

Original papers

WeedCLR: Weed contrastive learning through visual representations with class-optimized loss in long-tailed datasets

Alzayat Saleh^a, Alex Olsen^b, Jake Wood^b, Bronson Philippa^a, Mostafa Rahimi Azghadi^{a,*}^a College of Science and Engineering, James Cook University, Townsville, 4814, QLD, Australia^b AutoWeed Pty Ltd, Townsville, 4814, QLD, Australia

ARTICLE INFO

Keywords:

Weed classification
Self-supervised learning
Long-tailed datasets
Deep learning

ABSTRACT

Image classification is a crucial task in modern weed management and crop intervention technologies. However, the limited size, diversity, and balance of existing weed datasets hinder the development of deep learning models for generalizable weed identification. In addition, the expensive labeling requirements of mainstream fully-supervised weed classifiers make them cost- and time-prohibitive to deploy widely, for new weed species, and in site-specific weed management. To address these challenges, this paper proposes a novel method for Weed Contrastive Learning through visual Representations (WeedCLR), that uses class-optimized loss with Von Neumann Entropy of deep representation for weed classification. WeedCLR leverages self-supervised learning to learn rich and robust visual features without any labels and applies a class-optimized loss function to address the class imbalance problem in real-world long-tailed weed datasets. WeedCLR is evaluated on two public weed datasets: CottonWeedID15, containing 15 weed species, and DeepWeeds, containing 8 weed species. WeedCLR achieves an average accuracy improvement of 4.3% on CottonWeedID15 and 5.6% on DeepWeeds over previous methods. It also demonstrates better generalization ability and robustness to different environmental conditions than existing methods without the need for expensive and time-consuming human annotations, which could reduce deployment time from days to hours. This is fundamental in reducing weed impacts, which can grow and spread very rapidly further affecting crops and reducing yield. These significant improvements make WeedCLR an effective tool for weed classification in real-world scenarios and allow for more rapid and widespread deployment of site-specific weed management and crop intervention technologies.

1. Introduction

Weed management is a critical issue in agriculture, as weeds compete with crops for resources and can significantly reduce crop yields (Rai et al., 2023; Xu et al., 2023). Existing weed management relies heavily on broadcast application of herbicides to entire paddocks to control weeds, including in areas of paddocks which do not contain weeds. This over-application of herbicide has both negative economic and environmental effects that could be minimized with site-specific weed control. In the past two decades, new approaches for site-specific weed management have been developed to detect weeds and only apply herbicide where it is needed. WeedIT¹ and WeedSeeker² uses near-infrared (NIR) sensing technology to detect weeds which are limited to weed control in-fallow application, or *green-on-brown* spraying. While deep learning based approaches have shown promise for in-crop application (Arsa et al., 2023; Dang et al., 2023), or *green-on-green* spraying. The latter approaches are on the cusp of achieving

significant uptake in the commercial arena (Coleman et al., 2022). However, supervised learning is the predominant deep learning method for these approaches, which requires human involvement to annotate the presence of weeds within each image. This annotation process, being highly time-consuming, stands as a formidable barrier to the widespread adoption of these approaches. In addition, the need for annotation significantly slows down the development and deployment of weed classification techniques, which in turn could lead to a wider spread of weeds and a higher impact on crops. Furthermore, the specialized knowledge required for accurately identifying weed species in images, particularly within specific cropping systems, makes it impractical to outsource this work to online paid annotation platforms. This is underscored by the fact that even among trained plant consultants, a noteworthy 12% error rate has been reported (Dyrmann et al., 2016). Self-supervised learning techniques have the ability to group clusters within datasets without the need for human annotation. There

* Corresponding author.

E-mail address: mostafa.rahimiazghadi@jcu.edu.au (M. Rahimi Azghadi).¹ <https://www.weed-it.com/>.² <https://ag.trimble.com/weedseeker2>.

is limited research on their use for weed recognition (Coleman et al., 2022).

Another challenge for deep learning approaches is that the ideal use case for site specific weed management is when weed pressure is low, which maximizes the reduction in herbicide usage. Consequently, a small proportion of target weed species creates a small sample of target images in the dataset, otherwise known as long-tailed datasets. This class imbalance is an obstacle that must be overcome. Below, we explain how we extend the state-of-the-art to develop WeedCLR to address the two aforementioned challenges.

The possibility for aligning the embedding vectors of enhanced representations of a given training sample has been demonstrated using self-supervised learning techniques. One such technique is contrastive learning (Chen et al., 2020a), which treats every training sample as a separate class and compares them. The InfoNCE contrastive loss (Oord et al., 2018) is commonly used to do this, pushing representations of negative pairs of instances further away in the embedding space while bringing representations of positive pairs of examples closer together. It is recognized that this method necessitates a high number of negative samples. On the other hand, non-contradictory approaches do not depend on specific negative samples. Examples of these approaches are redundancy reduction techniques (Zbontar et al., 2021), clustering-based techniques (Caron et al., 2020), and approaches that make use of specialized architecture design (Grill et al., 2020; Xie et al., 2022). These approaches have yielded encouraging results in a variety of applications and remain an active field of research.

Several self-supervised techniques modify the input data in order to derive a supervised signal in the form of a task that has been pre-designed (pretext). One such task of particular interest is the jigsaw puzzle task, which has been encoded as an invariant for contrastive learning. In contrast to this approach, we have used a multi-crop strategy that simply samples multiple random crops of the input image, having two distinct sizes — a larger and a smaller one. This makes it possible to learn representations and cluster assignments simultaneously in an end-to-end manner, offering a more effective and efficient technique for self-supervised learning.

Self-supervised learning has demonstrated efficacy in learning appropriate features and has surpassed supervised learning in certain subsequent transfer learning criteria. The theoretical study of self-supervised learning has sought to explain the underlying dynamics of these systems (Chen et al., 2020a). Despite their success, the mechanics of these techniques remain relatively obscure and not properly understood. Several studies have provided theoretical evidence that representations learned via contrastive learning are useful for downstream tasks (Tosh et al., 2021; Lee et al., 2020; Arora et al., 2019b). Additionally, it has been explained by Tian et al. (2021) why non-contrastive learning methods such as SimSiam (Xie et al., 2022) and BYOL (Grill et al., 2020) work, the prevention of total collapse being largely dependent on its input correlation matrix's and the predictor's eigenspace alignment.

In a linear neural network scenario, gradient descent has been shown to theoretically lead adjacent matrices to align in terms of implicit regularization (Ji and Telgarsky, 2019). Gradient descent has the ability to produce nuclear-norm minimization solutions, provided that there are aligned matrices (Gunasekar et al., 2018). Arora et al. (2019a) has expanded this idea to deep linear networks, and provides empirical and theoretical support for the ability of deep linear networks to extract low-rank solutions. Neural networks that are too parametrized typically discover steeper local minima (Saxe et al., 2019; Neyshabur et al., 2019; Barrett and Dherin, 2021). In this study, we employ Von Neumann Entropy (VNE) of deep representation as a regularization technique. In addition to VNE, this regularizer may also successfully regulate other qualities that are conceptually related, such as decorrelation and rank. Our approach takes into account the autocorrelation matrix's eigenvalue distribution, which characterizes the relationship between the representation's various properties.

The main objective of this study is to develop an efficient and effective weed classification method by employing our above-mentioned deep learning contributions in self-supervised learning and regularization. To achieve this objective, we developed WeedCLR, for weed classification in long-tailed datasets using a self-visual features learning approach. WeedCLR uses a class-optimized loss function to improve classification accuracy and leverages the power of self-supervised learning to extract meaningful visual features from images of weeds without the need for any human annotation (Magistri et al., 2023; Espejo-Garcia et al., 2023). These features are then used to train a classifier that can accurately distinguish between different weed species. We evaluate our method on two public weed datasets: CottonWeedID15 (Chen et al., 2022) and DeepWeeds (Olsen et al., 2019), and demonstrate its effectiveness in improving classification accuracy compared to state-of-the-art self-supervised learning approaches.

Our approach includes the use of Von Neumann Entropy (VNE) to optimize the representation space, which helps us to achieve a more desirable representation that avoids dimensional collapse and produces more useful embeddings. We also introduce a regularization that encourages the model to assign labels uniformly across all classes, preventing degenerate solutions where all labels are assigned to a single class. Additionally, by maximizing the likelihood of the correct class while balancing the probabilities of the wrong classes, our approach tries to optimize the model behavior for classes other than the correct class. This helps us to achieve better classification accuracy and improve the performance of our model in practical long-tailed weed datasets.

Specifically, our approach is a straightforward yet powerful single-stage end-to-end self-supervised classification and representation learning method for classifying weeds. Unlike prior unsupervised classification studies, it does not need any kind of pre-training, pseudo-labeling, expectation-maximization technique, or external clustering (Chen et al., 2020a; Caron et al., 2020). In contrast to other unsupervised representation learning studies, our method does not need a stop-gradient operation, external clustering, a memory bank, a second network, or negative pairings (Chen et al., 2020b; Grill et al., 2020; Caron et al., 2021).

Our main contributions are summarized as follows:

1. We present WeedCLR, to the best of our knowledge, the first self-supervised weed classification technique for long-tailed weed datasets.
2. WeedCLR advances state-of-the-art weed detection technology in practical production agriculture in three ways; (i) Managing long-tailed unbalanced datasets, which is a real-world problem in weed datasets. (ii) Improving the accuracy of the state-of-the-art by approximately 5%, which is fundamental for practical weed management practices. (iii) Significantly reducing the development and deployment time, which is essential for the timely containment of weeds and reducing their impact on crops.
3. WeedCLR achieves its power by incorporating several novel deep learning contributions namely (i) Leveraging the power of multi-crop strategy with self-supervised learning. (ii) Utilizing a class-optimized loss function to improve classification accuracy in long-tailed datasets. (iii) Using the Von Neumann Entropy (VNE) to optimize the representation space; (iv) Introducing a regularization that encourages the model to assign labels uniformly across all classes. And (v) optimizing the model behavior for classes other than the correct class by increasing the probability of the right class and decreasing the probability of the wrong classes.
4. We evaluate WeedCLR on two public weed datasets: CottonWeedID15 (Chen et al., 2022) and DeepWeeds (Olsen et al., 2019), and demonstrate its effectiveness in improving classification accuracy compared to state-of-the-art self-supervised learning approaches.

The rest of this paper is organized as follows: Section 2 presents our proposed WeedCLR approach in detail. Section 3, presents the experimental setup and implementation details of WeedCLR. Section 4 evaluates the performance of our approach on two benchmark datasets and compares it to several state-of-the-art methods. Section 5 investigates the effect of different factors on the DeepWeeds and CottonWeedID datasets. Section 6 discusses the limitations of our approach and future work. Finally, Section 7 summarizes the main findings of this paper and outlines the potential impact of our work.

2. Method

Despite the recent successes of self-supervised learning (SSL) methods, they are not as effective when dealing with unbalanced datasets. This is because many SSL methods rely on a hidden uniform prior, which distributes the data uniformly in the representation space. As a result, in a particular mini-batch, the model learns the most discriminative features. Class-specific features are the most discriminative ones that the model will learn when the input is uniformly distributed across classes. However, when using unbalanced data, the class may no longer be the most discriminative feature inside the mini-batch; instead, additional low-level information may be present, which will decrease performance on downstream classification tasks.

The most common practice for pretraining SSL models is to use curated datasets such as ImageNet and PASS, which are usually class-balanced and contain images with a single object prominently featured in the center. However, these datasets are not always representative of the data found in the wild, such as weed datasets, which are often unbalanced.

In this section, we present our WeedCLR method for training a classifier that can accurately classify two distinct augmented views of the same image sample. Our goal is to train a classifier that can classify the two views similarly while avoiding degenerate solutions. By doing so, we aim to improve the performance of SSL methods on unbalanced weed datasets. However, our proposed technique can be applied to any other image datasets, especially unbalanced ones.

A straightforward approach to this problem would be to minimize a cross-entropy loss function, as shown in Eq. (1).

$$\tilde{\mathcal{L}}(x_1, x_2) = - \sum_{y \in \mathcal{C}} p(y|x_2) \log p(y|x_1), \quad (1)$$

where $p(y|x)$ represents the probability of class y given input x , calculated as a row softmax of the matrix of logits S in one-hot encoded representation. This matrix is produced by our model (encoder + classifier), see Fig. 1, for all classes (represented by columns) and batch samples (represented by rows).

