Self-Supervised Learning (SSL) for Crop Diseases Classification using Sentinel-2 Data

Rabina Twayana
Université Bretagne Sud
Vannes, France
twayana.e2706754@etud.univ-ubs.fr

Ethel Awuor Ogallo
Université Bretagne Sud
Vannes, France
ogallo.e2406751@etud.univ-ubs.fr

Omowonuola Molayosi Akintola
Université Bretagne Sud
Vannes, France
akintola.e2406755@etud.univ-ubs.fr

***Abstract*—Accurate crop disease classification is crucial for precision agriculture, yet acquiring labeled satellite imagery remains expensive and time-consuming. This study investigates self-supervised learning (SSL) approaches to address the challenge of limited labelled data in remote sensing. We evaluate the MoCo v2 contrastive learning framework, pretrained on unlabeled data using a ResNet50 encoder, and then use the pretrained encoder, replacing its projection head with a linear classifier for supervised downstream classification. In addition, Prithvi geospatial foundation model (300M and 600M parameter variants), pretrained on NASA's HLS dataset, was also trained for classification. Using 900 labeled crop disease images from the ICPR 2026 competition, we trained and evaluated classification performance across four disease categories (Aphid, Blast, RPH, and Rust). Our results demonstrate that SSL-pretrained weights trained on task-specific unlabeled data achieve the best performance, with a macro F1-score of 67.8%. The Prithvi foundation models (300M and 600M parameters) achieved 53.8% and 55.0% Macro F1-score, respectively. These findings indicate that while large-scale foundation models offer promising transferability to agricultural tasks, task-specific SSL pretraining might be critical for optimal performance in label-scarce scenarios.**

Keywords— SSL, ResNet, MoCo, Classification, Foundation Model, Prithvi, Crop Disease

I. INTRODUCTION

Recent advances in deep learning approaches, particularly deep neural networks (DNNs), have demonstrated remarkable success in the interpretation of Earth Observation imagery [1]. Despite these achievements, a fundamental limitation of DNN-based applications is that they require large amounts of annotated training data [1], [2]. Although the increasing availability of Earth observation data allows access to vast amounts of information, generating a good-quality labelled dataset from these complex remotely sensed images poses significant challenges [2].

Manual annotation is labor-intensive, time-consuming, and expensive procedure, which remains difficult to obtain at scale [3]. For dynamic domains such as agriculture, the challenge is more prominent when the focus is on crop disease classification and monitoring. The symptoms of crop disease can differ significantly across crop types, geographic regions, and seasons, making consistent annotation extremely difficult [3]. Additionally, plant disease is highly time-sensitive; labels generated at one particular instance may lose validity after a certain time due to changes in phenological stage, environmental, and weather conditions [3], [4]. As a result, producing reliable, up-to-date labelled datasets for agricultural disease detection and classification demands continuous expert involvement and substantial resources, which significantly constrain the scalability of supervised learning approaches [4]. This is where employing a self-supervised learning (SSL) approach becomes valuable to learn meaningful representations from large, unlabeled Sentinel satellite datasets before fine-tuning on a limited set of labelled samples [5].

The primary goal of this study is to improve the accuracy and efficiency of crop disease classification in a label-scare scenario. To address the critical challenge of limited labelled data in agricultural remote sensing, Self-supervised learning is leveraged to learn robust and meaningful feature representations from large volumes of unlabelled Sentinel satellite imagery. These representations are then subsequently fine-tuned on a small set of labelled samples. Additionally, this study explores the use of pretrained geospatial foundation models, leveraging their learned representations from large-scale satellite imagery and finetuning them directly on a small set of labelled samples. As a result, this study aims to contribute to significant advancements in precision farming and sustainable agricultural practices, ultimately supporting global food security.

The paper is organized as follows: Section 2 describes the datasets and methods adopted, Section 3 presents our experiments, Section 4 presents our results and analysis, and Section 5 summarizes our conclusions.

II. METHODOLOGY

This study implements two deep learning approaches to improve crop disease classification performance: Momentum Contrast (MoCo), a self-supervised contrastive learning framework, and Prithvi, a foundation model pretrained on large-scale geospatial data.

