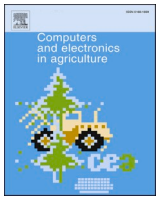
Computers and Electronics in Agriculture 191 (2021) 106510

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag



Self-supervised contrastive learning on agricultural images

Ronja Güldenring *, Lazaros Nalpantidis

Department of Electrical Engineering, Elektrovej, Lyngby 2800, Denmark

ARTICLE INFO

Keywords:
Contrastive learning
Deep learning
Self-supervision
SwAV
Transfer-learning

MSC:
68T45
68U10

ABSTRACT

Agriculture emerges as a prominent application domain for advanced computer vision algorithms. As much as deep learning approaches can help solve problems such as plant detection, they rely on the availability of large amounts of annotated images for training. However, relevant agricultural datasets are scarce and at the same time, generic well-established image datasets such as ImageNet do not necessarily capture the characteristics of agricultural environments. This observation has motivated us to explore the applicability of self-supervised contrastive learning on agricultural images. Our approach considers numerous non-annotated agricultural images, which are easy to obtain, and uses them to pre-train deep neural networks. We then require only a limited number of annotated images to fine-tune those networks in a supervised training manner for relevant downstream tasks, such as plant classification or segmentation. To the best of our knowledge, contrastive self-supervised learning has not been explored before in the area of agricultural images. Our results reveal that it outperforms conventional deep learning approaches in classification downstream tasks, especially for small amounts of available annotated training images where up to 14% increase of average top-1 classification accuracy has been observed. Furthermore, the computational cost for generating data-specific pre-trained weights is fairly low, allowing one to generate easily new pre-trained weights for any custom model architecture or task.

1. Introduction

In agriculture, there are only a few publicly available datasets that are large enough to be suitable for state-of-the-art deep learning approaches, as such approaches generally require large amounts of training images. Nevertheless, the required dataset size is proportional to the number of plant species considered in it. Datasets such as DeepWeeds (Olsen et al., 2019) or Sugar Beets 2016 (Chebrolu et al., 2017) are common choices for evaluating newly proposed computer vision approaches in the domain of agriculture, but they only consider a relatively small number of species. Yet, from a practical point of view, the introduction of vision-equipped robots in different precision agriculture tasks requires consideration of many different plant species. As is the trend with other domains, such as e.g. medical imaging (Azizi et al., 2021), where annotated training examples are difficult to obtain, label-efficiency gains stronger attention (Argüeso et al., 2020; Güldenring et al., 2021; Madsen et al., 2019; Skovsen et al., 2019). Label efficient training requires fewer manually annotated training examples, allowing one to easily integrate new plant species for custom robot applications.

One way to achieve label-efficiency is semi-supervised training with self-supervision. The key idea of self-supervised pre-training procedures is to generate automatic labels for unlabeled data; then, a deep neural network can be trained in a supervised manner with these labels. Although the formulated task in self-supervised pre-training is generally not meaningful, the network still learns relevant features of the given data. In a second step, the obtained network weights can be transferred to relevant downstream tasks, such as object detection or segmentation. Recently, self-supervised training using contrastive learning received strong attention within the domain of Deep Learning in Computer Vision. Contrastive approaches such as SimCLR (Chen et al., 2020) and SwAV (Caron et al., 2020) achieved comparable results to the well-known pre-trained ImageNet weights, even if the latter are obtained in a fully supervised manner.

In this work we explore the applicability of contrastive learning in the area of agriculture, motivated by the fact that well-established and widely-used ImageNet weights are obtained with image data that differs semantically as well as in texture and color from agricultural images. To the best of our knowledge, contrastive learning has not yet been explored in the area of agricultural images.

More specifically, we compare supervised pre-training from ImageNet with self-supervised training on agriculture-specific data. We use SwAV to obtain pre-trained weights for three different datasets: (1)

* Corresponding author.
E-mail address: ronjag@elektro.dtu.dk (R. Güldenring).

https://doi.org/10.1016/j.compag.2021.106510
Received 13 September 2021; Received in revised form 15 October 2021; Accepted 17 October 2021
Available online 6 November 2021

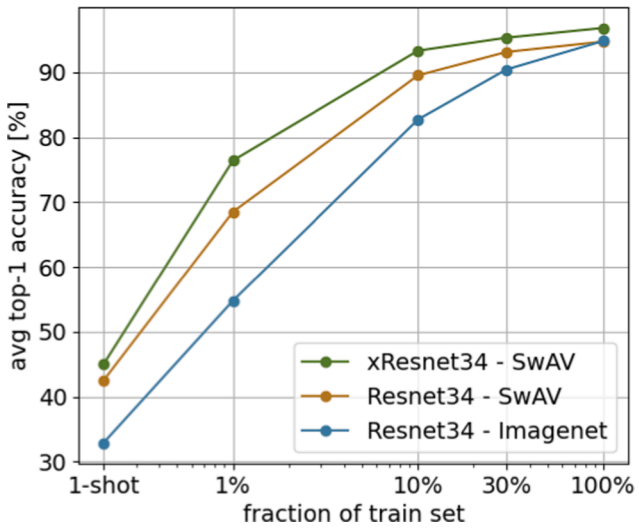


Fig. 1. Average top-1 classification performance for models trained on different fractions of the DeepWeeds dataset. We apply transfer learning and compare supervised ImageNet training to self-supervised pre-training using SwAV. With an increase of average top-1 accuracy up to 14%, SwAV is particularly efficient in the semi-supervised settings, where only a fraction of image annotations is needed. Furthermore, SwAV allows to easily obtain pre-trained weights for any network structure. Exemplary, we show that minimal network modifications such as concat pooling and xResNet lead to a further performance increase.

DeepWeeds (Olsen et al., 2019), (2) Aerial Farmland (Chiu et al., 2020), and (3) Grassland Europe. The latter is generated by scraping 24.000 grassland images from the web, which is a simple way of collecting plant image data. The dataset-specific pre-trained weights are compared to supervised ImageNet weights by transferring them to classification and segmentation downstream tasks on the different considered datasets. Fig. 1 shows the *average top-1 accuracy*, trained on different fractions of the DeepWeeds dataset, demonstrating the label-efficiency of self-supervised models. Furthermore, we find it beneficial that self-supervised training allows one to easily generate pre-trained weights for any network architecture. We demonstrate that minimal network modifications increase the top-1 accuracy even further, whereas pre-trained ImageNet weights are not available for custom network designs.

Our key findings can be summarized as in the following:

- Self-supervised models predominantly improve the classification performance compared to supervised models. They are especially effective in semi-supervised settings, where we performed fine-tuning on fractions of the datasets.
- Inspired by Azizi et al. (2021), we show that initializing the self-supervised model with supervised ImageNet pre-trained weights can lead to further minimal improvement, but also speeds up the convergence of the self-training.
- The computational cost of obtaining data-specific pre-trained weights with SwAV is relatively low and allows one to quickly obtain weights for custom model architectures. Exemplary, we show that experimenting with minimal network modification – including concat pooling and xResnet modifications – can boost the classification performance significantly.
- In our experiments, we show that self-supervised models are capable of dealing successfully with a variety of downstream tasks, but are less effective in segmentation tasks.

The structure of this research work is as follows: Section 2 gives an overview of relevant literature for advances in Unsupervised Representation Learning and Representation Learning in the context of Agriculture. Sections 3 presents the methodology and architectural design choices in great detail. Section 4 presents the considered datasets for evaluation and defines the performed experiments, while these experiments are presented and discussed individually in Section 5. Finally, Section 6 concludes our work and points out potential future work.

2. Related Work

2.1. Unsupervised Representation Learning

The motivation behind *Unsupervised Representation Learning*, also known as *Self-supervised Learning*, is to learn feature representation with unlabeled data only. This process is referred to as *pre-training* and it is commonly implemented by formulating a so-called *pre-text task*: a supervised task formulation where labels can be obtained automatically. The pre-text task has no necessary relation to the final network usage, but indirectly forces the network to learn relevant low- and high-level features. Common practice is to further use the pre-trained feature extractor and fine-tune it in a *downstream task*, where the final task—such as classification, object detection or segmentation—will be learned in a supervised manner. Optimally, the network has already learned all relevant features during the pre-text task and only needs to learn the relation between high-level features and label names. Usually, self-supervised pre-training allows the downstream task to converge faster and improves label efficiency. (Jing and Tian, 2021).