However, without additional regularization, this approach quickly converges to a degenerate solution, (i.e., the network predicts a constant y regardless of the input x). To address this issue, we propose optimizing our model's representation space by incorporating Von Neumann Entropy (VNE), and introducing a novel loss function called Class-optimized Loss (COL), as shown in Fig. 1. In the following subsections, we will explain the Representation Space Optimization and our proposed Class-optimized Loss (COL) in more detail.

2.1. Representation space optimization

Self-supervised learning techniques learn valuable representations by minimizing the distances between embedding vectors from augmented images (Fig. 1). This would lead to a collapsed solution in which the generated representation remains constant in the absence of further regularization (Fig. 2c). However, certain contrastive techniques limit total collapse by the use of a negative term that shifts the embedding vectors of distinct input images from one another (Chen et al., 2020b; Grill et al., 2020; Caron et al., 2021). Although contrastive techniques avoid complete collapse, they nevertheless suffer

a dimensional collapse when the embedding vectors span a subspace of a lower dimension than their dimension. (Fig. 2b).

To achieve the ideal representation space (Fig. 2a), we propose a method that uses Von Neumann Entropy (VNE), as shown in Fig. 1, to control the eigenvalue distribution of the autocorrelation matrix (Kim et al., 2023). By optimizing the representation space using VNE, we can achieve a more desirable representation that avoids dimensional collapse and produces more useful embeddings.

Von Neumann Entropy (VNE) is a mathematical formulation used to manipulate representation properties and is a measure of the diversity or spread of the eigenvalues of the autocorrelation matrix, which is an important property of a representation. By regularizing the VNE of the representation, the eigenvalue distribution can be effectively manipulated to refine the quality of the representation.

The representation's autocorrelation matrix, defined as $\mathcal{Z}_{\text{auto}}$, is a mathematical construct that describes the correlation between different components of a representation. The properties of the autocorrelation matrix are closely related to various representation properties, such as decorrelation, and rank. Decorrelation refers to the process of removing the correlation between different components of a representation, which can improve the quality of the representation. Rank is a measure of the dimensionality of the representation, which can affect its expressiveness and generalization ability. The representation matrix for a given mini-batch of N samples is $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_N]^T \in \mathbb{R}^{N \times d}$, where d is the representation vector's size. As in previous studies (Kim et al., 2023), we assume L_2 -normalized representation vectors satisfying $\|\mathbf{h}_i\|_2 = 1$. This assumption is important because it ensures that the representation vectors have a consistent scale and are not affected by differences in magnitude. By normalizing the vectors in this way, the method can focus on the distribution of the vectors rather than their absolute values. The representation's autocorrelation matrix is then defined as:

$$\mathcal{Z}_{\text{auto}} \triangleq \sum_{i=1}^N \frac{1}{N} \mathbf{h}_i \mathbf{h}_i^T = \frac{1}{N} \mathbf{H}^T \mathbf{H}, \quad (2)$$

where \mathbf{h}_i is a set of vectors, i ranges from 1 to N .

The autocorrelation matrix in Eq. (2) is denoted by $\mathcal{Z}_{\text{auto}}$, and is calculated by taking the outer product of each vector with itself, and then averaging over all N vectors. The outer product of a vector \mathbf{h}_i with itself is given by $\mathbf{h}_i \mathbf{h}_i^T$, where \mathbf{h}_i^T denotes the transpose of \mathbf{h}_i . The sum of all such outer products is then divided by N to obtain the average.

Alternatively, the expression for $\mathcal{Z}_{\text{auto}}$ can be written in terms of the matrix \mathbf{H} , where each column of \mathbf{H} corresponds to a vector \mathbf{h}_i . Specifically, $\mathcal{Z}_{\text{auto}}$ can be written as $\mathbf{H}^T \mathbf{H} / N$, where \mathbf{H}^T denotes the transpose of \mathbf{H} . This expression is useful for computing the autocorrelation matrix efficiently using matrix multiplication.

In the extreme case where $\mathcal{Z}_{\text{auto}} \rightarrow c \cdot \mathbf{I}_d$, where c is an adequate positive constant, \mathbf{I}_d a diagonal matrix, $\mathcal{Z}_{\text{auto}}$ eigenvalue distribution achieves full uniformity. This means that each feature in the representation contributes equally to the overall variance, which can be beneficial for self-supervised contrastive learning methods. The constant c is chosen to be an adequate positive value, which ensures that the matrix remains positive definite and invertible. This results in a full-rank representation and can prevent dimensional collapse in contrastive learning. Regularizing $\mathcal{Z}_{\text{auto}}$ is important because it permits a simple method to implement as a penalty loss.

The Shannon entropy across the eigenvalues of $\mathcal{Z}_{\text{auto}}$ is the definition of VNE of autocorrelation, as the following equation illustrates:

$$S(\mathcal{Z}_{\text{auto}}) \triangleq - \sum_j \lambda_j \log \lambda_j. \quad (3)$$

where, S denotes the Shannon entropy, which is a measure of the amount of uncertainty or randomness in a representation space. The subscript j represents the j th eigenvalue of the autocorrelation matrix, denoted by λ_j . The equation calculates the sum of the product of each

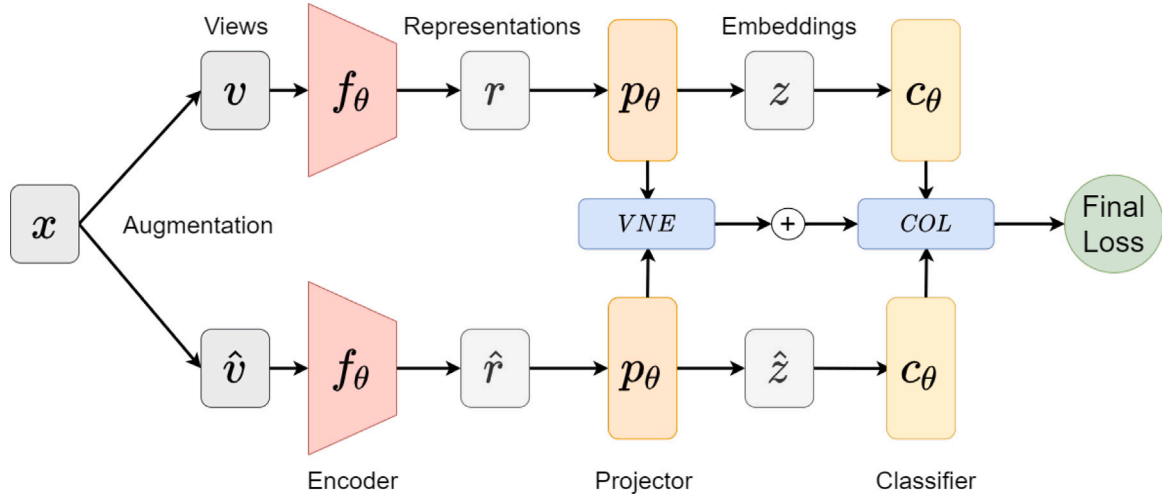


Fig. 1. Illustration of the architecture of our proposed WeedCLR model, which is designed for visual representation learning. The WeedCLR architecture uses a shared network consisting of a classifier (e.g., projection MLP + linear classification head) and an encoder f_θ (e.g., CNN) to analyze two augmented views of the same image. The eigenvalue distribution of the two views (Representations) is optimized using Von Neumann Entropy (VNE) to avoid dimensional collapse. By maintaining a uniform prior to class predictions, the Class-Optimized Loss (COL) is minimized to encourage similar class prediction while preventing degenerate solutions. The corresponding model performs an end-to-end, single-stage, unsupervised learning of representations and the underlying classes, allowing for efficient and effective classification of images in an unsupervised manner.

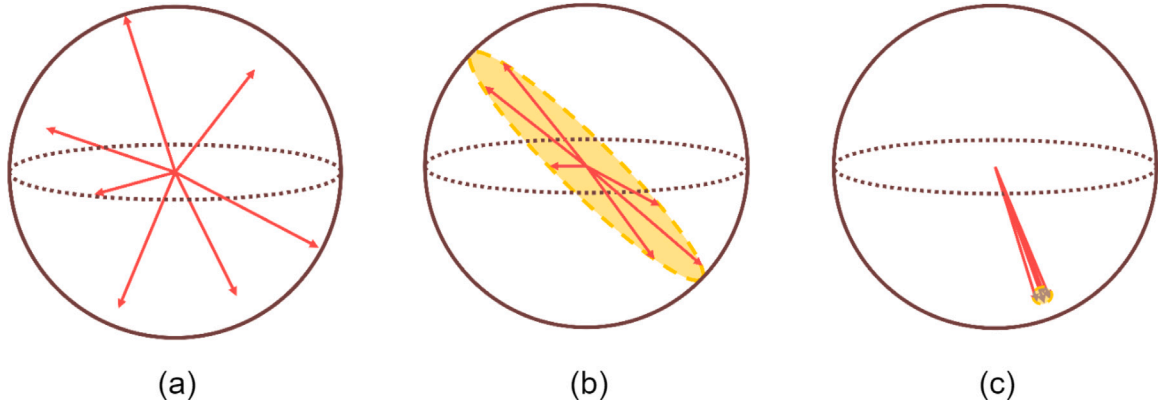


Fig. 2. Illustration of the representation collapse problem. (a) Ideal representation space (b) Dimensional collapse (c) Complete collapse. The embedding vectors for dimensional collapse only span a lower dimensional space. For complete collapse, the embedding vectors collapse to the same point.

eigenvalue and its natural logarithm, which is then multiplied by -1 to obtain the Von Neumann Entropy.

Implementing VNE regularization is simple. When training our model, we subtract $\alpha \cdot S(\mathcal{Z}_{\text{auto}})$ from the main loss \mathcal{L} .

$$\mathcal{L}_{\text{opt}} = \mathcal{L} - \alpha \cdot S(\mathcal{Z}_{\text{auto}}),$$

where \mathcal{L}_{opt} is the loss function that is being optimized, \mathcal{L} is the original loss function without any regularization, α is a hyperparameter that controls the strength of the regularization, $S(\mathcal{Z}_{\text{auto}})$ is the Von Neumann Entropy of the autocorrelation matrix $\mathcal{Z}_{\text{auto}}$ of the learned representation. The regularization term $-\alpha \cdot S(\mathcal{Z}_{\text{auto}})$ encourages the learned representation to have a more structured eigenvalue distribution, which can improve its quality.

In the following subsection, we will explain our proposed Class-optimized Loss (COL) function.

2.2. Class-optimized Loss (COL)

Our Class-Optimized Loss (COL) is composed of two components: The Uniform Prior Loss for the correct class and the Optimized Loss for the incorrect classes.

2.2.1. The uniform prior loss

The first component of our Class-Optimized Loss (COL) is the modified cross-entropy loss for the correct class and is derived by applying Bayes theorem and the law of total probability, resulting in Eqs. (4) and (5), which are used to compute the probability of a label given an augmented sample.

$$p(y|v_2) = \frac{p(y)p(v_2|y)}{p(v_2)} = \frac{p(y)p(v_2|y)}{\sum_{\tilde{y} \in [C]} p(v_2|\tilde{y})p(\tilde{y})}, \quad (4)$$

$$p(y|v_1) = \frac{p(y)p(y|v_1)}{p(y)} = \frac{p(y)p(y|v_1)}{\sum_{\tilde{v}_1 \in B_1} p(y|\tilde{v}_1)p(\tilde{v}_1)}, \quad (5)$$

where C is the number of classes, such that two augmented views (v_1, v_2) of the same sample are classified similarly, B represents a batch of N samples, with B_1 denoting the first set of augmentations for the samples in B . The term $p(v|y)$ refers to a column softmax of the matrix of logits S mentioned earlier.

In Eq. (4), the probability of y given v_2 is computed using Bayes theorem. It is equal to the product of the prior probability of y and the likelihood of v_2 given y , divided by the evidence or marginal probability of v_2 . The evidence is calculated by summing the product of the likelihood of v_2 given each possible label and the prior probability of each label over all possible labels. In Eq. (5), the probability of y given v_1 is also computed using the Bayes theorem. It is equal to the product of the prior probability of y and the likelihood of y given v_1 ,

divided by the prior probability of y . The prior probability of y cancels out in the numerator and denominator, so it is not necessary to compute it explicitly. The likelihood of y given v_1 is calculated by summing the product of the probability of y given each possible value of v_1 and the probability of each value of v_1 over all possible values of v_1 .