A. DATASET AND PRE-PROCESSING

This study utilizes the datasets provided by the ICPR 2026 *Beyond Visible Spectrum: AI for Agriculture* competition, which includes data for two main tasks: self-supervised representation learning and supervised crop disease classification [6].

1) Pre-Processing for Contrastive Learning Methods (MoCo): The self-supervised learning (SSL) dataset was approximately 125 GB in compressed tar.gz format, and about 130 GB when uncompressed. It consists of a total of 32567 instances. It consists of unlabelled Sentinel-2 (S2A) time-series imagery organized by geographical location and acquisition timestamp, with each scene stored as individual TIFF files corresponding to the 12 spectral bands available for Sentinel 2 multispectral imagery. For each image per timestamp, the spectral bands were stacked to form a multispectral tensor. To ensure compatibility with pretrained architectures that expect 13 channels, we generated a dummy band B10, which was missing in the imagery, resulting in a 13-band input tensor. The images were resampled to a fixed spatial dimension and resized to match Band 1 because Sentinel-2 multispectral images have varying spatial resolutions. Spectral normalization was applied, where each spectral band was standardized using per-band mean and standard deviation statistics computed over the entire dataset.

The supervised dataset consists of labelled crop disease images organized in a folder-based class structure, where each folder corresponds to a specific disease category. Similar to the SSL data, we stack all the spectral bands for each sample to form a multispectral tensor, and a synthetic B10 band was also generated as an all-zero channel to obtain a consistent 13-channel input representation with all images resampled and resized as well to match Band 1 to ensure a similar spatial resolution across all bands. Additionally, a spectral band data normalization was applied, where a per-band mean and std normalization was computed over the entire labelled dataset, ensuring consistent intensity distributions across images. The dataset showed a high-class imbalance and was therefore split into a train and a validation subset, using an 80/20 stratified split to preserve overall class distribution, while the evaluation unlabelled dataset provided by the competition was set aside as the test set. To mitigate the class imbalance in the training subset, we computed class frequencies and derived inverse-frequency weighting factors that were used to implement a weighted random sampling strategy that is used during mini-batch creation. This strategy reduces bias towards majority classes, ensuring the model sees the minority classes more during training.

Finally, the inputs for both SSL pretraining and supervised classification are multispectral tensors resized to the model-specific input shape of 13 × 224 × 224 to match the expected dimensions of the model architectures.

2) Pre-processing for Prithvi Foundation Model: For the Prithvi-based approach, a different preprocessing pipeline was employed to align with the model's pretraining specifications. Prithvi was pretrained on 6 specific Sentinel-2 bands (B2, B3, B4, B8, B11, B12) corresponding to visible, near-infrared, and shortwave infrared wavelengths. Each labelled crop disease image was processed to extract only these 6 bands and stack them into a multispectral tensor. All bands were resampled to a uniform 10-meter spatial resolution to match Band 2's native resolution, as Prithvi expects consistent spatial dimensions across all channels. This part of the data processing was performed once and saved to storage to enable reuse across multiple training sessions in Google Colab's ephemeral compute environment.

Unlike the contrastive learning approaches, normalization for Prithvi used the model's original pretraining statistics rather than dataset-computed values. Specifically, band-wise normalization was applied using Prithvi's mean values and standard deviations, which were derived from the HLS dataset used during pretraining. This ensures consistency with the pretrained weight distributions and enables effective transfer learning. All other aspects, including the 80/20 stratified train/validation split and class imbalance mitigation through inverse-frequency class weighting in the loss function, were implemented identically to the contrastive learning approaches.

Finally, the inputs for Prithvi-based classification are multispectral tensors with shape 6 × 224 × 224, corresponding to the 6 Sentinel-2 bands required by the pretrained model architecture.

B. SELF-SUPERVISED LEARNING

For self-supervised pretraining, we adopt the MoCo v2 framework to learn generalizable representations from large unlabeled Sentinel imagery. It is a momentum-based contrastive learning approach where the task is to maximize agreement between augmented views of the same image (positive pairs) while contrasting them across many stored negative pairs. Our choice to use MoCo is driven mainly by the computational resources constraint. Compared to other self-supervised frameworks such as SimCLR [7], MoCo requires far smaller batch sizes during pretraining [8]. We adopt MoCo v2 instead of MoCo v1 due to its improved feature representation quality as it incorporates stronger data augmentations and a two-layer MLP projection head instead of a linear projection, which leads to more robust and transferable embeddings [8].