A selection of methods for generating pre-text tasks is listed in the following:

- Transformation-based methods. They are based on the assumption that image transformations do not change the semantic content of an image. Therefore, an image along with all its possible image transformations can be treated as one class (Dosovitskiy et al., 2014; Gidaris et al., 2018).
- Patch-based methods. The task is to predict the relative position between patches originating from the same image (Doersch et al., 2015; Noroozi and Favaro, 2016).
- Generation-based methods. The task is to let the network generate information, that is already available and will be used as ground truth, such as image colorization (Zhang et al., 2016) or image patch reconstruction (Pathak et al., 2016).
- Temporal-based methods. The task is to predict the temporal order of images extracted from a video (Misra et al., 2016).
- Cluster-based methods. Feature vectors extracted during the forward pass of a network are further processed by cluster algorithms. The cluster assignments serve as pseudo labels for the loss computation and backward pass (Caron et al., 2018; Yang et al., 2016).

Very recently, self-supervised Learning using Noise Contrastive Estimation (NCE) (Gutmann and Hyvürinen, 2010; van den Oord et al., 2018) has gained attention, as it provides results comparable to state-of-the-art supervised methods trained on ImageNet (Deng et al., 2009). Instance discrimination methods apply the concept of contrastive learning to whole images. Common practice is to use image transformations to generate different image views, while image views originating from the same image form positive image pairs and those originating from different images form negative pairs. The objective of NCE is to push positive pairs close to one another and negative pairs further apart.

Google Research has proposed a relatively simple framework: SimCLR (Chen et al., 2020). SimCLR processes image batches through a simple encoder network, while for each image pair in the batch all other samples are treated as negative pairs. In their experiments, they show that contrastive learning benefits from large batch sizes (≈4096) and long training epochs. However, this requires extremely much computational capacity and is therefore not feasible for everyone.

In contrast to SimCLR, MoCo (He et al., 2020) from Facebook AI Research is more applicable for training on limited hardware. Instead of large batch sizes, they introduce a queue that keeps negative examples

Table 1
State-of-the-art self-supervised contrastive learning approaches with its key design choices.

Self-supervised Contrastive Learning Architecture	Key Design Choices
SimCLR (Chen et al., 2020)	• Simple Encoder Network. • InfoNCE loss (van den Oord et al., 2018), considering negative and positive pairs. • Extreme large batch sizes with many negative samples (≈4096).
MoCo (He et al., 2020)	• Encoder and momentum encoder network, which is a slow moving average of the encoder network and encodes the negative samples. • InfoNCE loss (van den Oord et al., abs/1807.03748 (2018).), considering negative and positive pairs. • Queue with negative samples from previous batches which allows smaller batch sizes.
BYOL (Grill et al., 2020)	• Online and target network, which is a slow moving average of the online network. Each network is predicting one view of an image. • Loss is the mean-square error of both network predictions. No contrasting towards negative samples, only positive pairs are considered.
SwAV (Caron et al., 2020)	• Encoder network followed by feature vector clustering. • Cross-entropy loss function on swapped predictions, considering negative and positive pairs. • SwAV works with small batch sizes and allows negative and positive examples to be similar. • Introduction of an effective data augmentation method *multi-crop*, which is also beneficial in the other approaches.
SimSiam (Chen and He, 2021)	• Encoder network 1 combined with a projection head (MLP) and encoder network 2 without projection head. Both encoder networks share weights. Only the first network gets updated. • Loss is the negative cosine similarity. No contrasting towards negative samples, only positive pairs are considered.

from previous batches. Furthermore, they introduce a momentum encoder for encoding negative examples that has the same architecture as the original encoder network. During training, only the encoder is updated while weights are copied to the momentum encoder with a certain time delay, i.e. the momentum encoder is behind the encoder.

While MoCo and SimCLR stress the importance of a large number of negative samples, BYOL (Grill et al., 2020) and SimSiam (Chen and He, 2021) show that state-of-the art performance can be achieved without providing any negative examples actively.

Caron et. al. proposes SwAV (Caron et al., 2020), which combines concepts of contrastive learning with cluster-based learning methods. The main advantage of SwAV is that it allows negative and positive examples to be similar, which can be easily the case when two different images are semantically similar. Furthermore, SwAV computes cluster assignments online and works with relatively small batch sizes. SwAV is explained in more details in Section 3.2 as it constitutes an integral part of our work presented in this paper.

Table 1 provides an overview of the most recent publications in the field of self-supervised contrastive learning and lists the key design choices of each approach.

All presented self-supervised contrastive learning approaches have been applied to the benchmark dataset ImageNet that includes more than 14 million images of diverse classes. But how well do those methods work in more specific domains with substantially smaller dataset sizes? Azizi et al. (2021) recently applied contrastive learning to the domain of medical images, considering classification datasets containing "only" ten to hundred thousands of images. They show, that classic supervised ImageNet transfer learning can be outperformed with domain-specific self-supervision. However, they achieve even better performance when combining pre-training both on ImageNet and medical images.

2.2. Agricultural Representation Learning

Although agricultural images provide significant characteristic differences compared to ImageNet data (Deng et al., 2009), it is a common practice to use ImageNet weights for transfer learning. A number of publications in the domain of agriculture, e.g. Kounalakis et al. (2019), Mehdipour Ghazi et al. (2017) and Suh et al. (2018), show that ImageNet pre-trained weights improve the model performance. However, little research has been performed on whether more domain-specific transfer learning could improve performance even further. Bargoti and Underwood (2017) transfer weights between fruit orchards, but benefits over ImageNet weights are only shown when training with very few images. Bosilj et al. (2020) train supervised on one specific crop and transfer these weights to fine-tune on another crop. They show that initializing networks with similar-domain weights leads to more efficient training and only requires a fraction of the ground truth labels to achieve similar performance. Espejo-Garcia et al. (2020) generate agricultural pre-trained weights on the highly diverse, but relatively small *Plant Seedlings Dataset* (Mehdipour Ghazi et al., 2017) and test them on the more specific *Early Crop Weeds Dataset* (Espejo-Garcia et al., 2020). They perform experiments for four popular CNN architectures and report an overall slight improvement in performance and reduction in training convergence time. Marin Zapata et al. (2020) learn a representation of a plant phenotyping dataset with a triplet network architecture, inspired by Wang et al. (2014). They take advantage of the underlying timing sequence of the images in their dataset, where consecutive image frames are treated as similar and non-consecutive image frames as dissimilar. The learned representation is used in various downstream tasks such as clustering and classification. Zhao et al. (2020) propose a multi-task learning framework that combines supervised classification with a hand-crafted self-supervision task, inspired by Gidaris et al. (2018). The purpose of the self-supervision component is to speed up the optimization by setting a focus on the encoding of image orientation. dos Santos Ferreira et al. (2019) applied two unsupervised deep clustering algorithms JULE (Yang et al., 2016) and DeepCluster (Caron et al., 2018) to two agricultural datasets. Unfortunately, they do not transfer the obtained weights, but use the trained clustering network to semi-automate the image annotation and re-train a classification model with ImageNet weight initialization.

In contrast to the works discussed above, we use self-supervised contrastive learning to obtain agriculture-specific pre-trained weights. Unsupervised learning is especially relevant in agriculture, because collecting images is relatively easy while their manual annotation requires a lot of additional effort. To the best of our knowledge, this is the first work that investigates self-supervised contrastive learning in the domain of agriculture.