Assuming a uniform prior for $p(y)$ and a uniform distribution for $p(v_1)$, the proposed loss function is mathematically equivalent to the naive cross-entropy loss under the assumption of uniform $p(y)$ and $p(v)$. Our proposed loss function, shown in Eq. (6), takes into account the prior probabilities of the classes and samples.

$$\ell(v_1, v_2) = - \sum_{y \in [C]} \frac{p(v_2|y)}{\sum_{\bar{y}} p(v_2|\bar{y})} \log \left(\frac{N}{C} \frac{p(y|v_1)}{\sum_{\bar{v}_1} p(y|\bar{v}_1)} \right), \quad (6)$$

where $p(y)$ and $p(\bar{y})$ cancel out in Eq. (4), and $p(y)/p(\bar{v}_1)$ becomes $\frac{N}{C}$ in Eq. (5).

This cancellation occurs in Eq. (4) because both terms represent the probability of the same label for two different augmented views of the same sample. In Eq. (5), the ratio of $p(y)$ and $p(\bar{v}_1)$ is simplified to $\frac{N}{C}$, where N is the total number of samples and C is the number of classes. This simplification is possible because $p(\bar{v}_1)$ represents the probability of the augmented view of the sample, which is assumed to be uniformly distributed across all samples. Therefore, $p(\bar{v}_1)$ can be approximated as $\frac{1}{N}$, and the ratio $\frac{p(y)}{p(\bar{v}_1)}$ becomes $Np(y)$. Since the loss function is defined as the negative log-likelihood of the predicted labels, this simplification results in a term of $-\log \left(\frac{N}{C} \right)$ in the loss function. This term acts as a regularization that encourages the model to assign labels uniformly across all classes, preventing degenerate solutions when a single class is allocated to all labels. This means that a solution with an equal distribution of data among all classes by asserting a uniform prior on the conventional cross-entropy loss function is considered an optimal solution.

In practice, we employ a symmetric variant of this loss, as shown in Eq. (7). Empirical evidence suggests that this variant yields better results.

$$\mathcal{L} = \frac{1}{2} \left(\ell(v_1, v_2) + \ell(v_2, v_1) \right). \quad (7)$$

While our Uniform Prior Loss (Eq. (6)) has shown improved results in avoiding degenerate solutions (see Section 5), it primarily exploits information from the correct class and largely ignores information from incorrect classes. This can be attributed to the fact that predicted probabilities other than \bar{y} are zeroed out during the dot product calculation with one-hot encoded y . As a result, model behavior for classes other than the correct class is not explicitly optimized (see Fig. 3 left graph). Instead, their predicted probabilities are indirectly minimized when \bar{y} is maximized, given that probabilities must sum to 1. This effect is more pronounced in datasets with imbalanced class distributions. To address this limitation, an optimized loss for incorrect classes has been proposed as an additional regularization to our cross-entropy variant.

2.2.2. The optimized loss

The second component of our Class-Optimized Loss (COL) aims to explicitly optimize the model behavior for classes other than the correct class, by maximizing the likelihood of the correct class while neutralizing the probabilities of the incorrect classes.

The Optimized Loss $O(\cdot)$ is defined as the average of sample-wise entropies over incorrect classes in a mini-batch, as shown in Eq. (8).

$$\begin{aligned} O(\hat{y}_{\bar{c}}) &= \frac{1}{N} \sum_{i=1}^N H(\hat{y}_{i\bar{c}}) \\ &= -\frac{1}{N} \sum_{i=1}^N \sum_{j=1, j \neq g}^K \left(\frac{\hat{y}_{ij}}{1 - \hat{y}_{ig}} \right) \log \left(\frac{\hat{y}_{ij}}{1 - \hat{y}_{ig}} \right) \end{aligned} \quad (8)$$

where $\hat{y}_{\bar{c}}$ is the predicted probabilities of the incorrect classes, $H(\cdot)$ is the entropy function, y_i is one-hot vector representing the label of the

i th sample, \hat{y}_i is the predicted probability for each class for the i th sample, g is the index of the correct class, y_{ij} or \hat{y}_{ij} is the j th class (element) of y_i or \hat{y}_i , N and K are the total number of samples and the total number of classes.

Eq. (8) calculates the Optimized Loss, $O(\cdot)$, using a set of predicted probabilities, $\hat{y}_{\bar{c}}$. The entropy function is denoted as $H(\cdot)$, which is a measure of the uncertainty or randomness of a probability distribution. The sample-wise entropy is calculated by considering only the incorrect classes other than the correct class g . This means that the entropy is calculated based on the predicted probabilities of all classes except the correct class for a given sample. The predicted probability for each class, denoted by \hat{y}_{ij} , is normalized by one minus the correct probability (i.e., $1 - \hat{y}_{ig}$). This normalization ensures that the predicted probabilities sum up to one.

The term $\frac{\hat{y}_{ij}}{1 - \hat{y}_{ig}}$ can be interpreted as the predicted probability of observing class j for the i th sample, given that the correct class g does not occur. Optimizing on the incorrect entropy reduces \hat{y}_{ij} to $\frac{(1 - \hat{y}_{ig})}{(K-1)}$, where K is the total number of classes. This is because entropy is maximized when events have an equal chance of occurring. This fundamentally neutralizes the predicted probability of incorrect classes as K grows large. Maximizing the incorrect entropy “flattens” the predicted probabilities of incorrect classes $\hat{y}_{j \neq g}$. This means that the predicted probabilities of incorrect classes are reduced, making it less likely for the neural net h_θ to make incorrect predictions (see Fig. 3 right graph).

We hypothesize that when the predicted probabilities of incorrect classes are neutralized, the neural net h_θ generalizes better. This is because the likelihood of an incorrect class having a high enough predicted probability to “challenge” the correct class is reduced. In other words, by maximizing the incorrect entropy and “flattening” the predicted probabilities of incorrect classes (see Fig. 3), the neural net is able to make more accurate predictions and generalize better to new data.

2.3. WeedCLR model

WeedCLR is a novel approach to self-supervised end-to-end classification learning that simultaneously learns representations and labels in a single stage, end-to-end manner, as shown in Fig. 1. This approach optimizes for same-class prediction of two augmented views of the same sample and employs a variant of the cross-entropy loss with a uniform prior coded on the predicted labels to prevent all labels from being assigned to the same class. WeedCLR explicitly optimizes model behavior for classes other than the correct class by maximizing the likelihood of the correct class while neutralizing the probabilities of the incorrect classes. WeedCLR is scalable and simple to use. It does not require negative pairs, pre-training, expectation-maximization, external clustering, stop-gradient operation, second networks, or pseudo-labeling, in contrast to other common unsupervised classification learning techniques.

Algorithm 1 describes a training procedure for our WeedCLR model, also illustrated in Fig. 1. The algorithm takes as input a training dataset \mathbf{D} and outputs the model parameters θ . The training process consists of several steps, which are repeated for a specified number of training steps $n_{\text{train_steps}}$.

First, a mini-batch of data \mathbf{X} is obtained from the training dataset \mathbf{D} (line 3). This mini-batch is then augmented to produce two sets of data, \mathbf{v} and $\hat{\mathbf{v}}$ (line 4). These augmented data are then passed through an MLP encoder to extract features \mathbf{z} and $\hat{\mathbf{z}}$ (line 5). These features are optimized by \mathcal{Z}_{VNE} which is calculated using Eq. (3) (line 6). These features are then classified using a CLF function (classifier) to produce predicted class probabilities $\mathbf{y}_{\bar{c}}, \hat{\mathbf{y}}_{\bar{c}}$ (line 7).

Next, the COL loss term \mathcal{L}_{COL} is calculated using Eq. (10) (line 8). The final loss is then calculated as the sum of the VNE and COL loss terms (line 9), and the model parameters are updated using this final loss and an optimizer (line 10).

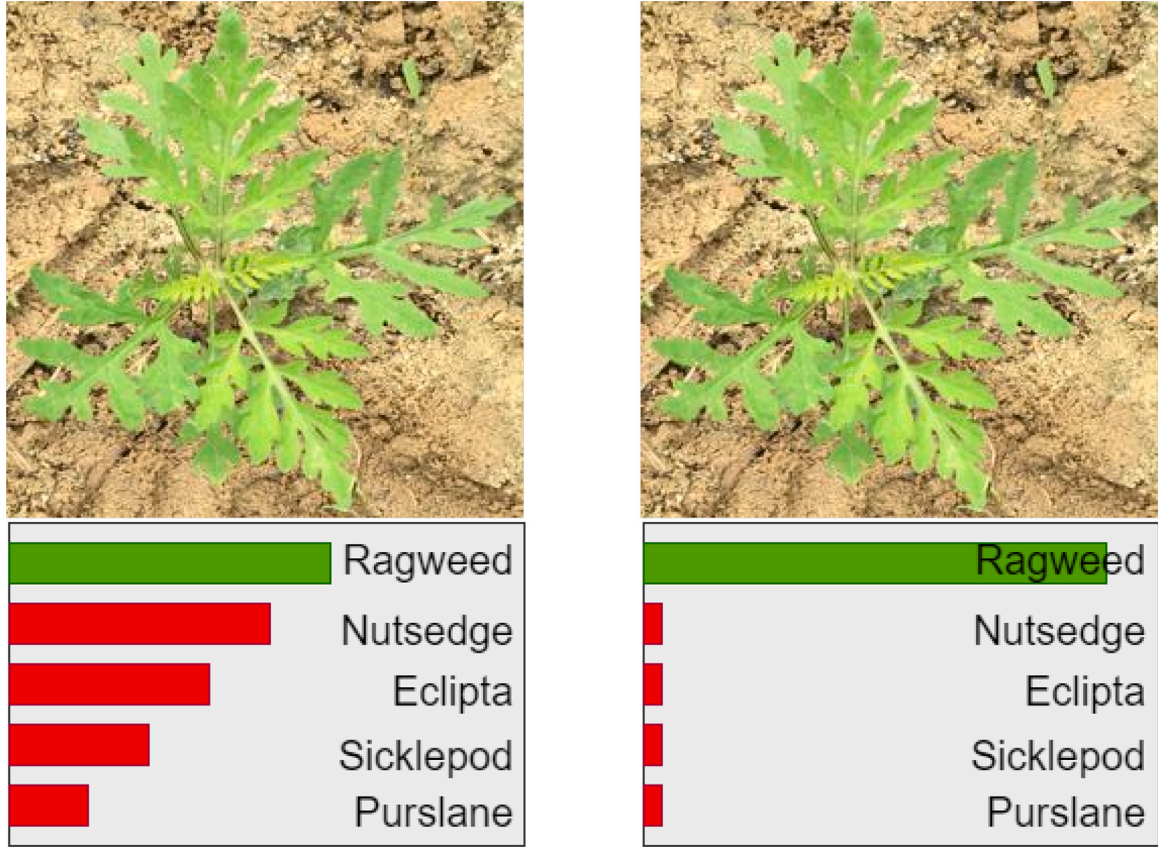


Fig. 3. Illustration of the top-5 predicted probabilities \hat{y} from two training paradigms for a sample image from the CottonWeedID15 (Chen et al., 2022) dataset, with the ground-truth class being “Ragweed”. The model used is ResNet-50. The left graph shows the predicted probabilities from the model trained with The Uniform Prior Loss only, while the right graph shows the predicted probabilities from the model trained with our proposed Class-optimized Loss (COL), which includes both The Uniform Prior Loss and The Optimized Loss. Compared to the right graph, the model in the left graph is confused by other classes such as “Nutsedge” and “Eclipta”, implying that it might be more vulnerable to adversarial attacks and limitations with generalization.