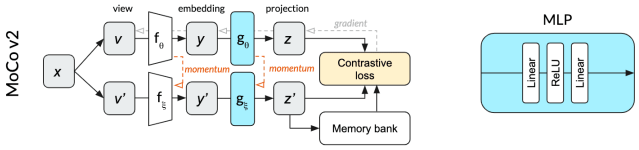

Figure 1: MoCo V2 Framework [9]

Figure 1 represents the MoCo v2 self-supervised contrastive learning framework. It consists of a feature encoder, a projection head (MLP), and a momentum encoder that maintains a queue of feature embeddings to stabilize contrastive training. MoCo splits the single network into an online network (top path) parameterized by θ and a momentum network (bottom path) parameterized by ξ. The online network is updated by stochastic gradient descent (backpropagation), while the momentum network is updated based on an exponential moving average of the online network weights. The momentum network allows MoCo to efficiently use a memory bank that stores many past projections as negative examples for contrastive loss. This memory bank provides a large set of negative samples without requiring huge batches [9]. During pretraining, the model learns to map different augmentations of the same image close together in the embedding space while pushing apart the embeddings of other images.

C. FOUNDATION MODEL APPROACH

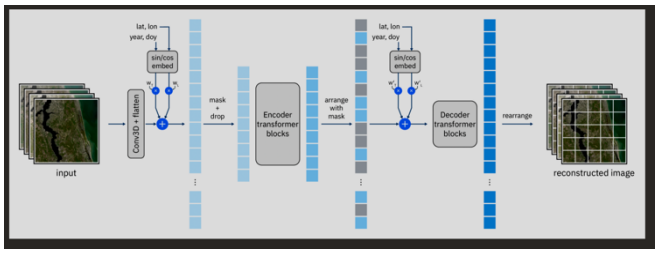

Figure 2 Architecture [10]

We adopt Prithvi, a geospatial foundation model pretrained on large-scale satellite imagery using masked autoencoding (MAE) as shown in Figure 2. Prithvi consists of an encoder, originally trained on NASA's Harmonized Landsat-Sentinel (HLS) dataset. Unlike contrastive learning methods, Prithvi learns representations by reconstructing masked patches of satellite imagery, enabling it to capture rich spatial and spectral patterns inherent in Earth observation data. The model is specifically designed for multi-spectral and multi-temporal satellite data, making it well-suited for agricultural applications. For this study, we leverage Prithvi's publicly available pretrained weights directly, without performing additional self-supervised pretraining on the competition's unlabeled data.

D. DOWNSTREAM TASK

For the MoCo Framework, the downstream classifier uses the pretrained MoCo ResNet50 encoder as a feature extractor. The projection head of the contrastive pretraining is removed, and a linear classification layer replaces it to map the 2048-dimensional feature embeddings to the target classes. The backbone is frozen for training the linear classification layer.

For the Prithvi model, a lightweight classification head is added to the pretrained encoder to map the output embeddings to the target classes. Two model variants are evaluated: Prithvi-300M with 300 million parameters producing 768-dimensional embeddings and Prithvi-600M with 600 million parameters producing 1024-dimensional embeddings. During fine-tuning, the encoder is frozen to train only the classification head.

E. LOSS FUNCTION AND METRICS

The self- supervised learning uses a contrastive loss function called InfoNCE [11], which is such that it brings representations of positive (similar) sample pairs closer while pushing representations of negative (dissimilar) samples apart.

$$L_{q,k+,k-} = -\log \frac{\exp(q \cdot k^{+}/\tau)}{\exp(q \cdot k^{+}/\tau) + \sum_{k^-} \exp(q \cdot k^{-}/\tau)}$$

where q is a query representation, k^+ is a representation of

The positive key sample, and $\{k^-\}$ are representations of the negative key samples. τ is a temperature hyper-parameter.