3. Methods

3.1. xResNet architecture

In this work we will consider two backbone architectures: ResNet-34 and xResNet-34. While the default ResNet architectures are widely known, the xResNet introduces small modifications in the input stem and downsampling block proposed by He et al. (2019). Note, that the name *xResNet* was not originally given by its authors He et al. (2019), but was established by the FastAI community [1]. The (7×7) convolution in the input stem of the ResNet is replaced with three (3×3) convolutions. In the downsampling block, two modifications are applied: (1) in path A, the stride is applied in the second convolution instead of the first,

[1] https://docs.fast.ai/vision.models.xresnet.html

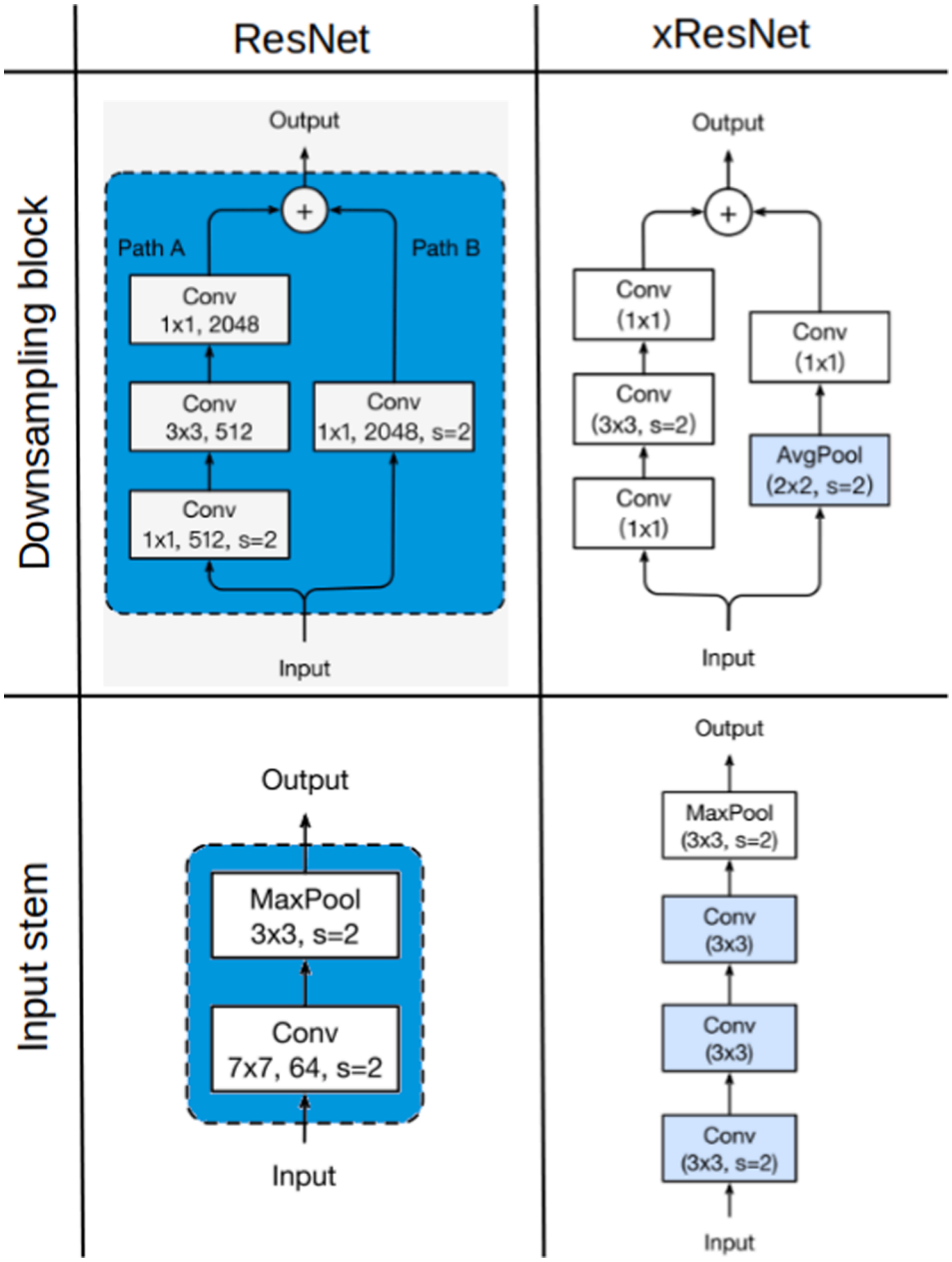


Fig. 2. The xResNet architecture proposed by He et al. (2019) (right) modifies the original ResNet (left) in the downsampling block and the input stem.

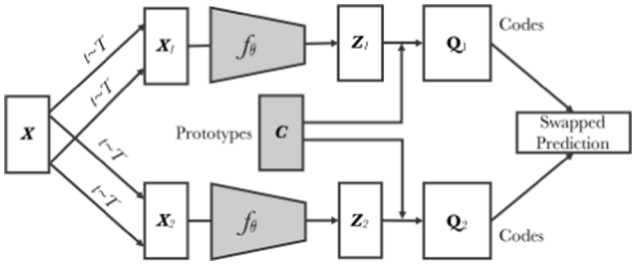


Fig. 3. Architecture of SwAV (Caron et al., 2020): An image batch X is augmented twice to obtain (X_1, X_2) which are then processed by the identical feature extractor f_θ. The obtained feature vectors (Z_1, Z_2) are mapped to prototype vectors that are further clustered to (Q_1, Q_2). The clusters serve as pseudo labels and are swapped between the two streams, so that the loss is computed between Z_1 and Q_2 and vice versa.

(2) in path B, a (2×2) average pooling with stride 2 precedes the convolutional layer, whose stride is reduced to 1. The modifications are illustrated in Fig. 2.

Such small modifications come with a significant cost: To apply transfer learning, new pre-trained weights must be obtained, as the ones for ResNet are not applicable. This is usually done with ImageNet (Deng et al., 2009), which consists of ≈14 million images, resulting in a computational expensive training. In this work, we do not have ImageNet pre-trained weights available because of computational limitations. Instead, we will obtain weights using SwAV (Caron et al., 2020) for self-supervised pre-training.

3.2. SwAV

The work of Caron et al. (2020) introduced SwAV, which combines cluster-based learning approaches with contrastive learning. Fig. 3 illustrated the architecture of SwAV, described in the following. A batch X of images $x_1, \ldots, x_n$ is augmented twice to obtain batches X_1, X_2. Both

augmented batches are processed by the same encoder network to obtain corresponding feature vectors Z_1, Z_2. These features are assigned to clusters Q_1, Q_2 by mapping Z_1, Z_2 to k learnable prototypes, which are further processed by the Sinkhorn Knopp algorithm (Cuturi, 2013). For the computation of the loss, as shown in Eqs. (1)–(3), let's assume one image x_i from X with two augmented views x_{i1}, x_{i2}. The obtained cluster assignments q_{i1}, q_{i2} serve as labels for the images. The trick hereby is to swap the assigned clusters between the image views, i.e. q_{i1} serves as target for x_{i2} and vice versa (Eq. 1). Concretely, the loss function is a cross-entropy loss of probability of the predictions z_{i1}, z_{i2} and the swapped targets q_{i2}, q_{i1} (Eq. 2). The probability of the predictions is computed by taking the Softmax of the vector similarity between the feature vector z_i and the prototype vectors C (Eq. 3).

$$L(z_{i1}, z_{i2}) = l(z_{i1}, q_{i2}) + l(z_{i2}, q_{i1}) \tag{1}$$

$$l(z_i, q_i) = -\sum_k q_i^{(k)} \log p_i^{(k)} \tag{2}$$

$$\text{where} \quad p_i^{(k)} = \frac{\exp(z_i^T c_k / \tau)}{\sum_{k'} \exp(z_i^T c_{k'} / \tau)} \tag{3}$$

The advantage of SwAV over other recent and successful contrastive learning approaches is, that it allows different images with similar features to form a cluster, therefore being close to each other. This makes SwAV especially applicable to the agricultural image domain. Compared to ImageNet, agricultural datasets contain relatively similar images. E.g. two different images from the same plant species in a similar growing state should be close, while images of different plant species or different growing stages provide less similar features.