Algorithm 1: Training WeedCLR

input : Training dataset, $\mathbf{D} = \{\mathbf{X}_1, \dots, \mathbf{X}_n\}_{i=1}^N$
output: Model parameters, $\{\theta_1, \dots, \theta_{n_{layers}}\}$

```

1 initialization;
2 for  $t \leftarrow 1$  to  $n_{train\_steps}$  do
3    $\mathbf{X} \leftarrow \text{mini\_batch}(\mathbf{D}, t)$ ;
4    $\mathbf{v}, \hat{\mathbf{v}} \leftarrow \text{augmentation}(\mathbf{X})$ ;
5    $\mathbf{z}, \hat{\mathbf{z}} \leftarrow \text{MLP}(\text{encoder}(\mathbf{v}, \hat{\mathbf{v}}))$ ;
6    $\mathcal{Z}_{\text{VNE}} \leftarrow -\alpha \cdot S(\mathcal{Z}_{\text{auto}})$ . ▷ Eq. (3);
7    $\mathbf{y}_{\hat{\mathbf{e}}}, \hat{\mathbf{y}}_{\hat{\mathbf{e}}} \leftarrow \text{CLF}(\mathbf{z}, \hat{\mathbf{z}})$ ;
8    $\mathcal{L}_{\text{COL}} \leftarrow H(\mathbf{y}_{\hat{\mathbf{e}}}, \hat{\mathbf{y}}_{\hat{\mathbf{e}}}) + \beta \tilde{O}(\hat{\mathbf{y}}_{\hat{\mathbf{e}}})$  ▷ Eq. (10);
9   final_loss  $\leftarrow \mathcal{L}_{\text{COL}} + \mathcal{Z}_{\text{VNE}}$  ▷ Eq. (11);
10  optimizer.step(final_loss);

```

As shown in Algorithm 1, this work combines the Uniform Prior Loss and the Optimized Loss with a single entropy during the training process (see line 8 in Algorithm 1). To balance the Uniform Prior Loss, Eq. (7), and the Optimized Loss, Eq. (8), a coefficient β is introduced to the Optimized Loss, as shown in Eq. (9).

$$\beta = \frac{\gamma}{K-1} \quad (9)$$

where γ is a modulating factor, and K is the number of classes.

This equation calculates the coefficient β as the ratio of the modulating factor γ to the total number of classes minus one ($K-1$). The modulating factor γ should be tuned to decide the amount that optimizes the Uniform Prior Loss, for example, $\gamma = -1$ ($\gamma < 0$).

The proposed loss, named Class-optimized Loss (COL), is defined in Eq. (10).

$$\mathcal{L}_{\text{COL}} = H(\mathbf{y}_{\hat{\mathbf{e}}}, \hat{\mathbf{y}}_{\hat{\mathbf{e}}}) + \beta \tilde{O}(\hat{\mathbf{y}}_{\hat{\mathbf{e}}}) \quad (10)$$

This equation calculates \mathcal{L}_{COL} as the sum of the Uniform Prior Loss $H(\mathbf{y}_{\hat{\mathbf{e}}}, \hat{\mathbf{y}}_{\hat{\mathbf{e}}})$ and the product of the coefficient β and the Optimized Loss $\tilde{O}(\hat{\mathbf{y}}_{\hat{\mathbf{e}}})$.

The final loss is then calculated by combining COL Eq. (10) and VNE loss Eq. (3), as shown in equation Eq. (11).

$$\mathcal{L}_{\text{COL}} + \mathcal{Z}_{\text{VNE}} \quad (11)$$

This approach allows for explicit optimization of model behavior for both correct and incorrect classes while taking into account information from both. To visualize the effect of our Class-Optimized Loss (COL), Fig. 4 shows the embeddings for DeepWeeds and CottonWeedID15 (Chen et al., 2022) test images obtained from two self-supervised training paradigms. On the left, the embeddings were obtained by training with cross-entropy only, while on the right, the embeddings were obtained by training with our COL. The model used is ResNet-50, and the embeddings represent the vector representation before feeding to the MLP and classification heads. The embeddings were projected to 2D vectors using t-SNE for visualization purposes. Compared to the left images, the clusters of each class in the right images are small in terms of intra-cluster distance. Furthermore, the boundaries of the clusters in the right images are distinct and clear, which contributes to more reliable and accurate classification outcomes.

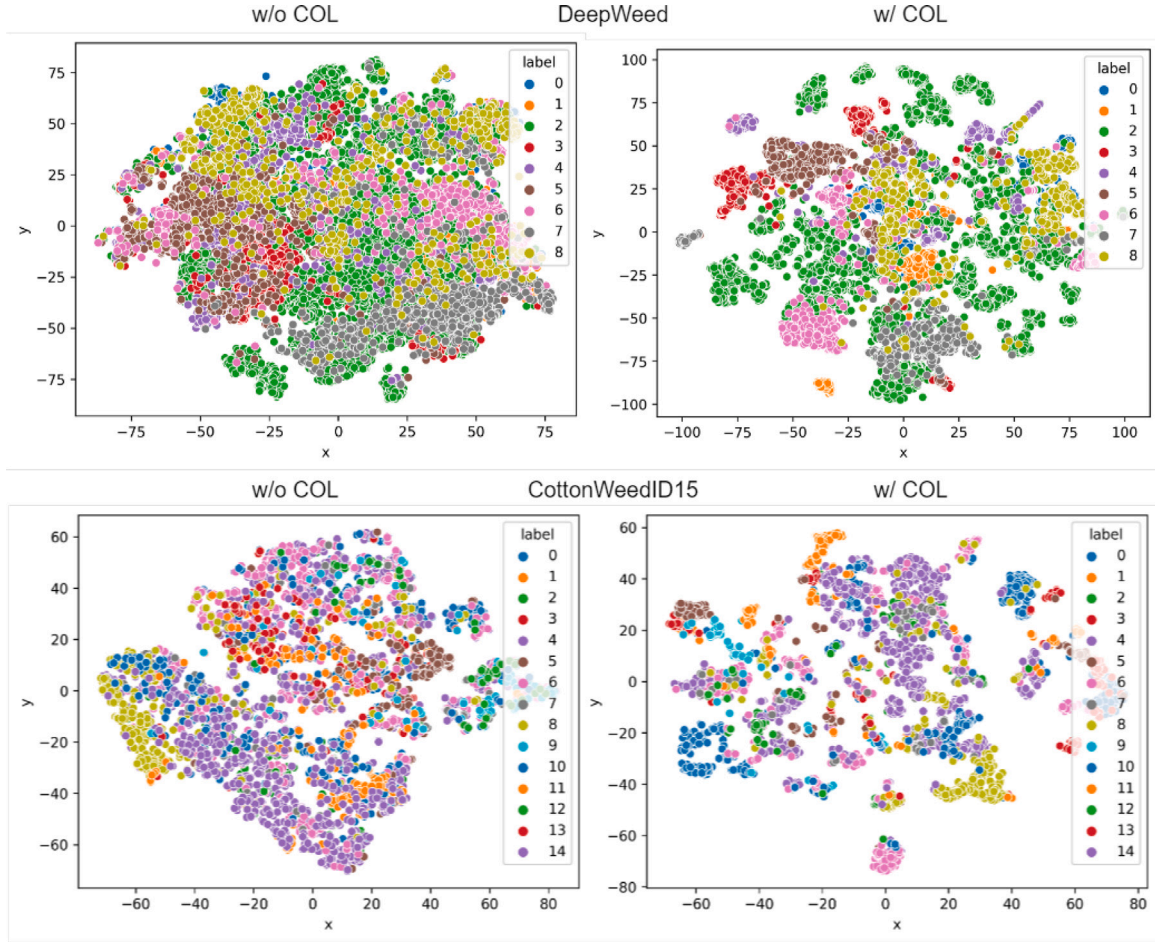


Fig. 4. The embeddings for DeepWeeds and CottonWeedID15 test images obtained from two self-supervised training paradigms. On the left, the embeddings were obtained by training with cross-entropy only, while on the right, the embeddings were obtained by training with our COL. The model used is ResNet-50, and the embeddings represent the vector representation before feeding to the MLP and classification heads. The embeddings were projected to 2D vectors using t-SNE for visualization purposes.

3. Experiments

This section first briefly overviews the experimental setup and implementation details and then presents experimental results demonstrating the effectiveness of our proposed method for weed classification in long-tailed datasets. We conducted experiments on two datasets of weed images, comparing the performance of our method to several baseline approaches.

3.1. Datasets

The first dataset used in our experiments is the DeepWeeds dataset (Olsen et al., 2019), which consists of 17,509 images capturing eight different weed species native to Australia in situ with neighboring flora. The images were collected from weed infestations in eight rangeland environments across northern Australia. The second dataset is the CottonWeedID15 dataset (Chen et al., 2022), which consists of 5187 RGB images of 15 weeds that are common in cotton fields in the southern U.S. states. These images were acquired by either smartphones or handheld digital cameras, under natural field light conditions and at varied stages of weed growth in 2020 and 2021. Fig. 5 compares the number of images per class for the two imbalanced datasets, CottonWeedID15 (left) and DeepWeeds (right). Both datasets have a long-tailed distribution, with some weed species being much more common than others. These datasets provide a valuable resource for the development and evaluation of self-supervised learning methods for weed identification.

3.2. Architecture

In our experiments, we employed a ResNet-50 architecture, a commonly utilized backbone in self-supervised learning studies. We also experimented with ResNet-9 and ResNet-18 architectures. The ResNet-(9, 18, 50) backbones were initialized randomly. The projection heads used in the experiments were two-layer MLPs with sizes of 4096 and one-layer with sizes of 128, respectively, and included batch normalization, leaky-ReLU activations, and ℓ_2 normalization after the last layer. Four classification heads were placed on top of the projection head MLPs, corresponding to 0.5K, 1K, 1.5K, and 2K classes, respectively. Without an additional bias term, each classification head was a simple linear layer. The temperature τ_{row} was set to 0.1 for the row-softmax and τ_{col} for the column-softmax. The evaluation for unsupervised classification was conducted using two different classification heads. For linear probe evaluation, the MLP was removed and replaced with a single linear layer of 15 and 9 classes for CottonWeedID and DeepWeeds datasets, respectively. A schematic diagram of the architecture is depicted in Fig. 1.

3.3. Image augmentations

In our experiments, we employed the data augmentations of SimCLR (Chen et al., 2020a), including color jittering, Gaussian blur, and random flips. We also utilized a multi-crop strategy, with two global views of size 128×128 and four local views of size 64×64 , as well as nearest neighbor augmentation, with a queue set to 200. The multi-crop strategy involves creating multiple crops of different sizes from

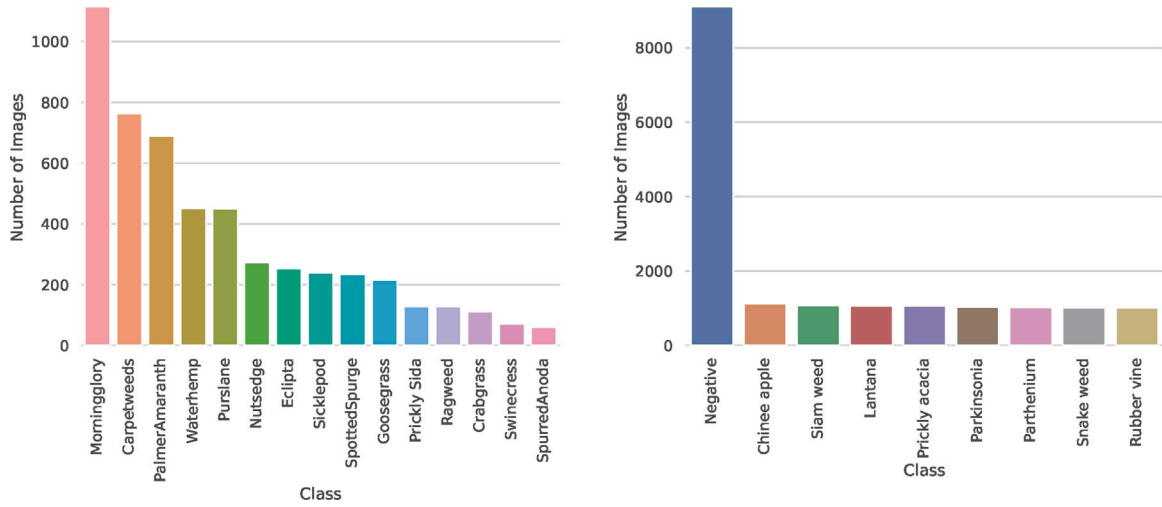


Fig. 5. Comparison of the number of images per class for two imbalanced datasets: CottonWeedID15 (Chen et al., 2022) (left) and DeepWeeds (Olsen et al., 2019) (right).

