF), Springer, 2025.
[5] H. Goëau, V. Espitalier, P. Bonnet, A. Joly, Overview of PlantCLEF 2024: multi-species plant identification in vegetation plot images, in: CEUR worshop proceedings, volume 3740 of *CEUR worshop proceedings*, Guglielmo Faggioli and Nicola Ferro and Petra Galuščáková and Alba García Seco de Herrera, Grenoble, France, 2024, pp. 1978–1988. URL: https://hal.inrae.fr/hal-04806900.
[6] 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.
[7] V. Espitalier, J.-C. Lombardo, H. Goëau, C. Botella, T. T. Høye, M. Dyrmann, P. Bonnet, A. Joly, Adapting a global plant identification model to detect invasive alien plant species in high-resolution road side images, Ecological Informatics 89 (2025) 103129. URL: https://www.sciencedirect.com/science/article/pii/S1574954125001384. doi:https://doi.org/10.1016/j.ecoinf.2025.103129.
[8] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: F. Pereira, C. Burges, L. Bottou, K. Weinberger (Eds.), Advances in Neural Information Processing Systems, volume 25, Curran Associates, Inc., 2012.
[9] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, Backpropagation Applied to Handwritten Zip Code Recognition, Neural Computation 1 (1989) 541–551. doi:10.1162/neco.1989.1.4.541.
[10] 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).
[11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, arXiv preprint arXiv:2010.11929 (2020).
[12] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers), 2019, pp. 4171–4186.
[13] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners, 2019. URL: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.
[14] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, R. Girshick, Masked autoencoders are scalable vision learners, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 16000–16009. URL: https://arxiv.org/abs/2111.06377.
[15] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, P. Bojanowski, Dinov2: Learning robust visual features without supervision, 2023. arXiv:2304.07193.
[16] T. Darcet, M. Oquab, J. Mairal, P. Bojanowski, Vision transformers need registers, 2024. arXiv:2309.16588.
[17] W. Saeed, C. Omlin, Explainable ai (xai): A systematic meta-survey of current challenges and future opportunities, Knowledge-Based Systems 263 (2023) 110273. URL: https://www.sciencedirect.com/science/article/pii/S0950705123000230. doi:https://doi.org/10.1016/j.knosys.2023.110273.
[18] Rust, 2010. URL: https://www.rust-lang.org, version 1.87.0.
[19] B. Stroustrup, The C++ Programming Language, 4th ed., Addison-Wesley, 2013.
[20] J. Perkel, Why scientists are turning to rust, Nature 588 (2020) 185–186. doi:10.1038/d41586-020-03382-2.
[21] A. Balasubramanian, M. S. Baranowski, A. Burtsev, A. Panda, Z. Rakamarić, L. Ryzhyk, System programming in rust: Beyond safety, in: Proceedings of the 16th Workshop on Hot Topics in Operating Systems, HotOS '17, Association for Computing Machinery, New York, NY, USA, 2017, p. 156–161. URL: https://doi.org/10.1145/3102980.3103006. doi:10.1145/3102980.3103006.