4. Experimental Setup

Since we can not directly measure the quality of pre-trained weights, we evaluate their quality by applying downstream tasks to differently initialized models: *Kaiming* initialization (He et al., 2015), supervised *ImageNet* pre-trained weights and weights obtained with unsupervised representation learning using SwAV. Inspired by Azizi et al. (2021), we actually consider two self-training protocols: *SwAV* indicates representation learning from scratch, while in *ImageNet → SwAV* we first load the supervised ImageNet weights and apply self-supervised training with the targeted dataset on top. In the following subsection, the setup of the different experiments (Section 4.2.1 and 4.2.2) and and datasets (Section 4.4) is specified in more detail. Furthermore, in each of our conducted experiments we list the concrete hyper-parameters we used, which in all cases were determined using a grid search approach.

4.1. Unsupervised representation Learning with SwAV

The representation is obtained using SwAV as introduced in Section 3.2. The encoder network (either ResNet-34 or xResNet-34) is followed by a pooling layer and a 2-layer multi-layer perceptron (MLP) with a hidden layer of size 256. We perform experiments with *average pooling* as well as *concat pooling*, which concatenates the output of average and max pooling. The MLP maps the feature vectors to 1000 prototype vectors. We use an image input size of $[244 \times 244]$ and a batch size of 128, that is the maximum we can fit on our Nvidia Tesla V100. Due to the relatively small batch size, we apply a queue of size 960 after 50 epochs for xResnet-34 and after 100 epochs for Resnet-34, containing feature vectors from previous iterations. An earlier introduction of the queue was found to lead to a collapsed training. In total, we train 600 epochs. The network is updated with the ranger optimizer[2] and a constant learning rate of 0.01. The following augmentations are applied to the images: multi-crop (Caron et al., 2020), scaling, color jitter, grayscale transform, Gaussian blur and random rotation. Weights are either initialized with Kaiming intialization or with supervised ImageNet weights.

4.2. Downstream Tasks

We considered two relevant downstream tasks—Classification and Segmentation—for our experiments.

4.2.1. Classification

In this work, we investigate whether agricultural domain-specific pre-trained weights can improve label efficiency. Thus, our goal is not to maximise classification performance as such. Therefore, all our experiments on the classification downstream task are performed with the same set of parameters. The encoder network (either ResNet-34 or xResNet-34) is followed by a classification head, consisting of a pooling layer and a linear layer that projects the feature vectors to final class predictions. We use an input image size of $[244 \times 244]$ and train each model for 50 epochs with a batch size of 32. We use Stochastic Gradient Descent (SGD) as optimizer and cross-entropy loss as criterion. For the pooling layer in the network head we perform experiments with *average pooling* or *concat pooling*. We differentiate between two experiments:

- **Linear Evaluation**. It is a measurement of how well the representation is learned. During training, only the weights of the head are updated, while the encoder weights are kept frozen. The learning rate is set to $lr\_head = 0.01$ and decayed twice at epoch [28, 41] with $\gamma = 0.2$. Augmentations similar to the DeepWeeds baseline in Olsen et al. (2019) are applied during training: random scale, random crop, random rotation, RGB shift, horizontal and vertical flip.
- **Classification Fine-tuning**. The model is fine-tuned with an initial learning rate of $lr\_enc = 0.001$ for the encoder and $lr\_head = 0.01$ for the head. These learning rates are decayed twice at epoch [28, 41] with $\gamma = 0.2$. The same augmentations as in the *Linear Evaluation Model* are applied.

For all our experiments the *average top-1 accuracy* is reported, i.e. for each class the top-1 accuracy is obtained separately and finally, the sum is averaged over all classes.

4.2.2. Segmentation Fine-tuning

We also perform experiments on segmentation downstream tasks, which are—compared to classification downstream tasks—more complex and less related to instance discrimination methods such as SwAV. Again, we don't aim for performance maximization but rather investigate whether transferring weights obtained with SwAV is beneficial in the agriculture domain. Therefore, we apply the widely known UNet (Ronneberger et al., 9351) with a ResNet-34 as encoder. We use the Adam optimizer (Kingma and Ba, 2015) with a cosine annealing (Loshchilov and Hutter, 2017) as learning rate scheduler, while the initial learning rates are 0.0001 for the encoder and 0.001 for the decoder. As criterion we apply Lovasz Loss (Berman et al., 2018). For all segmentation experiments, we report the *mean Intersection over Union (mIoU) – also called Jaccard Index –* (Jaccard, 1912) on the test set. For each class a separate IoU is computed by dividing the number of intersecting pixels between prediction and ground truth with the sum of all pixels presented in both. The mIoU is the averaged sum over all classes.

[2] https://lessw.medium.com/new-deep-learning-optimizer-ranger-synergistic-combination-of-radam-lookahead-for-the-best-of-2dc83f79a48d

Fig. 4. One sample image per species of the DeepWeeds dataset (Olsen et al., 2019). Top row: chinee apple, lantana, parkinsonia, parthenium; bottom row: prickly acacia, rubber vine, siam weed, snake weed.

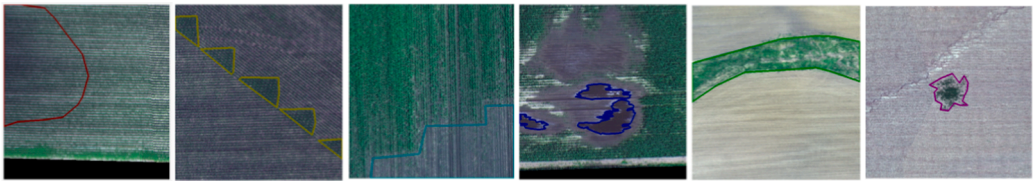


Fig. 5. One sample image per anomaly class of the Aerial Farmland dataset (Chiu et al., 2020). From left to right: clowd shadow, double plant, plant skip, standing water, waterway, weed cluster.

4.3. Similarity of Neural Network Representation

Neural network representations can be compared using Linear Centered Kernel Alignment (CKA) (Cortes et al., 2012; Cristianini et al., 2002), as proposed by Kornblith et al. (2019). Thereby, the similarity between layers of two representations is evaluated. To get a better understanding of what is learned with SwAV, we perform the Linear CKA analysis on representations learned with the DeepWeeds Dataset. We use the code provided by Kornblith et al. (2019) and compute the similarity for the output of each ResNet-34 block, resulting in a similarity matrix of 17×17 blocks. For the similarity computation, we consider 100 randomly drawn test images.

4.4. Datasets

In our work, we benchmark the proposed method on publicly available large datasets such as *DeepWeeds* (Olsen et al., 2019) and *Aerial Farmland* (Chiu et al., 2020). Additionally, we apply the method to a more practical scenario where we collect a relatively small and platform-specific dataset *Rumex Denmark 2020* of the Grassland weed *Rumex obtusifolius*. Then, we enrich the data from *Rumex Denmark 2020* further with images from different Grassland species scraped from the internet, that we call *Grassland Europe*.

4.4.1. DeepWeeds

DeepWeeds (Olsen et al., 2019) is an image classification dataset of weed species from Australian rangelands. In total, it contains 9 different classes—8 weed species shown in Fig. 4 and a background class. It provides 17,509 RGB images of size $[256 \times 256]$, while the number of negative images predominates with a share of 52%. We split the dataset randomly in 60% training–, 20% validation– and 20 % test–split. Experiment-specific settings are defined in the following.

- Representation Learning: All images from the training–split are used to obtain network weights with SwAV.
- Linear Evaluation: We balance the training data by under-sampling all classes to the smallest number, which is represented by the species *Snake Weed*. The balanced images are used to obtain the results for the linear evaluation model. Results are reported on the raw test–split.
- Classification Fine-tuning: We fine-tune the model on the balanced training data as described previously. The model is fine-tuned on the whole balanced training–split as well as on randomly drawn fractions of the training–split (30%, 10%, 1%, 0.2%). For each fraction, we draw 5 different folds and report the mean and standard deviation over the 5 folds. Note, that 0.2% is equal to one-shot learning with one image per class. Results are reported on the raw test–split.