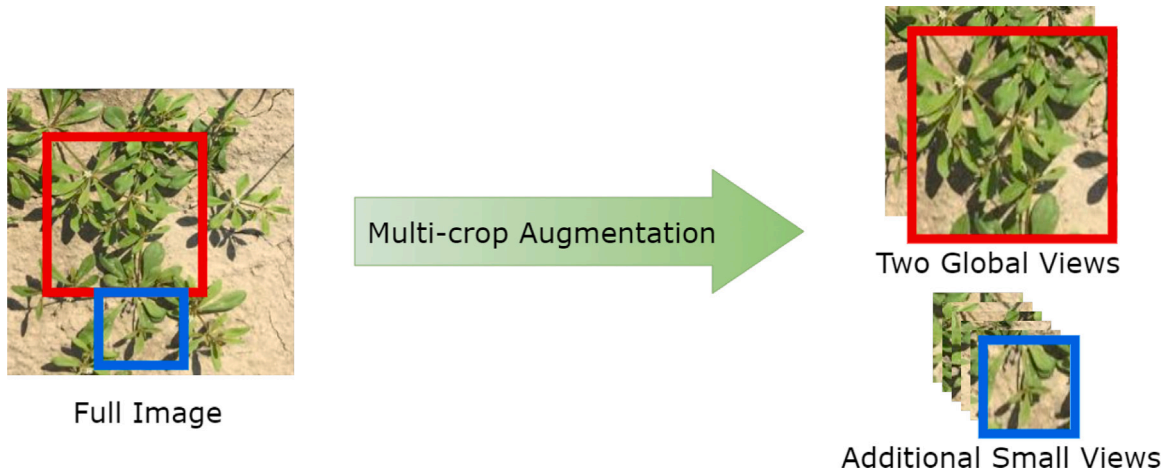


Fig. 6. Multi-Crop Augmentation: This process transforms a single image into $V+2$ distinct views, comprising two global perspectives and V small resolution zoomed perspectives. By doing so, it introduces a higher level of diversity into the training data, thereby enhancing the robustness and generalization ability of the trained model.

the same image for training a model, see Fig. 6. This approach was introduced in the SwAV (Caron et al., 2020). In SwAV, instead of using a fixed size for cropping images, a multi-crop strategy is employed where two larger crops and up to four smaller crops are taken from the same image. This has been shown to boost the performance of the model compared to other approaches that use a fixed crop size. The proposed approach is simple, yet efficient, and can be used in several self-supervised methods to consistently improve performance.

3.4. Experimental setup

We used an SGD optimizer (You et al., 2017) with a learning rate of 4.8 and weight decay of 10^{-6} for unsupervised pre-training and classification. For the first 10 epochs, the learning rate was scaled up linearly from 0.3. After that, it was reduced using a cosine scheduler for 390 epochs, ending at 0.0040, for a total of 400 epochs. We used a batch size of 256 on a single NVIDIA GeForce RTX 2080 Ti GPU.

We measured the quality of our WeedCLR model representations using two approaches. The first approach involves using the k-nearest neighbors (KNN) classifier, which makes classifications based on proximity to other data points. The second approach, known as linear probing, involves using the pre-trained model as a feature extractor.

Given labeled examples (X, Y) , the model is applied to X to produce features f_X . A linear classifier is then trained on (f_X, Y) . Linear probing is based on the idea that effective features should linearly separate the classes in transfer tasks. This approach helps to distinguish between the quality of features and the complexity of the model architecture.

For the linear probe evaluation, we used a similar experimental approach to KNN and used an SGD optimizer (You et al., 2017) with a learning rate of 0.1 and no weight decay. A cosine scheduler for 100 epochs was used to decrease the learning rate. Our NVIDIA GeForce RTX 2080 Ti GPU was used with a batch size of 256.

3.5. Online training accuracy

In this section, we compare the performance of our WeedCLR method with other state-of-the-art self-supervised learning methods on the DeepWeeds and CottonWeedID15 datasets during training. Fig. 7 shows that WeedCLR learns faster, reaching higher online KNN accuracy in fewer epochs than the other methods. With an increased ResNet from 9 to 18 and more epochs, WeedCLR still achieves high accuracy and maintains its advantage over the other methods. Our approach's effectiveness is demonstrated by these results in improving classification accuracy compared to state-of-the-art self-supervised learning methods.

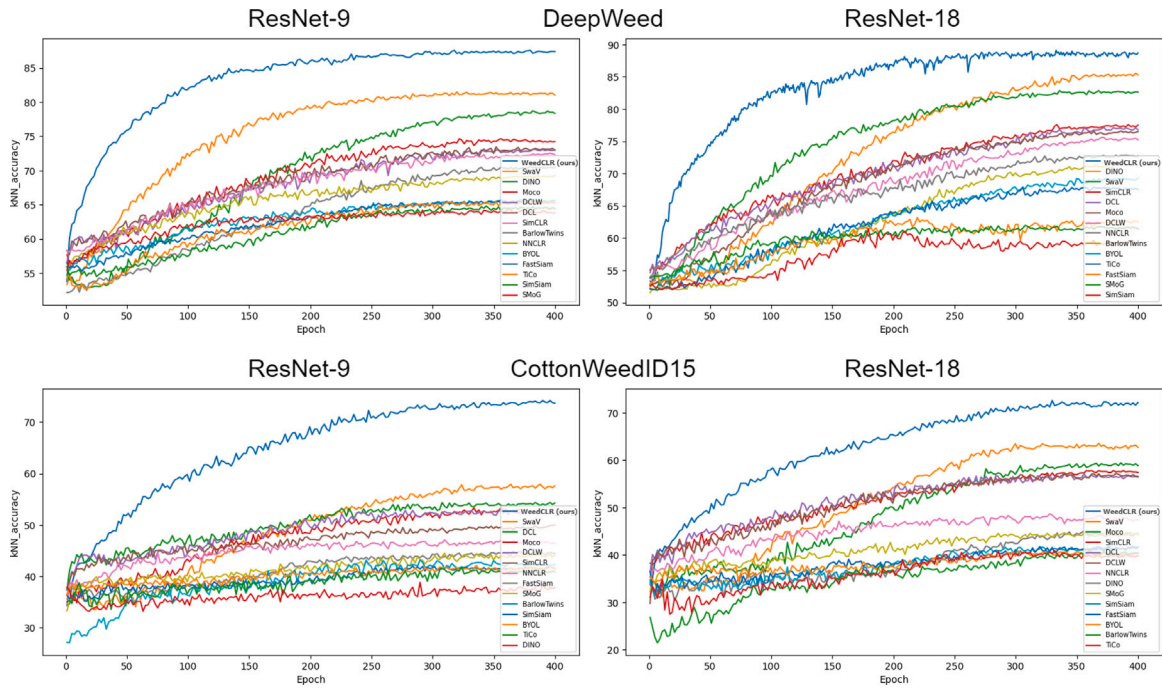


Fig. 7. Performance comparison of our WeedCLR method with other state-of-the-art self-supervised learning methods on the DeepWeeds and CottonWeedID15 datasets. Our results show that WeedCLR learns faster, reaching 76% online KNN accuracy in just 50 epochs, while SwaV (Caron et al., 2020) requires 200 epochs to achieve the same level of accuracy. With an increased ResNet from 9 to 18 and more epochs, WeedCLR achieves 89% accuracy on DeepWeeds and 73% accuracy using no momentum encoder, a small batch size, and fewer stored features. These results demonstrate the effectiveness of our approach in improving classification accuracy compared to state-of-the-art self-supervised learning methods. The figure may be better viewed online where you can zoom in for more detail.

The following section quantitatively compares our method with other state-of-the-art methods in more detail.

4. Results

In some studies, an alternative approach to our utilized linear probe evaluation or KNN may be employed to measure the quality of representations. This involves fine-tuning the pre-trained model for image classification tasks. The process includes adding a small classification head to the model and adjusting all weights accordingly. However, it is important to note that during fine-tuning, one model may outperform another not necessarily because of superior pre-training, but possibly because its architecture is more compatible with the downstream task at hand. Therefore, we did not use fine-tuning as a representation quality metric in our study. Accordingly, our approach consists of a pre-training stage followed by a linear probe and a KNN classifier. In pre-training, we explore three variants of the ResNet architecture (ResNet-9, 18, 50) as a backbone.

In our experiments, similar to the online training accuracy test we performed, we evaluated the performance of our approach, WeedCLR, against other self-supervised learning methods using both KNN classifier and linear probe evaluations on the DeepWeeds and CottonWeedID datasets. Our results, presented in Tables 1, 2, 3, and 4, demonstrate that WeedCLR outperformed other methods in terms of top-1 and top-5 accuracy for all three variants of the ResNet architecture (ResNet-9, 18, 50). Moreover, Fig. 8 compares the performance of our approach, WeedCLR, against other self-supervised learning methods using both KNN classifier and linear probe evaluations using ResNet-50.

4.1. DeepWeeds dataset

The results of the KNN classification and linear probing experiments are shown in Tables 1 and 2, respectively. WeedCLR achieved the best top-1 and top-5 accuracies for all three ResNet architectures on both

evaluation metrics. Our method, WeedCLR, outperformed the second-best model across all metrics and ResNet architectures. This superiority can be attributed to WeedCLR's ability to learn more discriminative features, which is crucial for accurate classification.

When compared to the supervised model, WeedCLR shows competitive performance. This is particularly noteworthy given that WeedCLR is a self-supervised method, which typically has a harder task as it does not have access to label information during training.

Interestingly, the top-5 accuracy of WeedCLR is very close to that of the supervised model. This suggests that while the top-1 predictions might differ, the set of top-5 predictions between the two models are quite similar. This could be due to both models learning similar feature representations for the classes, leading to similar predictions among the top-5 classes.

In conclusion, these results demonstrate the effectiveness of WeedCLR in learning discriminative features for weed classification, even in a self-supervised setting.

4.2. CottonWeedID dataset

The results of the KNN classification and linear probing experiments are shown in Tables 3 and 4, respectively. WeedCLR consistently outperforms other methods in most of the evaluations, demonstrating its robustness and effectiveness. It is particularly noteworthy that WeedCLR achieves the best top-1 and top-5 accuracy across all three backbones using the KNN classifier. This suggests that the features learned by WeedCLR are highly discriminative, enabling accurate nearest neighbor classification.

In the linear probing evaluation, WeedCLR continues to show strong performance, achieving the best results in 5 out of 6 tests. The only exception is for ResNet-9 backbone for Top-5 accuracy, where WeedCLR comes second. This could be due to the lower capacity of ResNet-9 compared to the other architectures, which might limit the effectiveness of the learned representations.