Because of the imbalanced nature of the dataset for the downstream task, we utilize the focal loss [12] which is defined as:

$$FL(p_t) = -(1 - p_t)^{\gamma} \log(p_t)$$

where p_t is the predicted probability for the true class, α_t is the weighting factor, and $\gamma>0$ is the tunable focusing parameter.

We assess the performance of the model with the Macro F1-score, which is the mean of all the per-class F1 scores, the harmonic mean of precision and recall per class. This score treats all classes equally, regardless of their weight.

III. EXPERIMENTAL SETUP

A. Self-Supervised Pretraining

We performed MoCo pretraining on two sets of data using ResNet50 backbone adapted to 13 Sentinel2 bands (224*224 patches) using strong spatial and radiometric augmentations to generate two views per image. The encoder is initialized from MoCo pretrained Sentinel2 weights and optimized with Adam at a learning rate of 1×10−4 and temperature of 0.15. Both the ResNet50 backbone (23.54M params) and MoCo projection head (26.22M params) are fully trainable during pretraining, totaling 47.76M trainable parameters out of 99.52M overall. ResNet50 backbone momentum (23.54M params) and projection head momentum (26.22M params) remain frozen. Table 1 summarizes the model parameters and pre-training setup. Data augmentation was applied following standard contrastive learning principles that include random resized cropping, horizontal and vertical flips, Gaussian blur, and brightness adjustments, which helps the model to learn invariant representations under different conditions.

1) Subset of data: A subset of data was created by randomly selecting 3000 scenes without replacement, which results in 15426 patches due to multiple time-series instances per scene. The configuration uses a batch size of 128, a memory bank of 2,048 negatives, and a maximum of 50 epochs with a batch size of 128. This experiment was trained in Kaggle resources.

2) Full dataset: The complete dataset comprises 32567 scenes, which results in 166,775 patches. All other hyperparameters were kept identical to the subset configuration, except for an increased memory_bank_size of 16,000, 100 max_epochs, and batch_size of 96.

TABLE I. PRETRAINING CONFIGURATIONS

Exp Id	***Dataset***	***Epoch***	***Batch Size***	***Memory Bank Size***	***#Parameter(M)***	
					Total	Trainable
1	Subset	50	128	2048	47.8	99.5
2	Full	100	96	16000	47.8	99.5

B. Downstreaming Task

As a baseline, we utilized a ResNet50 backbone initialized with Sentinel-2 MoCo pretrained weights. The training was conducted using a batch size of 16, AdamW optimizer, weight decay of 10^{-5}, and a learning rate of 10^{-3} over 100 maximum epochs. Due to class imbalance, we use focal loss and early stopping with patience of 30 epochs based on the macro F1-score on the validation set. Building on the baseline, we conducted several experiments and optimized the model hyperparameters using Optuna with a shared space for all the experiments. We defined the objective function with the batch sizes {8,16,24}, learning rate logarithmically of range 10^{-5} to 10^{-2}, weight decay logarithmically of range 10^{-6} to 10^{-3}, focal gamma of range {1.0-4.0}, and label smoothing {0.0-0.2}. The final hyperparameter configuration chosen was experiment-based. Experiment 1 involved a backbone initialized with SSL-pretrained weights obtained from training on a subset of the unlabeled data. Experiment 2 involved the same SSL-pretrained backbone where the downstream task data was augmented to reduce overfitting observed in experiment 1. Experiment 3 involved using a backbone initialized with SSL-pretrained weights trained on the full unlabeled dataset.

For Prithvi, we utilized the pretrained encoder initialized with weights from NASA's HLS V2 product. The training was conducted using a batch size of 16, AdamW optimizer, weight decay of 0.1, and a learning rate of $2x10^{-4}$ over 100 maximum epochs. Due to class imbalance, we use cross-entropy loss with inverse frequency class weighting and performance is monitored using F1-score on the validation set. The experiment began with the Prithvi-300M to evaluate the performance of the pretrained model with a frozen encoder backbone, training only the classification head. Subsequently, Prithvi-600M was evaluated to assess whether increased model capacity would improve classification performance, using identical training configurations.