4.4.2. Aerial Farmland

We consider the publicly available challenge dataset[3] for semantic segmentation of six different anomaly patterns, that is a subset of the Aerial Farmland Dataset (Chiu et al., 2020). Example images of the considered anomaly patterns are shown in Fig. 5. Originally, it includes 21,061 (train: 12,901, validation: 4,431, test: 3,729) high-quality aerial RGB as well as corresponding near infra-red (NIR) images from farmlands in the United States of America. Since, it was made public within an on-going challenge, we have no access to the annotations of the test split. Therefore, we split the validation set in two halves serving as validation– and test–splits in our experiments. Furthermore, additional binary masks (which we call *valid_mask* in the following) are provided by Chiu et al. (2020), that indicate which areas in the images are valid farmland and which not, like e.g. streets, rivers or woods. These invalid farmland pixels are marked in black in Fig. 5.

Looking more closely at the data, one can observe that the majority of images contain only a single segmentation class (more precisely train: 11,436, validation: 4,157). This allows us to re-formulate the task as

[3] https://www.agriculture-vision.com/agriculture-vision-2020/dataset

Fig. 6. One sample image per species of the Grasslands Europe dataset. Top row: Cirsium Vulgare, Crepis Vesicaria, Glechoma Hederacea, Plantage Major, Rumex Crispus; bottom row: Rumex Obtusifolius, Ulex Europaeus, Urtica Dioica, Urtica Urens.

Fig. 7. Sample images of *Rumex Denmark 2020* dataset from different fields and days with overlayed segmentation mask in orange.

classification task, where images containing annotations of only one class are considered and labeled with that specific class. The information of the *valid_mask* is incorporated by blacking out the stated invalid pixels. We consider only RGB images of size [244 × 244]. Experiment-specific settings are defined in the following.

- Representation Learning: All images from the training–split are used to obtain network weights with SwAV.
- Linear Evaluation: The dataset is highly imbalanced. In contrast to DeepWeeds, under-sampling is no option because it would reduce the amount of data too heavily. Instead, we weight the contribution of each class in the cross entropy loss by the inverse of their number of appearances. Results are reported on the test–split.
- Classification Fine-tuning: During fine-tuning, we use the weighted cross-entropy loss as described in the Linear Evaluation. We evaluate the classification performance leaned on the classification experiments of the DeepWeeds dataset: The model is fine-tuned on the whole training–split as well as on randomly drawn fractions of the training–split (30%, 10%, 1%), while 5 different folds are drawm and the mean and standard deviation are reported over these 5 folds. For the Aerial Farmland dataset, 1% is the lowest possible fraction because then one of the classes is represented with exactly one image.
- Segmentation Fine-tuning: As with the other experiments, we use an input image size of [244 × 244] and train each model for ≈ 12000 iterations with a batch size of 32. The model is fine-tuned on the whole training–split as well as on fractions (10%, 1%) like in previous experiments.

4.4.3. Grassland Europe

We created a dataset specifically for applications in Grasslands of Europe by scraping relevant images from PlantNet[4]. In total, we downloaded 24,665 RGB images of nine common grassland species in Europe: Cirsium Vulgare, Crepis Vesicaria, Glechoma Hederacea, Plantago Major, Rumex Crispus, Rumex Obtusifolius, Ulex Europaeus, Urtica Dioica, Urtica Urens. Example images for each species are shown in Fig. 6. All images are resized to 244 pixels on the smaller edge, while keeping the original aspect ratio, followed by a random crop of [244 × 244]. We split the dataset in 60 % training–, 20% validation– and 20% test–split. Experiment-specific settings are defined in the following.

- Representation Learning: All images from the training–split are used to obtain network weights with SwAV.

[4] Pl@ntNet project: https://identify.plantnet.org/

Table 2
Overview of performed experiments.

Experiments	Model	Dataset			
		DeepWeeds (Olsen et al., 2019)	Aerial Farmland (Chiu et al., 2020)	Grassland Europe	Rumex Denmark 2020
Linear Evaluation	ResNet-34	✓	✓	✓	
Classification Fine-tuning	ResNet-34	✓	✓	✓	
	xResNet-34	✓			
Segmentation Fine-tuning	ResNet-34		✓		✓
Representation Similarity	ResNet-34	✓			

Table 3
Linear Evaluation Model for the three classification datasets: DeepWeeds, Aerial Farmland and Grassland Europe. During fine-tuning, only the layers on top of the encoder architecture are updated. We compare the average top-1 accuracy of four weight initialization, fine-tuned on 100% of the data.

	DeepWeeds	Aerial Farmland	Grassland Europe
weight initialization	average top-1 accuracy		
Kaiming	29.5	37.7	20.7
ImageNet	75.3	61.4	71.2
SwAV	94.4	68.4	85.5
ImageNet → SwAV	**94.9**	**70.6**	**86.4**

- Linear Evaluation: Similarly to the Aerial Farmland dataset, we address the class imbalance by weighting each contribution to the cross entropy loss.
- Classification Fine-tuning: Again, the model is fine-tuned on the whole training–split as well as on randomly drawn fractions of the training–split (30%, 10%, 1%, 0.5%). We report the mean and standard deviation of the average top-1 accuracy over the 5 folds. For the Grassland Europe dataset, 0.5% is the lowest possible fraction because then one of the classes is represented with exactly one image.

4.4.4. Rumex Denmark 2020

Furthermore, we have collected 282 images of the invasive grassland species *Rumex obtusifolius* and created manual pixel-wise annotations. The images have been taken with a 8-megapixel 1/3-inch camera with a resolution of $[2448 \times 3263]$ while freely moving through the field. As a consequence, the collected images contain different levels of blur. Images have been collected at four different locations in Denmark on four different days in Spring and Autumn 2020, providing a certain level of variance in lightning, plant as well as background appearance. Example data is shown in Fig. 7.

- Segmentation Fine-tuning: The model is fine-tuned on image sizes of $[480 \times 640]$ for 30 epochs and with a batch size of 32. In this experiment, we only consider 100% of the data because the dataset is already relatively small.

5. Results and Discussion

Now, we will show and discuss the results for the defined experiments in Section 4, while Table 2 serves as an overview of which experiments are performed for which datasets.

5.1. Linear Evaluation

In this section, we examine how well the representation is learned with SwAV for the different datasets. We perform *Linear Evaluation* for the ResNet-34 and compare different weight initialization: Kaiming initialization, supervised ImageNet pre-trained weights and weights obtained with unsupervised representation learning using SwAV. Table 3 lists the results for all three classification datasets. For all three datasets, the learned representation using SwAV outperforms the ImageNet representation significantly. The results confirm the expectation that more data-specific features are learned. Note, that initializing SwAV with ImageNet weights (ImageNet → SwAV) even leads to a slightly better representation. Another advantage of initializing the self-training with ImageNet weights is faster convergence during training as shown in Fig. 8. For all three datasets, the loss drops faster for networks that have been initialized with ImageNet weights (*ImageNet → SwAV*, green). A possible explanation is that general image features such as edges and corners are learned already and SwAV can directly start learning more data-specific features.

5.2. Classification Fine-tuning

In this section, we examine whether our newly obtained pre-trained weights can reduce the amount of labeled data needed for fine-tuning. Our model has been fine-tuned on different fractions of the training dataset for all three classification datasets as shown in Table 4. Note,

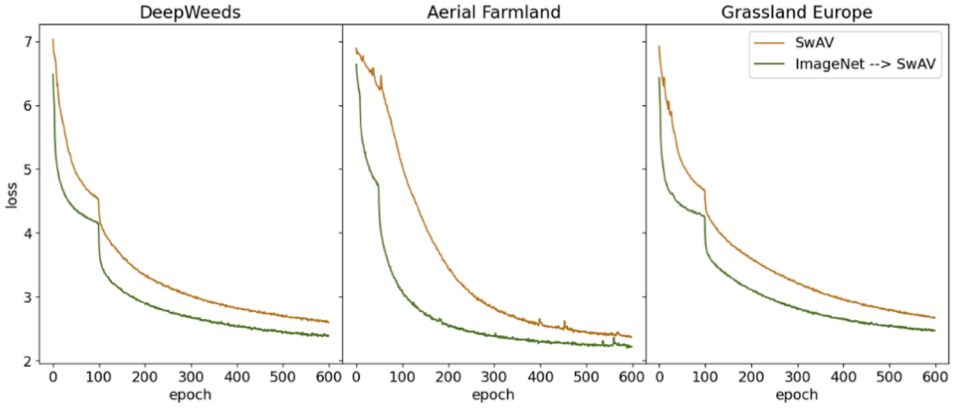


Fig. 8. Training loss curve for training from scratch (*SwAV*, orange) vs. with ImageNet initialization (*ImageNet → SwAV*, green) for all three datasets. For all three datasets, the loss drops faster for networks that have been initialized with ImageNet weights. Note, that the knik is caused by the introduction of the queue.