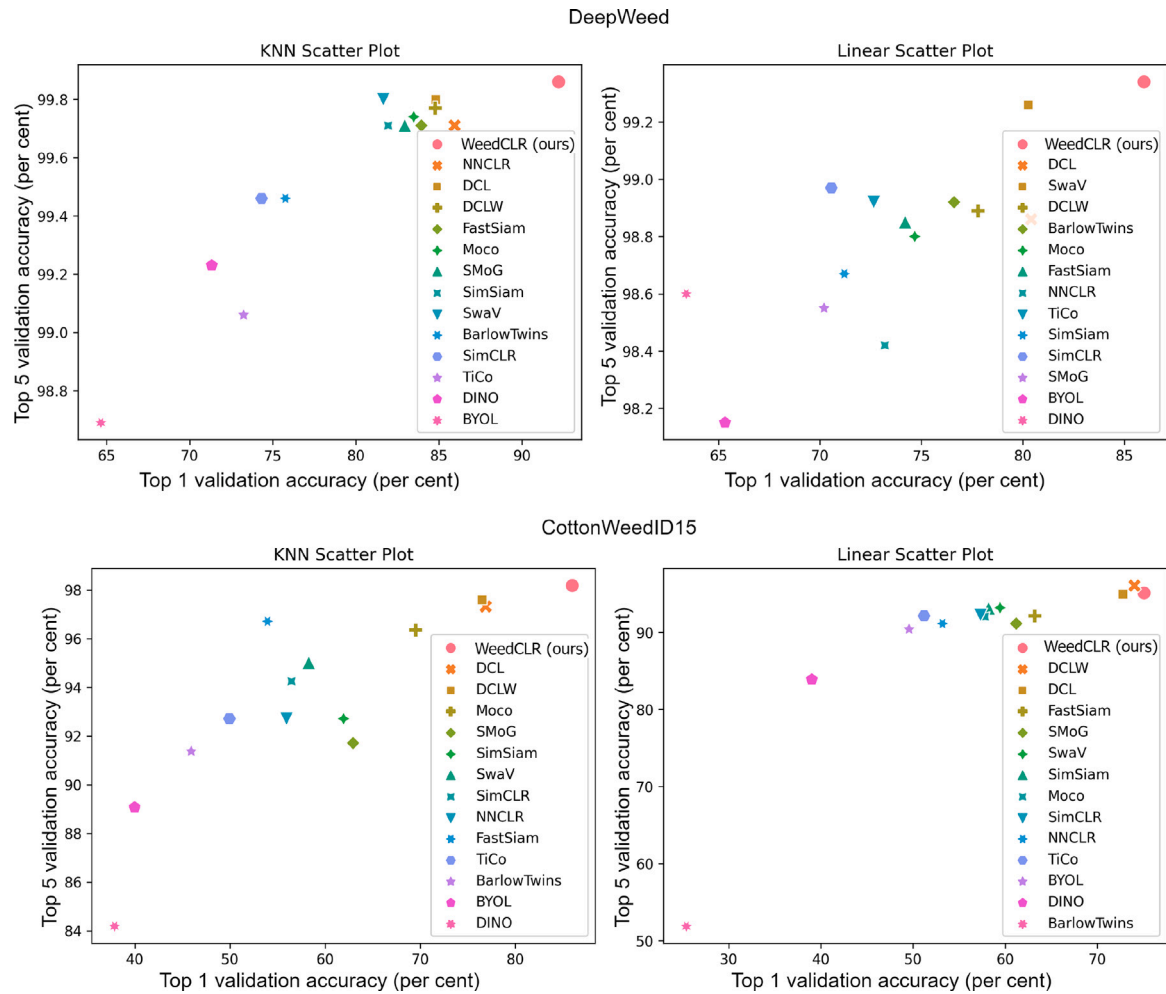


Fig. 8. Performance comparison of our approach, WeedCLR, against other self-supervised learning methods using both KNN classifier and linear probe evaluations on the DeepWeeds and CottonWeedID datasets. Our results demonstrate that WeedCLR outperformed other methods in terms of top-1 and top-5 accuracy for both datasets and accuracy metrics. For more details, please see Tables 1, 2, 3, and 4. The figure may be better viewed online where you can zoom in for more detail.

Overall, the results of our experiments demonstrate that WeedCLR is a powerful self-supervised learning method for plant classification. Our approach can be used to learn high-quality representations that are capable of distinguishing between different plant species. In addition to our quantitative results, we also conducted qualitative experiments in the next section to visualize the feature representations learned by WeedCLR. These experiments showed that WeedCLR learned discriminative features that can effectively separate different plant species.

4.3. Qualitative results

In this section, we present qualitative results from our experiments with WeedCLR. These results provide a visual representation of the effectiveness of our approach in learning semantically meaningful classes without the use of labels. Figs. 9 and 10 show sample images of classes predicted with high accuracy by our WeedCLR method on the DeepWeeds (Olsen et al., 2019) and CottonWeedID15 (Chen et al., 2022) validation sets, respectively. The images shown are randomly selected from each predicted class. The variety of backgrounds and weeds in the predicted classes demonstrates the ability of our WeedCLR method to learn high-quality representations for plant classification.

5. Ablation study

This ablation study investigates the effect of different factors on the top-1 accuracy of KNN and linear classifiers on the DeepWeeds and

CottonWeedID datasets. The factors considered are the loss function generality, batch size, softmax temperature, and MLP architecture.

5.1. Loss function generality

The loss function generality refers to the ability of the loss function to handle imbalanced datasets. The baseline model uses the cross-entropy loss function, which is not as robust to imbalanced datasets as other loss functions. The VNE is optimizing the representation space that can achieve a more desirable representation that avoids dimensional collapse and produces more useful embeddings. The COL loss function is our proposed variant of the cross-entropy loss function that is more robust to imbalanced datasets.

The results shown in Table 5 demonstrate that the VNE and COL loss functions consistently and significantly improve the top-1 accuracy of the KNN and linear classifiers on both datasets. The (VNE + COL) loss function achieves the best results, with a top-1 accuracy of 91.21% on the DeepWeeds dataset and 83.84% on the CottonWeedID dataset.

5.2. Batch size

The term ‘batch size’ denotes the quantity of samples that are processed during each iteration of the training algorithm. The baseline batch size is 32. The other batch sizes are 64, 128, and 256. The results shown in Table 6 demonstrate that the accuracy of the top-1 of the KNN and linear classifiers increases with the batch size. It is clear that

Table 1
k-nearest neighbors classifier evaluation on DeepWeeds (Olsen et al., 2019).

Method	ResNet-9		ResNet-18		ResNet-50	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
Supervised (Olsen et al., 2019)	97.51	99.92	97.64	99.93	97.34	99.92
BYOL (Grill et al., 2020)	64.66	98.69	69.94	99.03	69.08	98.50
BarlowTwins (Zbontar et al., 2021)	75.76	99.46	75.39	99.34	74.72	98.41
DCL (Yeh et al., 2022)	84.80	99.80	87.02	99.89	86.39	99.04
DCLW (Yeh et al., 2022)	84.77	99.77	86.42	99.77	85.73	98.78
DINO (Caron et al., 2021)	71.33	99.23	78.58	99.49	77.89	98.68
FastSiam (Pototzky et al., 2022)	83.95	99.71	83.30	98.79	82.78	97.79
Moco (Chen et al., 2020b)	83.49	99.74	84.57	99.74	83.63	99.03
NNCLR (Dwibedi et al., 2021)	85.95	99.71	85.13	99.14	84.36	98.28
SMoG (Pang et al., 2022)	82.95	99.71	82.31	98.82	81.38	98.06
SimCLR (Chen et al., 2020a)	74.33	99.46	78.64	99.60	77.80	99.00
SimSiam (Xie et al., 2022)	81.95	99.71	81.03	98.83	80.36	98.32
SwaV (Caron et al., 2020)	81.66	99.80	83.37	99.74	82.49	98.82
TiCo (Zhu et al., 2022)	73.25	99.06	76.95	99.34	76.45	98.52
WeedCLR (ours)	92.21	99.86	91.93	99.77	91.21	99.76

Table 2
Linear probe evaluation on DeepWeeds (Olsen et al., 2019).

Method	ResNet-9		ResNet-18		ResNet-50	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
Supervised (Olsen et al., 2019)	94.17	99.73	94.84	99.83	95.70	99.33
BYOL (Grill et al., 2020)	65.32	98.15	67.17	98.60	66.59	97.97
BarlowTwins (Zbontar et al., 2021)	76.61	98.92	74.81	99.00	73.89	98.03
DCL (Yeh et al., 2022)	80.41	98.86	79.89	98.49	78.89	97.79
DCLW (Yeh et al., 2022)	77.78	98.89	81.72	99.26	81.11	98.38
DINO (Caron et al., 2021)	63.41	98.60	75.07	98.83	74.14	98.29
FastSiam (Pototzky et al., 2022)	74.20	98.85	73.58	98.19	72.91	97.23
Moco (Chen et al., 2020b)	74.67	98.80	76.70	99.14	76.01	98.43
NNCLR (Dwibedi et al., 2021)	73.20	98.42	72.69	97.71	71.93	97.03
SMoG (Pang et al., 2022)	70.20	98.55	69.34	97.65	68.38	97.05
SimCLR (Chen et al., 2020a)	70.56	98.97	76.70	99.29	76.03	98.46
SimSiam (Xie et al., 2022)	71.20	98.67	70.22	98.17	69.32	97.32
SwaV (Caron et al., 2020)	80.26	99.26	80.98	99.17	80.38	98.56
TiCo (Zhu et al., 2022)	72.65	98.92	74.33	99.03	73.51	98.06
WeedCLR (ours)	85.97	99.34	87.22	99.49	86.63	99.33

Table 3
k-nearest neighbors classifier evaluation on CottonWeedID (Chen et al., 2022).

Method	ResNet-9		ResNet-18		ResNet-50	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
Supervised (Chen et al., 2022)	96.14	99.15	97.32	99.68	98.01	99.21
BYOL (Grill et al., 2020)	39.98	89.07	40.94	90.12	40.05	89.55
BarlowTwins (Zbontar et al., 2021)	45.93	91.37	42.19	90.03	41.57	89.31
DCL (Yeh et al., 2022)	76.89	97.32	79.67	97.03	78.88	96.27
DCLW (Yeh et al., 2022)	76.51	97.60	77.66	96.84	76.98	96.20
DINO (Caron et al., 2021)	37.87	84.18	41.51	87.15	40.78	86.50
FastSiam (Pototzky et al., 2022)	53.95	96.71	53.15	95.96	52.31	95.26
Moco (Chen et al., 2020b)	69.51	96.36	72.29	96.93	71.54	96.02
NNCLR (Dwibedi et al., 2021)	55.95	92.71	55.10	91.97	54.36	91.33
SMoG (Pang et al., 2022)	62.95	91.71	62.20	90.81	61.23	90.19
SimCLR (Chen et al., 2020a)	56.47	94.25	63.18	96.07	62.35	95.09
SimSiam (Xie et al., 2022)	61.95	92.71	61.30	91.82	60.64	90.89
SwaV (Caron et al., 2020)	58.29	95.01	65.77	95.88	65.20	95.04
TiCo (Zhu et al., 2022)	49.95	92.71	49.95	91.18	49.21	90.21
WeedCLR (ours)	86.00	98.18	84.66	98.47	83.84	97.52

the most significant improvement in top-1 accuracy for both KNN and linear classifiers occurs when the batch size is increased from 128 to 256. This is true for both the DeepWeeds and CottonWeedID datasets.

This substantial improvement could be attributed to the fact that a larger batch size allows the model to estimate the gradient more accurately during training. However, it is also important to note that using a larger batch size requires more memory, which might not always be feasible depending on the hardware constraints.

In conclusion, while our method shows robust performance across different batch sizes, using a larger batch size of 256 leads to the best

performance in terms of top-1 accuracy. Future work could explore the impact of even larger batch sizes on performance, given sufficient computational resources.

5.3. Softmax temperature

The softmax temperature is a hyperparameter that controls the degree of confidence in the predicted probabilities assigned to each class. A higher temperature results in a softer probability distribution, where the predicted probabilities are more evenly distributed among

Table 4
Linear probe evaluation on CottonWeedID (Chen et al., 2022).