The experiments were conducted within the Pytorch framework. The SSL MoCo pretraining was conducted using Kaggle GPU for the subset of data, while a high-end laptop with an Intel Core i9-14900HX (24 cores/32 threads), NVIDIA RTX 4090 Laptop GPU, 16 GB GPU Memory, and

64 GB RAM was used for the full dataset. The downstream task based on the MoCo encoder was trained on the Kaggle GPU environment using the dual NVIDIA Tesla T4 GPUs and Prithvi was trained on Google Colab environment using the T4 GPUs.

IV. Results and Discussions

A. Pre-Training

Table II presents the pre-training performance of two experimental configurations, reporting the number of epochs, total training time, final training loss, and standard deviation.

Experiment 1 (using a subset of data) achieved the lowest training loss (3.2313). The loss standard deviation (0.0159) is slightly higher than the other experiments, indicating marginally greater fluctuation during training. However, the lower loss suggests that this configuration converged faster and more effectively within fewer computational resources.

Experiment 2 (using the full dataset) was initially trained for 50 epochs (a) and later extended to 100 epochs (b). Despite training on the full dataset, experiment 2 resulted in a higher training loss (3.8461(a) and 3.5682(b)). However, the loss standard deviation is lower than that of experiment 1, which indicates slightly more stable training. Experiment 2(b) extended the training duration to 100 epochs, showing the gradual convergence (loss from 3.8461 to 3.5682) as compared to Experiment 2(a). However, despite the increased computational cost and amount of dataset, the loss still did not surpass the performance of Experiment 1.

TABLE II. Pre-training Performance

Exp Id	*Epoch*	*Training Time*	*Train Loss*	*Train Std*
1	50	3 hr 29 min	3.2313	0.0159
2 (a)	50	7 hr 53 min	3.8461	0.0143
2 (b)	100	15 hr 46 min	3.5682	0.0137

B. DownstreamingTask

Table III presents the performance metric of the experiments done for the downstream crop disease classification task. The baseline model achieved a Macro F1-score of 58.7%, which provided a reference for assessing the benefit of self-supervised learning pretraining.

TABLE III. Performance metrics for each experiment

Experiment	*Weights*	*Val Macro F1-score (%)*	*Evaluation Score (%)*
Baseline	ResNet50 S2 weights	58.7	81.25
Exp 1	SSL pretrained subset	**67.8**	**87.5**
Exp 2	SSL pretrained subset + aug*	64.9	**87.5**
Exp 3	SSL pretrained full	26.8	50.0
Exp 4	Prithvi 300M	53.8	75.0
Exp 5	Prithvi 600M	55.0	81.5

[a.] aug* is augmentation applied onto the labelled dataset

Experiment 1 uses the SSL-pretrained weights from a subset of the unlabeled dataset which shows an improvement on the Macro F1-score to 67.8%, showing that domain-specific pretraining enhances feature extraction for crop disease classification. This improvement suggests that SSL enabled the model to learn meaningful representations related to crop disease patterns. However, this experiment also showed signs of overfitting likely due to the relatively small size of the labeled dataset used during fine-tuning.

To address the overfitting, experiment 2 applied data augmentation during downstream training, resulting in a slight decrease in the Macro F1-score to 64.8% but with improved generalization. This is because data augmentation introduces variability in the training samples, allowing the model to learn more invariant features, improving model stability and reducing overfitting. Experiment 3 uses the SSL-pretrained weights provided from the full unlabelled dataset. This resulted in an unlikely performance drop, with the Macro F1-score decreasing to 26.8% with increased overfitting during fine-tuning. Experiments 4 and 5 uses the Prithvi weights, where experiment 4 achieved a Macro F1-score of 53.8% and experiment 5 improved performance to 55.0%. The analysis of the training curves revealed that both Prithvi's model training loss was still decreasing at 100 epochs. The performance may be attributed to insufficient training time, and this gap suggests potential for improvement through extended training beyond 100 epochs or incorporating task-specific unlabelled data for intermediate adaptation.