Table 4
Classification performance for different weight initialization and all three agricultural classification datasets. The model is fine-tuned on the complete training dataset (100%) and on randomly drawn subsets. For each dataset the lowest possible fraction is where at least one of the classes is represented with exactly one image, i.e. DeepWeeds: 0.2%, Aerial Farmland: 1%, Grassland Europe 0.5%.

train img fraction [%]	weight init	DeepWeeds	Aerial Farmland	Grassland Europe
		average top-1 accuracy (mean ± std)		
100	Kaiming	73.8 ± 0	65.0 ± 0	73.6 ± 0
	ImageNet	94.3 ± 0	**80.6** ± 0	**91.7** ± 0
	SwAV	95.9 ± 0	72.1 ± 0	88.6 ± 0
	ImageNet → SwAV	**96.1** ± 0	77.2 ± 0	89.9 ± 0
30	Kaiming	51.7 ± 2.0	50.9 ± 2.0	42.8 ± 1.1
	ImageNet	89.9 ± 0.9	67.8 ± 9.3	86.7 ± 0.3
	SwAV	93.5 ± 0.7	70.5 ± 1.1	85.9 ± 0.5
	ImageNet → SwAV	**94.6** ± 0.7	**72.2** ± 1.0	**88.3** ± 0.3
10	Kaiming	40.2 ± 0.8	38.4 ± 2.0	29.4 ± 0.5
	ImageNet	82.3 ± 1.0	57.5 ± 10.6	80.4 ± 0.8
	SwAV	89.4 ± 0.4	**68.4** ± 1.1	**83.3** ± 1.2
	ImageNet → SwAV	**92.0** ± 0.7	68.3 ± 0.8	83.1 ± 7.4
1	Kaiming	25.4 ± 0.6	28.5 ± 1.9	17.9 ± 0.8
	ImageNet	54.0 ± 2.7	50.7 ± 2.9	53.2 ± 2.0
	SwAV	68.0 ± 4.0	**56.6** ± 2.3	67.9 ± 2.7
	ImageNet → SwAV	**69.0** ± 0.3	56.4 ± 2.5	**74.3** ± 2.3
0.5	Kaiming	22.5 ± 2.2		17.3 ± 0.7
	ImageNet	45.6 ± 2.5		46.7 ± 1.1
	SwAV	60.0 ± 2.8		62.7 ± 0.9
	ImageNet → SwAV	**62.8** ± 2.6		**69.7** ± 3.0
0.2	Kaiming	17.7 ± 3.7		
	ImageNet	33.8 ± 4.1		
	SwAV	44.3 ± 4.2		
	ImageNet → SwAV	**46.4** ± 3.6		

that for each dataset the lowest possible fraction is where at least one of the classes is represented with exactly one image. The results show, that SwAV pre-training is especially effective when fine-tuning on less labeled data. For all datasets and all chosen subsets, we get a higher accuracy up to 15% with SwAV weights. For DeepWeeds, using SwAV pre-training, even with 30% of labeled data, we outperform the classical transfer learning with ImageNet weights and 100% of labeled data. However, for Aerial Farmland and Grassland Europe fine-tuning on all images (100%), leads to a better performance with standard ImageNet weights. All in all, the results show that one can profit highly from SwAV pre-training, when the image data collection is relatively easy, while the labour of manual annotations remains expensive. One would pre-train on all collected images, while fine-tuning only on a fraction labeled images.

Unsupervised Representation Learning is not just a way to generate better weight initialization but also provides a flexibility in the design of encoder networks. One doesn't rely anymore on architectures, that come with pre-trained ImageNet weights such as ResNet (He et al., 2016), VGG16 (Simonyan and Zisserman, 2015) or InceptionV3 (Szegedy et al., 2016). Exemplary, we modified the baseline encoder ResNet-34, obtained new SwAV weights and fine-tuned the network on the fractions of the DeepWeeds dataset. Two modifications are performed:

- Concat Pooling: We simply switch out the average pooling in the classification head with concat pooling. Since this modification doesn't affect the encoder network directly, we can further compare SwAV weights with ImageNet weights.
- xResNet-34: Additionally, we use a xResNet-34 as described in Section 3.1. ImageNet weights do not exist for the xResNet-34 architecture. Therefore, we only apply SwAV pre-training which is computationally cheaper and therefore feasible to obtain.

Table 5 lists the average top-1 accuracy for the mentioned modifications as well as the performance difference compared to Table 4. Interestingly, replacing average pooling with concat pooling improved performance for the SwAV initialization, but reduced the performance heavily for the ImageNet weights. The performance drop can be explained with the fact that pre-trained ImageNet weights have been obtained with average pooling in the classification head, while for the generation of SwAV weights, we directly used concat pooling to adapt the encoder design optimally. Furthermore, the modification that come with the xResNet architecture improved the average top-1 accuracy even further for all data fractions. With only 30% of the training data, the xResNet-34 outperforms the results obtained with ResNet-34 initialized with ImageNet weights and 100% of the data (cf. Table 4). These results show that one is not restricted anymore to architectures that come with pre-trained weights ImageNet but can design application-specific architectures and obtain pre-trained weights relatively cheaply using self-supervised contrastive learning. Fig. 1 visually reflects the main results for fine-tuning on DeepWeeds with different weight initialization and network modifications. Note, that the presented model modifications are specifically selected for DeepWeeds and did not show the same effect on the other two datasets.

It is also worth mentioning that the cost of obtaining SwAV weights is lower, compared to the generation of ImageNet pre-trained weights. We performed our SwAV training only on a single Nvidia Tesla V100. The training time scales obviously with the number of images per epoch, but also with the selected augmentations methods; e.g. Gaussian blur is a time-consuming augmentation method that increases the training time

Table 5
We apply two network modifications: (1) ResNet-34 + concat pooling in the classification head, (2) xResNet-34 modifications, and show their effects on the fine-tuning performance of the balanced training dataset (100%) and training subsets (30%, 10%, 1%, 0.2%) of the DeepWeeds dataset.

train img fraction [%]	arch	weight init	avg top-1 acc (mean ± std)	diff
100	ResNet-34 (+ concat pool)	Kaiming	50.5 ± 0	−23.3
		ImageNet	92.9 ± 0	−1.4
		SwAV	96.3 ± 0	+ 0.4
	xResNet-34	SwAV	**96.6** ± 0	+ 0.7
30	ResNet-34 (+ concat pool)	Kaiming	39.0 ± 2.9	−12.7
		ImageNet	84.5 ± 1.5	−5.4
		SwAV	94.4 ± 0.6	+ 0.9
	xResNet-34	SwAV	**95.0** ± 0.5	+ 1.5
10	ResNet-34 (+ concat pool)	Kaiming	27.4 ± 3.1	−12.8
		ImageNet	63.7 ± 4.6	−18.6
		SwAV	91.0 ± 0.7	+ 1.6
	xResNet-34	SwAV	**92.6** ± 0.4	+ 3.2
1	ResNet-34 (+ concat pool)	Kaiming	19.4 ± 0.9	−6.0
		ImageNet	30.6 ± 5.2	−23.4
		SwAV	74.2 ± 3.0	+ 6.2
	xResNet-34	SwAV	**76.2** ± 1.6	+ 8.2
one-shot	ResNet-34 (+ concat pool)	Kaiming	16.9 ± 2.0	−0.8
		ImageNet	28.0 ± 3.8	−5.8
		SwAV	44.7 ± 3.3	+ 0.4
	xResNet-34	SwAV	**45.9** ± 3.6	+ 1.2

of SwAV significantly. The SwAV training for a ResNet-34 on Deepweeds with 10513 training images for 600 epochs lasts 13:19 (h:min)—without Gaussian blur augmentation this can be reduced to 7:19 (h:min).