Method	ResNet-9		ResNet-18		ResNet-50	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
Supervised (Chen et al., 2022)	95.32	99.15	96.31	99.68	98.14	99.34
BYOL (Grill et al., 2020)	49.57	90.41	45.06	90.22	44.46	89.53
BarlowTwins (Zbontar et al., 2021)	25.41	51.87	15.15	46.02	14.22	45.13
DCL (Yeh et al., 2022)	72.77	94.97	74.22	95.12	74.03	94.33
DCLW (Yeh et al., 2022)	74.02	96.07	73.06	95.45	72.35	94.26
DINO (Caron et al., 2021)	39.02	83.89	43.43	86.67	42.45	85.78
FastSiam (Pototzky et al., 2022)	63.20	92.14	62.26	91.46	61.72	90.90
Moco (Chen et al., 2020b)	57.81	92.04	61.27	93.48	60.42	92.51
NNCLR (Dwibedi et al., 2021)	53.20	91.14	52.24	90.42	51.58	89.60
SMoG (Pang et al., 2022)	61.20	91.14	60.37	90.52	59.44	89.96
SimCLR (Chen et al., 2020a)	57.33	92.23	63.95	94.82	63.25	93.94
SimSiam (Xie et al., 2022)	58.20	93.14	57.24	92.32	56.47	91.44
SwaV (Caron et al., 2020)	59.44	93.19	66.25	94.63	65.30	94.10
TiCo (Zhu et al., 2022)	51.20	92.14	48.42	90.80	47.47	90.23
WeedCLR (ours)	75.07	95.11	75.55	95.59	75.04	94.64

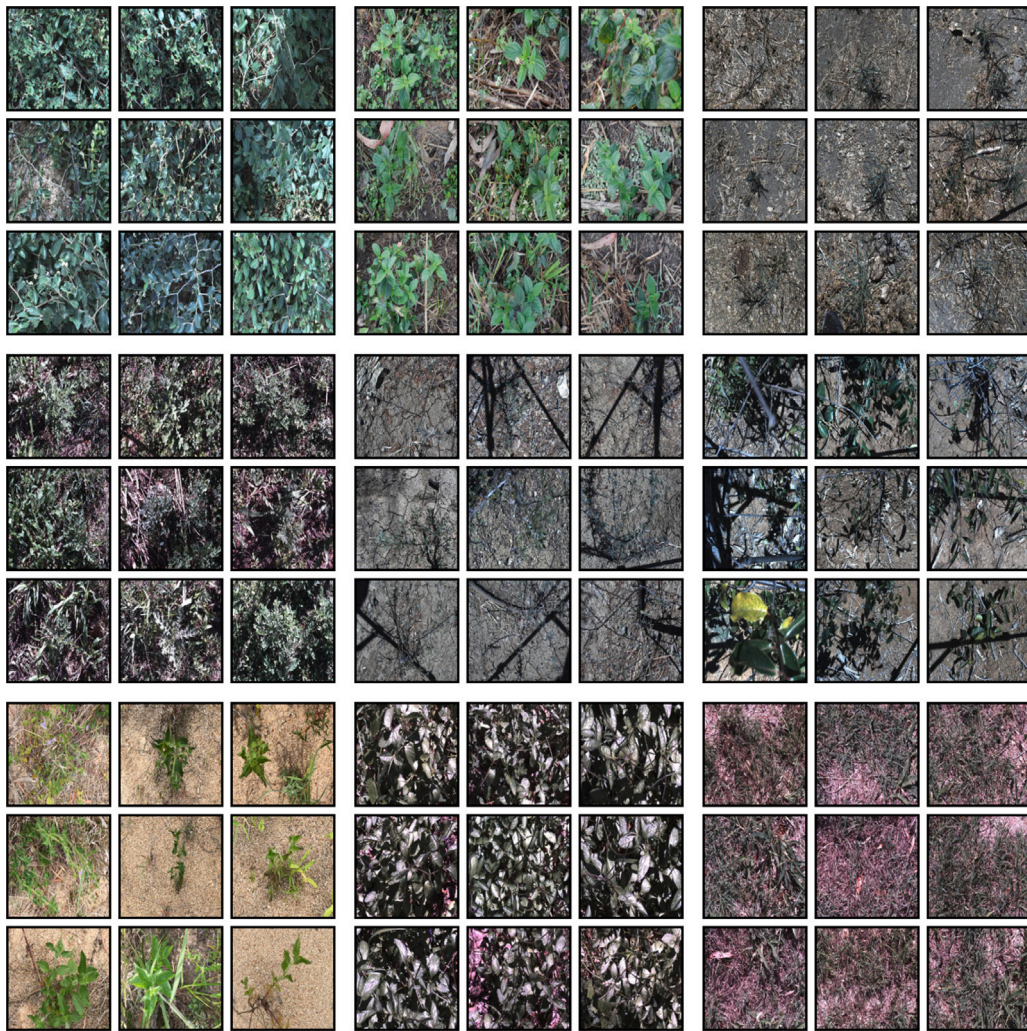


Fig. 9. Classes predicted with high accuracy by our *WeedCLR* on the DeepWeeds (Olsen et al., 2019) validation set, which was not seen during training. The images shown are randomly selected from each predicted class, namely from the top left: Chinese apple, Lantana, Parkinsonia, Parthenium, Prickly acacia, Rubber vine, Siam weed, Snake weed and Negatives. The variety of backgrounds and weeds in the predicted classes demonstrates that the *WeedCLR* is able to learn semantically meaningful classes without the use of labels.

the classes, while a lower temperature results in a sharper probability distribution, where the predicted probabilities are more concentrated on the most likely class.

The results shown in Table 7 demonstrate that the top-1 accuracy of the KNN classifier increases with the softmax temperature. The most

significant improvement in top-1 accuracy for the KNN classifier occurs when the softmax temperature is increased from 0.03 to 0.05. This is true for both the DeepWeeds and CottonWeedID datasets. These results suggest that a higher softmax temperature can lead to improved performance of the KNN classifier in this context.

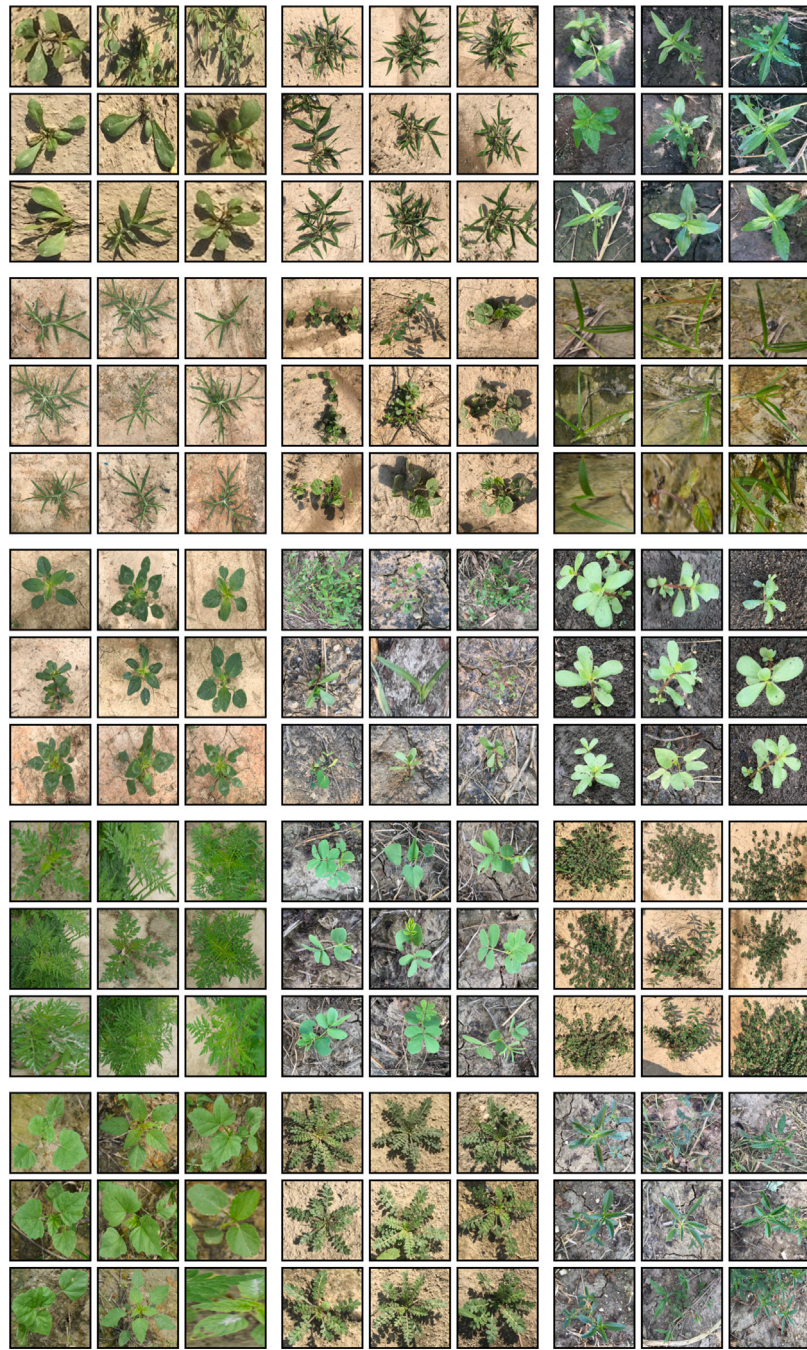


Fig. 10. Classes predicted with high accuracy by our *WeedCLR* on the CottonWeedID15 (Chen et al., 2022) validation set. The images shown are randomly selected from each predicted class, namely from the top left: Carpetweeds, Crabgrass, Eclipta, Goosegrass, Morningglory, Nutsedge, PalmerAmaranth, Prickly Sida, Purslane, Ragweed, Sicklepod, SpottedSpurge, SpurredAnoda, Swinecress, Waterhemp.

Table 5
Loss function generality - Top-1 accuracy.

Parameter	DeepWeed		CottonWeedID	
	KNN	Linear	KNN	Linear
Baseline	54.35	55.13	45.78	45.42
Baseline + COL	84.76	82.84	72.63	67.89
Baseline + VNE	87.68	84.32	79.51	71.48
Baseline + COL + VNE	91.21	86.63	83.84	75.04

Table 6
Batch size - Top-1 accuracy.

Batch size	DeepWeed		CottonWeedID	
	KNN	Linear	KNN	Linear
32	82.24	77.31	76.82	70.34
64	85.17	79.48	77.92	70.27
128	87.34	80.27	79.61	72.98
256	91.21	86.63	83.84	75.04

Table 7
Softmax temperature — KNN classifier Top-1 accuracy.

τ_{column}	τ_{row}			
	DeepWeed		CottonWeedID	
	0.07	0.1	0.07	0.1
0.03	90.78	91.04	82.45	82.78
0.05	91.08	91.21	83.25	83.84

Table 8
MLP architecture — KNN classifier Top-1 accuracy.

MLP hidden layer(s)	MLP output layer			
	DeepWeed		CottonWeedID	
	128	256	128	256
1 × 4096	90.78	89.01	82.27	81.65
2 × 4096	91.21	90.34	83.84	82.61
2 × 8192	90.32	88.98	82.78	81.64

5.4. MLP architecture

The architecture of a Multilayer Perceptron (MLP) is defined by the number of hidden layers it has and the number of units present in each of these hidden layers. The baseline MLP architecture has one hidden layer with 4096 units. The other MLP architectures have two hidden layers with 4096 units each and two hidden layers with 8192 units each.

The results shown in Table 8 demonstrate that the architecture with two hidden layers of 4096 units each provides slightly higher top-1 accuracy for the KNN classifier compared to the baseline architecture (one hidden layer with 4096 units). This suggests that adding an additional layer might help the model capture more complex patterns in the data, leading to improved performance.

However, it is important to note that increasing the size of the hidden layers (from 4096 to 8192 units) in a two-layer architecture does not lead to further improvements. This could be due to overfitting, where the model becomes too complex and starts to fit the noise in the training data rather than the underlying patterns. In our results presented above, we have used the architecture with two hidden layers of 4096 units each, as it provided the best performance in the ablation study.

6. Discussion

In this paper, we presented a novel method for weed classification in long-tailed datasets using a self-visual features learning approach. Our method, called Weed Contrastive Learning through visual Representation (WeedCLR), utilizes a class-optimized loss function to improve classification accuracy. We demonstrated the effectiveness of our approach on two datasets of weed images, achieving state-of-the-art performance.

Our results show that the proposed method is able to effectively learn discriminative visual features for weed classification without the use of any annotations even in the presence of long-tailed data distributions. This has significant implications for the development of automated weed management systems (Arsa et al., 2023; Dang et al., 2023). By removing the requirement of human labeling, this self-supervised approach could reduce the time and cost burden of existing fully supervised approaches. For example, Calvert et al. (2021) reports a time requirement of one hour to label 2000 images for weed classification. For their 58,153 image dataset, approximately 29 h of labeling by domain experts and the equivalent labor cost could be avoided, while achieving a similar or better classification accuracy. This increased efficiency also has a second advantage allowing for deep learning based approaches to be more rapidly deployed on-farm when targeting a new weed or crop scenario. Currently, the time it takes from

dataset collection to field implementation is strongly influenced by the time to annotate images. With the removal of this annotation time, the window from dataset collection to implementation is shortened allowing for more timely and effective weed control on new target weed crop scenarios. As such, WeedCLR can help to overcome a major hurdle of data annotation for widespread adoption of deep learning based site specific weed management approaches.

One limitation of our approach is that it relies on the availability of large and diverse datasets for training. While our method is effective in classifying weeds in long-tailed datasets, its performance may be limited by the size and diversity of the available data. In future work, it would be interesting to explore methods for data augmentation or transfer learning to improve the performance of our approach in scenarios where data is limited.