[22] L. Mazare, Candle, 2023. URL: https://github.com/huggingface/candle, version 0.9.1.
[23] G. van Rossum, F. L. D. Jr., Python Tutorial, 2020. URL: https://docs.python.org/3/tutorial/, accessed: 2025-06-04.
[24] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, 2019. URL: https://arxiv.org/abs/1912.01703. arXiv:1912.01703.
[25] M. Gustineli, A. Miyaguchi, I. Stalter, Multi-label plant species classification with self-supervised vision transformers, 2024. URL: https://arxiv.org/abs/2407.06298. arXiv:2407.06298.
[26] S. Chulif, H. A. Ishrat, Y. L. Chang, S. H. Lee, Patch-wise inference using pre-trained vision transformers: Neuon submission to plantclef 2024, 2024.
[27] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 10012–10022. URL: https://arxiv.org/abs/2103.14030.
[28] Y. Li, H. Mao, R. Girshick, K. He, Exploring plain vision transformer backbones for object detection, 2022. URL: https://arxiv.org/abs/2203.16527. arXiv:2203.16527.
[29] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, R. Girshick, Segment anything, 2023. URL: https://arxiv.org/abs/2304.02643. arXiv:2304.02643.
[30] Quandela, Merlin - photonic quantum machine learning framework, 2025. URL: https://merlinquantum.ai/, accessed: 2025-06-12.
[31] R. d'Andrimont, M. Yordanov, L. Martinez-Sanchez, P. Haub, O. Buck, C. Haub, B. Eiselt, M. van der Velde, Lucas cover photos 2006–2018 over the eu: 874 646 spatially distributed geo-tagged close-up photos with land cover and plant species label, Earth System Science Data 14 (2022) 4463–4472. URL: https://essd.copernicus.org/articles/14/4463/2022/. doi:10.5194/essd-14-4463-2022.
[32] S. Foy, S. McLoughlin, Utilising dinov2 for domain adaptation in vegetation plot analysis, 2024.
[33] K. Turkowski, Filters for common resampling tasks, in: Graphics gems, 1990, pp. 147–165.
[34] K. R. Rao, P. C. Yip, V. Britanak, Discrete cosine transform: Algorithms, advantages, applications, 1990. URL: https://api.semanticscholar.org/CorpusID:12270940.
[35] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv: 1409.1556 (2014).
[36] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, C. Ré, Flashattention: Fast and memory-efficient exact attention with io-awareness, 2022. URL: https://arxiv.org/abs/2205.14135. arXiv:2205.14135, local:PDF/80_divers/2022_Dao_FlashAttention_2205.14135v2.pdf.
[37] A. H. Salamah, K. Zheng, Y. Liu, E.-H. Yang, Jpeg inspired deep learning, 2025. URL: https://arxiv.org/abs/2410.07081. arXiv:2410.07081.
[38] G. Gomez, Tutoriel rust, 2025. URL: https://blog.guillaume-gomez.fr/Rust, accessed: 2025-06-11.
[39] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, 2015. arXiv:1512.03385.
[40] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, 2019. arXiv:1801.04381.
[41] M. Reck, A. Zeilinger, H. J. Bernstein, P. Bertani, Experimental realization of any discrete unitary operator, Phys. Rev. Lett. 73 (1994) 58–61. URL: https://link.aps.org/doi/10.1103/PhysRevLett.73.58. doi:10.1103/PhysRevLett.73.58.

A. Supplementary Experiment in Collaboration with Quandela

During the final week of the competition, a late collaboration was established with Quandela, a company specializing in quantum photonics and offering machine learning solutions that leverage this technology.

The Python libraries Perceval and Merlin [30] enable the design of quantum circuits and their seamless integration into a PyTorch pipeline as an additional processing layer, similar to a conventional linear

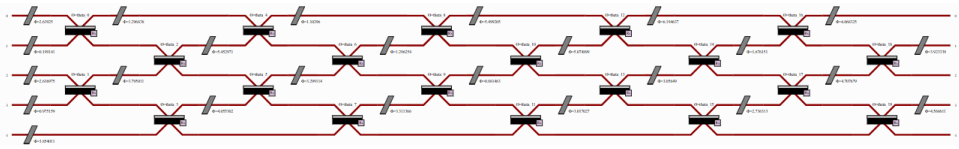

Figure 10: Here is a representation of what a circuit looks like. The grey parallelograms represent the phase shifters, the black rectangles beam splitters and the red lines are the different modes of the circuit. The input state is the number of photons provided to the circuit on every mode and the output state is the number of photons measured on every output mode of the circuit.