5.3. Similarity of Neural Network Representation for DeepWeeds

In Fig. 9, the similarity of different representations is shown. Fig. 9a shows the similarity of the SwAV and the ImageNet representation. The similarity matrix undermines the expectation, that early blocks ($\approx 0-7$) learn more general low-level features and later layers ($\approx 8-16$) contain more dataset-specific high-level features which certainly differ between the two representations. In Fig. 9b, c we compare representations of pre-trained weights with final representations of the trained classifier to investigate how much the initial weights have evolved throughout fine-tuning the network. Fig. 9b shows how much ImageNet weights have changed during fine-tuning. There is a high similarity on the diagonal of the first half of the network, meaning early layers have not changed much throughout fine-tuning, while later layers have evolved more strongly. On the contrary, in Fig. 9c the bright diagonal stretches out through the whole network. The weights of the network initialized with SwAV have not changed much during fine-tuning. It can be concluded, that during self-supervised training relevant data-specific features have been learned for the classification task. These findings are in correspondence with the results from Section 5.1.

Table 6
Segmentation performance on the Aerial Farmland dataset for different weight initialization. The model is fine-tuned on the complete (100%) and on randomly drawn subsets (30%, 10%, 1%) of the Aerial Farmland dataset.

train img fraction [%]	weight init	mIoU (mean ± std)
100	Kaiming	0.296 ± 0
	ImageNet	**0.377** ± 0
	SwAV	0.367 ± 0
	ImageNet → SwAV	0.365 ± 0
10	Kaiming	0.283 ± 0.011
	ImageNet	**0.388** ± 0.005
	SwAV	0.352 ± 0.008
	ImageNet → SwAV	0.370 ± 0.007
1	Kaiming	0.224 ± 0.011
	ImageNet	**0.304** ± 0.019
	SwAV	0.289 ± 0.012
	ImageNet → SwAV	0.293 ± 0.016

Table 7
Segmentation performance on the Rumex Denmark 2020 dataset for different weight initializations. As SwAV initialization, we use the pre-trained weights obtained with the Grassland Europe and DeepWeeds datasets since they are semantically closest to the Rumex segmentation task. The whole Rumex Denmark 2020 dataset is considered since it is already relatively small with 282 images.

weight init	pre-trained dataset	mIoU
Kaiming	-	0.758
ImageNet	ImageNet	0.836
SwAV	Grassland Europe	0.830
ImageNet → SwAV	Grassland Europe	0.834
SwAV	DeepWeeds	0.820
ImageNet → SwAV	DeepWeeds	0.839

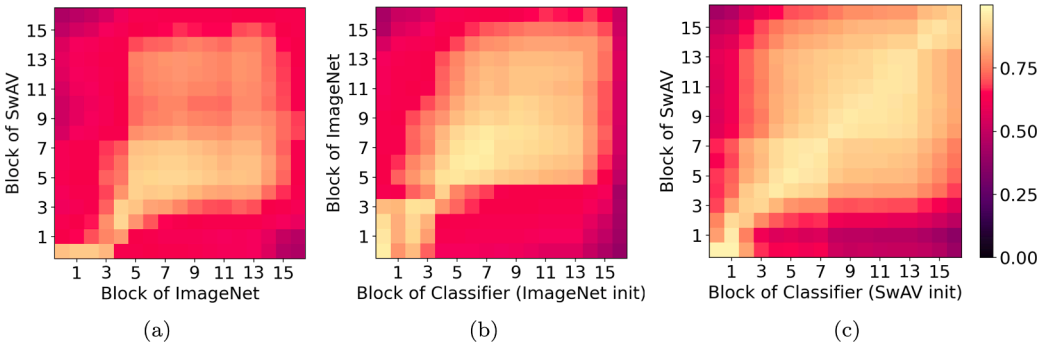


Fig. 9. Similarity between ResNet-34 blocks for different representations. In (a) SwAV weights are more similar to ImageNet weights in lower layers. Thus, the low-level features learned turn out to be common in both cases. In (b), (c) we compare representations of pre-trained weights (on the vertical axes) with the final representations of trained classifiers (on horizontal axes) to show how much the initial weights have evolved throughout fine-tuning the network. For SwAV (c), the bright diagonal stretches out through the whole similarity matrix, which indicates that low-level as well as high-level layers do not change much during fine-tuning. We can conclude, that—contrary to ImageNet (b)—relevant high-level features have been extracted during self-supervised pre-training.

5.4. Segmentation Fine-tuning

In this section, we present experiments on two segmentation downstream tasks. The encoder is initialized with transfer-learning, while the decoder is always initialized with Kaiming weights. This means, that only half of the network layers are transferred. For the selected segmentation tasks, our data-specific obtained SwAV weights do not perform as convincing as in classification downstream tasks. More specifically, they do not improve over standard ImageNet weights, not even for very small fractions of training data. For *Aerial Farmland* as shown in Table 6, models with ImageNet pre-trained weights perform clearly better. For *Rumex Denmark 2020* as shown in Table 7, the mIoU is very close for ImageNet weights and the different SwAV variants, pre-trained on the two weed datasets *Grassland Europe* and *Deepweeds*.

Segmentation downstream tasks are generally more complex and less similar to instance discriminstaion methods like SwAV. Since classification is performed on pixel-level, different features might be relevant in order to obtain accurate segmentation masks. Therefore, instance discrimination methods like SwAV do not seem to be suitable for generating data-specific pre-trained weights for more complex downstream tasks.

6. Conclusion

In this work, we investigated the potential of self-supervised pre-training in the context of agricultural images. As example, we applied one of the recent popular contrastive learning methods—SwAV (Caron et al., 2020)—to three different agricultural datasets. For each dataset, we generated data-specific pre-trained weights and show their clear success for classification downstream tasks, in particular improving label efficiency in semi-supervised settings. Furthermore, the computational cost for generating data-specific pre-trained weights is fairly low, allowing one to generate easily new pre-trained weights for any custom model architecture. We also showed that SwAV is less effective for segmentation downstream tasks, showing no significant improvement on the mIoU performance as shown in Tables 6 and 7. SwAV belongs to the group of instance discrimination methods, learning features on image level but not explicitly on pixel level, which can be considered as crucial for segmentation downstream tasks. With the late success of contrastive instance discrimination methods, it is expected that contrastive self-supervised approaches on pixel level will gain attention within research in computer vision to increase performance on more complex downstream tasks. Just recently, work in that area has been published (Xie et al., 2021). In our experiments, we clearly showed the potential of self-supervised approaches in the domain of agriculture and therefore, future developments will be followed carefully and taken into consideration for segmentation downstream tasks.

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.

Acknowledgements

This work has been supported by the European Commission and European GNSS Agency through the project “Galileo-assisted robot to tackle the weed Rumex obtusifolius and increase the profitability and sustainability of dairy farming (GALIRUMI)”, H2020-SPACE-EGNSS-2019–870258.

References

Olsen, A., Konovalov, D.A., Philippa, B., Ridd, P., Wood, J.C., Johns, J., Banks, W., Girgenti, B., Kenny, O., Whinney, J., Calvert, B., Rahimi Azghadi, M., White, R.D., 2019. DeepWeeds: A Multiclass Weed Species Image Dataset for Deep Learning. Scientific Reports 9.

Chebrolu, N., Lottes, P., Schaefer, A., Winterhalter, W., Burgard, W., Stachniss, C., 2017. Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields. The International Journal of Robotics Research.

S. Azizi, B. Mustafa, F. Ryan, Z. Beaver, J. Freyberg, J. Deaton, A. Loh, A. Karthikesalingam, S. Kornblith, T. Chen, V. Natarajan, M. Norouzi, Big self-supervised models advance medical image classification, ArXiv abs/2101.05224 (2021).