V. Conclusion

This study investigated the effectiveness of two deep learning approaches for crop disease classification using the Sentinel-2 dataset: MoCo, a self-supervised contrastive learning approach, and Prithvi, a foundation model-based approach. Our study shows that the SSL approach performs relatively better than the Prithvi model, considering the macro F1-score evaluation metric. SSL pre-training using a small subset helps to enhance the model performance. However, unexpected behaviour was observed in the downstream task when using the SSL-pretrained weights trained on the full dataset, leading to reduced performance. This outcome suggests that larger pre-training volumes do not necessarily guarantee better feature generalization and may introduce noise in contrastive learning. Furthermore, the application of the augmentation technique in the downstream task declines in performance, which implies that augmentations might distort the multispectral image characteristics.

Overall, the findings suggest that pre-training strategies, dataset volume, and augmentation techniques play critical roles in determining model performance. Further experiments and investigation are required to understand why the SSL pre-training using the entire dataset did not perform as expected.

Data and Code Availability

The full dataset used for SSL is accessed and available at "Beyond Visible Spectrum: AI for Agriculture 2026" Kaggle competition [Link]. The subset of the full dataset for SSL is uploaded in Kaggle datasets [Link]. The source code for model training and evaluation is available at GitHub [link].

Acknowledgemnts

We would like to acknowledge our colleague Kshitij Raj Sharma for giving access to the computational resources to run the SSL pre-training in full dataset, as uploading the huge dataset to Kaggle was not feasible, and therefore pre-training would not have been possible without this support.

REFERENCES

[1] S. Lahrichi, Z. Sheng, S. Xia, K. Bradbury, and J. Malof, "Is Self-Supervised Pre-training on Satellite Imagery Better than ImageNet? A Systematic Study with Sentinel-2," Feb. 15, 2025, *arXiv*: arXiv:2502.10669. doi: 10.48550/arXiv.2502.10669.

[2] Y. Wang, C. M. Albrecht, N. A. A. Braham, L. Mou, and X. X. Zhu, "Self-supervised Learning in Remote Sensing: A Review," Sep. 02, 2022, *arXiv*: arXiv:2206.13188. doi: 10.48550/arXiv.2206.13188.

[3] M. Shafay *et al.*, "Recent advances in plant disease detection: challenges and opportunities," *Plant Methods*, vol. 21, p. 140, Oct. 2025, doi: 10.1186/s13007-025-01450-0.

[4] J. Dong *et al.*, "Data-centric annotation analysis for plant disease detection: Strategy, consistency, and performance," *Front. Plant Sci.*, vol. 13, Dec. 2022, doi: 10.3389/fpls.2022.1037655.

[5] Y. Xu, Y. Ma, and Z. Zhang, "Self-supervised pre-training for large-scale crop mapping using Sentinel-2 time series," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 207, pp. 312–325, Jan. 2024, doi: 10.1016/j.isprsjprs.2023.12.005.

[6] "AgVision – ICPR 2026 Competition on "Beyond Visible Spectrum: AI for Agriculture." Accessed: Feb. 16, 2026. [Online]. Available: https://han-research.gitlab.io/Agvision/

[7] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations," Jul. 01, 2020, *arXiv*: arXiv:2002.05709. doi: 10.48550/arXiv.2002.05709.

[8] X. Chen, H. Fan, R. Girshick, and K. He, "Improved Baselines with Momentum Contrastive Learning," Mar. 09, 2020, *arXiv*: arXiv:2003.04297. doi: 10.48550/arXiv.2003.04297.

[9] "Understanding self-supervised and contrastive learning with 'Bootstrap Your Own Latent' (BYOL) - imbue." Accessed: Feb. 16, 2026. [Online]. Available: https://imbue.com/research/2020-08-24-understanding-self-supervised-contrastive-learning/

[10] D. Szwarcman *et al.*, "Prithvi-EO-2.0: A Versatile Multi-Temporal Foundation Model for Earth Observation Applications," Feb. 03, 2025, *arXiv*: arXiv:2412.02732. doi: 10.48550/arXiv.2412.02732.

[11] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum Contrast for Unsupervised Visual Representation Learning," Mar. 23, 2020, *arXiv*: arXiv:1911.05722. doi: 10.48550/arXiv.1911.05722.

[12] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," Feb. 07, 2018, *arXiv*: arXiv:1708.02002. doi: 10.48550/arXiv.1708.02002.