Another limitation of our approach is that it is currently designed for weed classification in long-tailed datasets. While our method shows promising results in this specific application, it remains to be seen how well it generalizes to other classification tasks with different data distributions. In future work, it would be interesting to evaluate the performance of our approach on other classification tasks and explore methods for adapting it to different data distributions.

Another potential direction for future work is to explore the integration of our method with other components of an automated weed management system, such as robotic platforms for weed management that our team has previously developed (Calvert et al., 2021). Additionally, further research could investigate the use of our method in other domains such as medical imaging where long-tailed data distributions are common.

7. Conclusions

This paper introduced Weed Contrastive Learning through visual Representation (WeedCLR), a novel method for weed classification in long-tailed datasets, which leverages self-supervised learning to extract meaningful visual features from images of weeds. The utilization of a class-optimized loss function with Von Neumann Entropy of deep representation significantly improved classification accuracy.

The findings from this study have advanced the current state-of-the-art by demonstrating WeedCLR's ability to discern visual characteristics crucial for weed identification, even in the presence of long-tailed data distributions and without the requirement for laborious human annotation of images. This has substantial implications for the field of automated weed control systems, as it unlocks self-supervised learning for weed recognition, reducing reliance on manual labor and increasing the speed of implementation in new weed and crop scenarios for site specific weed management.

The current findings suggest potential new research directions in improving the accuracy and efficiency of weed classification tasks under varying environmental conditions. They also open up possibilities for integrating WeedCLR into existing automated weed control systems and exploring its effectiveness in different agricultural contexts. This study contributes to the body of knowledge by providing a promising new method for weed classification in long-tailed datasets, thereby paving the way for more effective and efficient weed management systems.

7.1. CO2 emission related to experiments

Experiments were conducted using a private infrastructure, which has a carbon efficiency of 0.432 kgCO₂eq/kWh. A cumulative of 850 h of computation was performed on hardware of type RTX 2080 Ti (TDP of 250 W). Total emissions are estimated to be 91.8 kgCO₂eq of which 0 percent was directly offset. Estimations were conducted using the Machine Learning Impact calculator presented in Lacoste et al. (2019).

CRedit authorship contribution statement

Alzayat Saleh: Conceptualisation, Data curation, Data analysis, Software development, DL algorithm design, Visualization, Writing – original draft. **Alex Olsen:** Data curation, Reviewing/editing the draft. **Jake Wood:** Reviewing/editing the draft. **Bronson Philippa:** Reviewing/editing the draft. **Mostafa Rahimi Azghadi:** Conceptualisation, Data curation, Data analysis, Reviewing/editing the draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research is funded by the partnership between the Australian Government's Reef Trust and the Great Barrier Reef Foundation, Australia.

Data availability

Data will be made available on request.

References

- Arora, S., Cohen, N., Hu, W., Luo, Y., 2019a. Implicit regularization in deep matrix factorization. In: *NeurIPS*.
- Arora, S., Khandeparkar, H., Khodak, M., Plevrakis, O., Saunshi, N., 2019b. A theoretical analysis of contrastive unsupervised representation learning. In: *ICML*.
- Arsa, D.M.S., Ilyas, T., Park, S.H., Won, O., Kim, H., 2023. Eco-friendly weeding through precise detection of growing points via efficient multi-branch convolutional neural networks. *Comput. Electron. Agric.* 209, 107830. <http://dx.doi.org/10.1016/J.COMPAG.2023.107830>.
- Barrett, D., Dherin, B., 2021. Implicit gradient regularization. *ArXiv abs/2009.11162*.
- Calvert, B., Olsen, A., Whinney, J., Rahimi Azghadi, M., 2021. Robotic spot spraying of harrisia cactus (*Harrisia martinii*) in grazing pastures of the Australian Rangelands. *Plants* 10 (10), 2054. <http://dx.doi.org/10.3390/plants10102054>, URL: <https://www.mdpi.com/2223-7747/10/10/2054>.
- Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A., 2020. Unsupervised learning of visual features by contrasting cluster assignments. In: *Advances in Neural Information Processing Systems*. Vol. 2020-December.
- Caron, M., Touvron, H., Misra, I., Jegou, H., Mairal, J., Bojanowski, P., Joulin, A., 2021. Emerging properties in self-supervised vision transformers. In: *Proceedings of the IEEE International Conference on Computer Vision*. <http://dx.doi.org/10.1109/ICCV48922.2021.00951>.
- Chen, X., Fan, H., Girshick, R., He, K., 2020b. Improved baselines with momentum contrastive learning. URL: <https://arxiv.org/abs/2003.04297v1>.
- Chen, T., Kornblith, S., Norouzi, M., Hinton, G., 2020a. A simple framework for contrastive learning of visual representations. In: *International Conference on Machine Learning*. PMLR, pp. 1597–1607.
- Chen, D., Lu, Y., Li, Z., Young, S., 2022. Performance evaluation of deep transfer learning on multi-class identification of common weed species in cotton production systems. *Comput. Electron. Agric.* 198, 107091. <http://dx.doi.org/10.1016/j.compag.2022.107091>, URL: <https://linkinghub.elsevier.com/retrieve/pii/S0168169922004082>.
- Coleman, G.R., Bender, A., Hu, K., Sharpe, S.M., Schumann, A.W., Wang, Z., Bagavathiannan, M.V., Boyd, N.S., Walsh, M.J., 2022. Weed detection to weed recognition: reviewing 50 years of research to identify constraints and opportunities for large-scale cropping systems. *Weed Technol.* 36 (6), 741–757. <http://dx.doi.org/10.1017/WET.2022.84>, URL: <https://www.cambridge.org/core/journals/weed-technology/article/weed-detection-to-weed-recognition-reviewing-50-years-of-research-to-identify-constraints-and-opportunities-for-largescale-cropping-systems/2616454D5EF61EE036C7DFC5F9181E26>.
- Dang, F., Chen, D., Lu, Y., Li, Z., 2023. YOLOWeeds: A novel benchmark of YOLO object detectors for multi-class weed detection in cotton production systems. *Comput. Electron. Agric.* 205, 107655. <http://dx.doi.org/10.1016/J.COMPAG.2023.107655>.
- Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., Zisserman, A., 2021. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In: *Proceedings of the IEEE International Conference on Computer Vision*. <http://dx.doi.org/10.1109/ICCV48922.2021.00945>.
- Dyrmann, M., Midtby, H.S., Jørgensen, R.N., et al., 2016. Evaluation of intra variability between annotators of weed species in color images. In: *4th CIGR International Conference of Agricultural Engineering*. Aarhus, Denmark: CIGR, pp. 1–6.
- Espejo-Garcia, B., Panoutsopoulos, H., Anastasiou, E., Rodríguez-Rigueiroz, F.J., Fountas, S., 2023. Top-tuning on transformers and data augmentation transferring for boosting the performance of weed identification. *Comput. Electron. Agric.* 211, 108055. <http://dx.doi.org/10.1016/J.COMPAG.2023.108055>.
- Grill, J.-B., Strub, F., Altché, F., Tallec, C., H. Richemond, P., Buchatskaya, E., Doersch, C., Pires, B.A., Guo, Z.D., Azar, M.G., Piot, B., Kavukcuoglu, K., Munos, R., Valko, M., 2020. Bootstrap your own latent: A new approach to self-supervised learning. In: *Advances in Neural Information Processing Systems*.
- Gunasekar, S., Woodworth, B.E., Bhojanapalli, S., Neyshabur, B., Srebro, N., 2018. Implicit regularization in matrix factorization. In: *2018 Information Theory and Applications Workshop*. ITA, pp. 1–10.
- Ji, Z., Telgarsky, M., 2019. Gradient descent aligns the layers of deep linear networks. *ArXiv abs/1810.02032*.
- Kim, J., Kang, S., Hwang, D., Shin, J., Rhee, W., 2023. VNE: An effective method for improving deep representation by manipulating eigenvalue distribution. URL: <https://arxiv.org/abs/2304.01434v1>.
- Lacoste, A., Luccioni, A., Schmidt, V., Dandres, T., 2019. Quantifying the carbon emissions of machine learning. URL: [/green-ai/publications/2019-11-lacoste-quantifying.html](https://green-ai/publications/2019-11-lacoste-quantifying.html).
- Lee, J., Lei, Q., Saunshi, N., Zhuo, J., 2020. Predicting what you already know helps: Provable self-supervised learning. *ArXiv abs/2008.01064*.
- Magistri, F., Weyler, J., Gogoll, D., Lottes, P., Behley, J., Petrinic, N., Stachniss, C., 2023. From one field to another—Unsupervised domain adaptation for semantic segmentation in agricultural robotics. *Comput. Electron. Agric.* 212, 108114. <http://dx.doi.org/10.1016/J.COMPAG.2023.108114>.
- Neyshabur, B., Li, Z., Bhojanapalli, S., LeCun, Y., Srebro, N., 2019. Towards understanding the role of over-parametrization in generalization of neural networks. *ArXiv abs/1805.12076*.
- Olsen, A., Kononov, D.A., Philippa, B., Ridd, P., Wood, J.C., Johns, J., Banks, W., Girgenti, B., Kenny, O., Whinney, J., Calvert, B., Azghadi, M.R., White, R.D., 2019. DeepWeeds: A multiclass weed species image dataset for deep learning. *Sci. Rep.* 9 (1), 2058. <http://dx.doi.org/10.1038/s41598-018-38343-3>, URL: <https://www.nature.com/articles/s41598-018-38343-3>.
- Oord, A.v.d., Li, Y., Vinyals, O., 2018. Representation learning with contrastive predictive coding. URL: <https://arxiv.org/abs/1807.03748v2>.
- Pang, B., Zhang, Y., Li, Y., Cai, J., Lu, C., 2022. Unsupervised visual representation learning by synchronous momentum grouping. In: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 13690 LNCS, http://dx.doi.org/10.1007/978-3-031-20056-4_16.
- Pototzky, D., Sultan, A., Schmidt-Thieme, L., 2022. FastSiam: Resource-efficient self-supervised learning on a single GPU. In: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 13485 LNCS, http://dx.doi.org/10.1007/978-3-031-16788-1_4.
- Rai, N., Zhang, Y., Ram, B.G., Schumacher, L., Yellavajjala, R.K., Bajwa, S., Sun, X., 2023. Applications of deep learning in precision weed management: A review. *Comput. Electron. Agric.* 206, 107698. <http://dx.doi.org/10.1016/J.COMPAG.2023.107698>.
- Saxe, A.M., McClelland, J.L., Ganguli, S., 2019. A mathematical theory of semantic development in deep neural networks. *Proc. Natl. Acad. Sci.* 116, 11537–11546.
- Tian, Y., Chen, X., Ganguli, S., 2021. Understanding self-supervised learning dynamics without contrastive pairs. *ArXiv abs/2102.06810*.
- Tosh, C., Krishnamurthy, A., Hsu, D.J., 2021. Contrastive learning, multi-view redundancy, and linear models. *ArXiv abs/2008.10150*.
- Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., Hu, H., 2022. SimMIM: a simple framework for masked image modeling. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 2022-June, <http://dx.doi.org/10.1109/CVPR52688.2022.00943>.
- Xu, K., Shu, L., Xie, Q., Song, M., Zhu, Y., Cao, W., Ni, J., 2023. Precision weed detection in wheat fields for agriculture 4.0: A survey of enabling technologies, methods, and research challenges. *Comput. Electron. Agric.* 212, 108106. <http://dx.doi.org/10.1016/J.COMPAG.2023.108106>.
- Yeh, C.H., Hong, C.Y., Hsu, Y.C., Liu, T.L., Chen, Y., LeCun, Y., 2022. Decoupled contrastive learning. In: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 13686 LNCS, http://dx.doi.org/10.1007/978-3-031-19809-0_38.
- You, Y., Gitman, I., Ginsburg, B., 2017. Large batch training of convolutional networks. *Arxiv*.
- Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S., 2021. Barlow twins: Self-supervised learning via redundancy reduction. In: *Proceedings of Machine Learning Research*. Vol. 139.
- Zhu, J., Moraes, R.M., Karakulak, S., Sobol, V., Canziani, A., LeCun, Y., 2022. TiCo: Transformation invariance and covariance contrast for self-supervised visual representation learning. URL: <https://arxiv.org/abs/2206.10698v2>.