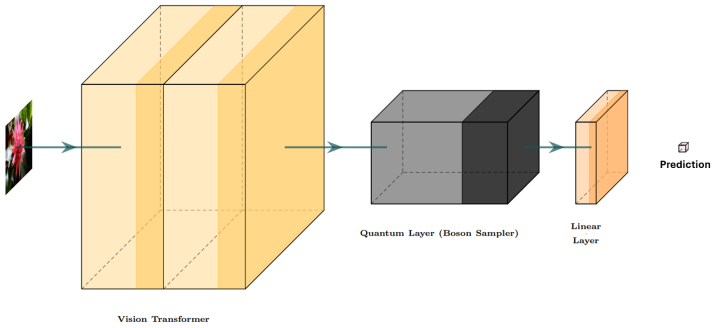

Figure 11: The figure represents the first architecture tested. The 768 output features of the VisionTransformer are then passed into the Quantum Layer in the amplitudes of the input state. Those states are simulated to obtain the probability of measuring every output state. Those probabilities are then transformed into 768 space vectors that are passed through the Linear Layer, predicting a probability for every plant to be on the image.

layer (with n input neurons and m output neurons).

Although the preliminary results were not conclusive at this stage, the experiments conducted as part of the participation in PlantCLEF 2025, in collaboration with Quandela, are briefly outlined here to facilitate their continuation or to inspire future research.

The computations were performed according to the principles of quantum photonics: the quantum circuit, known as the BosonSampler, consists of two main types of components: phaseShifters and BeamSplitters. The floating-point input values are encoded into the superposed input state given to the circuit to perform quantum computations. The phase shifters and beam splitters will then modify the probability of observing a photon in one specific mode, creating a superposed output state different from the input one. Those output probabilities are then converted back into floating-point values. This is called strong simulation, all of this is calculated thanks to a classical computer, calling this method a "quantum inspired".

It is also possible to run the same algorithm on an actual quantum computer, running the same experiment multiple times (called shots) measuring the output state, and thus obtaining an estimation of the output probabilities. See Figure 10 for an example of quantum circuit.

A first approach aimed to analyze the 2,105 quadrat images and predict the species present among the 7,806. The classification method used resizes the test images to the model's input size of 518x518 pixels. The quantum layer was inserted between the image encoder and the linear classifier, with input and output dimensions set to 768 floating-point values. The core idea behind this method is to create a new embedding based on the Vision Transformer one, using quantum to explore new areas of the latent space. This process involved semi-manual techniques, including manually testing different quantum circuits, as the Quantum Layer was not fully optimized for GPU execution at that time, leaving some room for improvement on this project. Indeed, optimizing such a high-dimensional quantum circuit using gradient descent proved challenging due to the large volume of data to be processed, especially in a limited amount of time. The linear classifier was fine-tuned using a limited set of training images (a few hundred thousand), selected by capping the number of train images of each of the 7,806 species.

A second approach was explored to reduce computational complexity and allow for multiple tests: the study was restricted to the 15 quadrat test images from the LISAH-JAS transect and the 78 species detected by the official public Python script submission (which processes images directly at a resolution of 518x518 pixels). Two successive linear layers with dimensions $768 \rightarrow 32 \rightarrow 78$ were trained (initialized by SVD factorization of the weights of the initial model's linear classifier) to limit the number of parameters according to the bottleneck principle ([39, 40]). The second linear layer was then removed, resulting in an image encoder that outputs deep features of size 32, containing the necessary information to classify the 78 selected species. A quantum layer of size 32x32 was inserted, followed by a linear classifier. The fine-tuning of the classifier was significantly accelerated. Several different quantum circuit hyperparameters and networks were tested (the Figure 11 represents one of them). However, tests conducted over just three days did not yield conclusive results. A longer experimentation period and deeper exploration of this new technology's capabilities might lead to more definitive outcomes, especially with the release of MerLin ([30]), a Pytorch compatible framework developed by Quandela to create Hybrid Quantum Classic algorithms in a very accessible and optimized way. In particular, it is now possible to compute gradient descent through a Quantum Layer. Also, previous theoretical research [41] has indicated that quantum photonics exhibits certain universality properties. It is anticipated that these properties could enhance the expressive power of neural networks, thereby improving classification performance.