Argüeso, D., Picon, A., Irusta, U., Medela, A., San-Emeterio, M.G., Bereciartua, A., Alvarez-Gila, A., 2020. Few-shot learning approach for plant disease classification using images taken in the field. Computers and Electronics in Agriculture 175, 105542.

Güldenring, R., Boukas, E., Ravn, O., Nalpantidis, L., 2021. Few-leaf learning: Weed segmentation in grasslands. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

Madsen, S.L., Dyrmann, M., Jørgensen, R.N., Karstoft, H., 2019. Generating artificial images of plant seedlings using generative adversarial networks. Biosyst. Eng. 187, 147–159.

Skovsen, S., Dyrmann, M., Mortensen, A.K., Laursen, M.S., Gislum, R., Eriksen, J., Farkhani, S., Karstoft, H., Jørgensen, R.N., 2019. The grassclover image dataset for semantic and hierarchical species understanding in agriculture. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 2676–2684.

T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: H.D. III, A. Singh (Eds.), Proceedings of the 37th International Conference on Machine Learning, volume 119 of Proceedings of Machine Learning Research, PMLR, 2020, pp. 1597–1607.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A., 2020. Unsupervised learning of visual features by contrasting cluster assignments. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (Eds.), Advances in Neural Information Processing Systems, volume 33. Curran Associates Inc, pp. 9912–9924.

M.T. Chiu, X. Xu, Y. Wei, Z. Huang, A.G. Schwing, R. Brunner, H. Khachatrian, H. Karapetyan, I. Dozier, G. Rose, D. Wilson, A. Tudor, N. Hovakimyan, T.S. Huang, H. Shi, Agriculture-vision: A large aerial image database for agricultural pattern analysis, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

Jing, L., Tian, Y., 2021. Self-supervised visual feature learning with deep neural networks: A survey. IEEE Transactions on Pattern Analysis & Machine Intelligence 43, 4037–4058.

A. Dosovitskiy, J.T. Springenberg, M. Riedmiller, T. Brox, Discriminative unsupervised feature learning with convolutional neural networks, in: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1, NIPS'14, MIT Press, Cambridge, MA, USA, 2014, p. 766–774.

S. Gidaris, P. Singh, N. Komodakis, Unsupervised representation learning by predicting image rotations, in: International Conference on Learning Representations, 2018.

Doersch, C., Gupta, A., Efros, A.A., 2015. Unsupervised visual representation learning by context prediction. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1422–1430.

Noroozi, M., Favaro, P., 2016. Unsupervised learning of visual representations by solving jigsaw puzzles. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.), Computer Vision – ECCV 2016. Springer International Publishing, Cham, pp. 69–84.

R. Zhang, P. Isola, A.A. Efros, Colorful image colorization, in: ECCV, 2016.

Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., Efros, A.A., 2016. Context encoders: Feature learning by inpainting. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2536–2544.

I. Misra, C.L. Zitnick, M. Hebert, Unsupervised learning using sequential verification for action recognition, CoRR abs/1603.08561 (2016).

Caron, M., Bojanowski, P., Joulin, A., Douze, M., 2018. Deep clustering for unsupervised learning of visual features. In: European Conference on Computer Vision.

Yang, J., Parikh, D., Batra, D., 2016. Joint unsupervised learning of deep representations and image clusters. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5147–5156.

M. Gutmann, A. Hyvürinen, Noise-contrastive estimation: A new estimation principle for unnormalized statistical models, in: Y.W. Teh, M. Titterington (Eds.), Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, volume 9 of Proceedings of Machine Learning Research, PMLR, Chia Laguna Resort, Sardinia, Italy, 2010, pp. 297–304.

A. van den Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, ArXiv abs/1807.03748 (2018).

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, IEEE, 2009, pp. 248–255.

He, K., Fan, H., Wu, Y., Xie, S., Girshick, R., 2020. Momentum contrast for unsupervised visual representation learning. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9726–9735.

J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, B. Piot, k. kavukcuoglu, R. Munos, M. Valko, Bootstrap your own latent - a new approach to self-supervised learning, in: H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, volume 33, Curran Associates Inc, 2020, pp. 21271–21284.

Chen, X., He, K., 2021. Exploring simple siamese representation learning, in. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 15750–15758.

Kounalakis, T., Triantafyllidis, G.A., Nalpantidis, L., 2019. Deep learning-based visual recognition of rumex for robotic precision farming. Computers and Electronics in Agriculture 165, 104973.
Mehdipour Ghazi, M., Yanikoglu, B., Aptoula, E., 2017. Plant identification using deep neural networks via optimization of transfer learning parameters. Neurocomputing 235, 228–235.
H.K. Suh, J. IJsselmuiden, J.W. Hofstee, E.J. van Henten, Transfer learning for the classification of sugar beet and volunteer potato under field conditions, Biosystems Engineering 174 (2018) 50–65.
Bargoti, S., Underwood, J., 2017. Deep fruit detection in orchards. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 3626–3633.
Bosilj, P., Aptoula, E., Duckett, T., Cielniak, G., 2020. Transfer learning between crop types for semantic segmentation of crops versus weeds in precision agriculture. J. Field Robotics 37, 7–19.
Espejo-Garcia, B., Mylonas, N., Athanasakos, L., Fountas, S., 2020. Improving weeds identification with a repository of agricultural pre-trained deep neural networks. Computers and Electronics in Agriculture 175, 105593.
Espejo-Garcia, B., Mylonas, N., Athanasakos, L., Fountas, S., Vasilakoglou, I., 2020. Towards weeds identification assistance through transfer learning. Computers and Electronics in Agriculture 171, 105306.
P.A. Marin Zapata, S. Roth, D. Schmutzler, T. Wolf, E. Manesso, D.-A. Clevert, Self-supervised feature extraction from image time series in plant phenotyping using triplet networks, Bioinformatics (Oxford, England) (2020).
Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., Wu, Y., 2014. Learning fine-grained image similarity with deep ranking. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1386–1393.
Zhao, Z., Luo, Z., Li, J., Chen, C., Piao, Y., 2020. When self-supervised learning meets scene classification: Remote sensing scene classification based on a multitask learning framework. Remote Sensing 12.
dos Santos Ferreira, A., Freitas, D.M., da Silva, G.G., Pistori, H., Folhes, M.T., 2019. Unsupervised deep learning and semi-automatic data labeling in weed discrimination. Computers and Electronics in Agriculture 165, 104963.
He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., Li, M., 2019. Bag of tricks for image classification with convolutional neural networks. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 558–567.
Cuturi, M., 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems, volume 26. Curran Associates Inc.
He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), ICCV '15. IEEE Computer Society, USA, pp. 1026–1034.
O. Ronneberger, P.Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention (MICCAI), volume 9351 of LNCS, Springer, 2015, pp. 234–241.
D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings, 2015.
I. Loshchilov, F. Hutter, SGDR: stochastic gradient descent with warm restarts, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings, OpenReview.net, 2017.
Berman, M., Rannen Triki, A., Blaschko, M.B., 2018. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4413–4421.
Jaccard, 1912. The distribution of the flora of the alpine zone. New Phytol. 11, 37–50.
Cortes, C., Mohri, M., Rostamizadeh, A., 2012. Algorithms for learning kernels based on centered alignment. J. Mach. Learn. Res. 13, 795–828.
Cristianini, N., Shawe-Taylor, J., Elisseeff, A., Kandola, J., 2002. On kernel-target alignment. In: Dietterich, T., Becker, S., Ghahramani, Z. (Eds.), Advances in Neural Information Processing Systems, volume 14. MIT Press.
S. Kornblith, M. Norouzi, H. Lee, G. Hinton, Similarity of neural network representations revisited, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, PMLR, 2019, pp. 3519–3529.
He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778.
K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings, 2015.
Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2016. Rethinking the inception architecture for computer vision. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.
Z. Xie, Y. Lin, Z. Zhang, Y. Cao, S. Lin, H. Hu, Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning (2021).