Self-Supervised Leaf Segmentation under Complex Lighting Conditions

Xufeng Lin[a,*], Chang-Tsun Li[a], Scott Adams[b], Abbas Kouzani[b], Richard Jiang[e], Ligang He[d], Yongjian Hu[f], Michael Vernon[b], Egan Doeven[b], Lawrence Webb[c], Todd Mcclellan[g], Adam Guskich[g]

[a]*School of Information Technology, Deakin University, Waurn Ponds, Australia*
[b]*School of Engineering, Deakin University, Waurn Ponds, Australia*
[c]*School of Life and Environmental Sciences, Deakin University, Waurn Ponds, Australia*
[d]*Department of Computer Science, The University of Warwick, Coventry, UK*
[e]*School of Computing and Communications, Lancaster University, Lancaster, UK*
[f]*School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China*
[g]*Medigrowth Australia Pty Ltd, Waurn Ponds, Australia*

Abstract

As an essential prerequisite task in image-based plant phenotyping, leaf segmentation has garnered increasing attention in recent years. While self-supervised learning is emerging as an effective alternative to various computer vision tasks, its adaptation for image-based plant phenotyping remains rather unexplored. In this work, we present a self-supervised leaf segmentation framework consisting of a self-supervised semantic segmentation model, a color-based leaf segmentation algorithm, and a self-supervised color correction model. The self-supervised semantic segmentation model groups the semantically similar pixels by iteratively referring to the self-contained information, allowing the pixels of the same semantic object to be jointly considered by the color-based leaf segmentation algorithm for identifying the leaf regions. Additionally, we propose to use a self-supervised color correction model for images taken under complex illumi-

*Corresponding author
Email addresses: xufeng.lin@deakin.edu.au (Xufeng Lin), changtsun.li@deakin.edu.au (Chang-Tsun Li), scott.adams@deakin.edu.au (Scott Adams), abbas.kouzani@deakin.edu.au (Abbas Kouzani), r.jiang2@lancaster.ac.uk (Richard Jiang), ligang.he@warwick.ac.uk (Ligang He), eeyjhu@scut.edu.cn (Yongjian Hu), m.vernon@deakin.edu.au (Michael Vernon), egan.doeven@deakin.edu.au (Egan Doeven), lawrence.webb@deakin.edu.au (Lawrence Webb), todd@medigrowth.com.au (Todd Mcclellan), adam@medigrowth.com.au (Adam Guskich)

nation conditions. Experimental results on datasets of different plant species demonstrate the potential of the proposed self-supervised framework in achieving effective and generalizable leaf segmentation[1].

Keywords: Self-supervised learning, convolutional neural networks, image-based plant phenotyping, leaf segmentation, color correction, cannabis.

1. Introduction

Plant phenotyping is the field of scientific inquiry concerned with the quantitative measurement of observable plant traits [1–4] developed from the dynamic interaction of the genotype with environmental conditions. It provides an important tool to understand the effects of environment on the cultivated plants, and enables a wide range of applications in plant breeding [1], crop monitoring [5, 6], disease prevention and control [7, 8], etc. While traditional plant phenotyping relies on labor-intensive and error-prone manual measurements, the advances in digital imaging and computer vision techniques have allowed for quantifying plant traits from images in a non-invasive and automatic manner. In achieving the goal of image-based plant phenotyping, automated segmentation of plant leaves is the fundamental prerequisite for measuring more complex phenotypic traits. It is often performed at two granular levels: category-level and instance-level. The former is concerned with segmenting the pixels belonging to the 'leaf' category from background, while the latter moves a step further and separates individual leaves from each other. Instance-level leaf segmentation allows for fine-grained measurement of individual leaf area, leaf count and leaf growth rate, which could be beneficial for responsive plant growth monitoring and growth regulation [9]. However, the variability in leaf shape and appearance, constant self-occlusion and varying imaging conditions often render instance-level leaf segmentation an extremely challenging problem even in controlled environments. In comparison, category-level leaf segmentation is rel-

[1]The developed code and datasets will be made publicly available on https://github.com/lxfhfut/Self-Supervised-Leaf-Segmentation

atively easier and provides a good approximation of plant size, thus making it a more feasible and practical means for many application scenarios, such as plant growth monitoring [10] and yield prediction [11].

Facilitated by the Leaf Segmentation Challenge (LSC) of the Computer Vision Problems in Plant Phenotyping (CVPPP) workshop, significant advances have been achieved for both category-level and instance-level leaf segmentation. Focusing on rosette plants [12], the LSC challenge has been instrumental in advancing the research in leaf segmentation and beyond within the application domain of image-based plant phenotyping. While earlier works rely on hand-crafted image features [13–15], state-of-the-art methods [16–18] are mainly based on supervised training of deep convolutional neural networks (CNNs), particularly U-net [19] and Mask-RCNN [20], which have demonstrated superior performance in segmenting common objects, e.g., person and car. However, despite the substantial progress made over the years, there are still many challenges hindering the wide applicability of existing techniques in practical deployment:

- Firstly, the training of deep models requires a large amount of annotated data, but obtaining pixel-wise annotation for segmentation could be a highly labor-intensive, time-consuming and error-prone process. For the image-based plant phenotyping problem at hand, annotated data must contain sufficient examples of different mutations, genotypes and environmental conditions covering different growth stages, which makes the problem thornier than expected.

- Secondly, deep learning models trained on datasets of specific plant species usually do not generalize well to other unseen species. This problem is particularly acute for plant leaf segmentation because different plant species vary dramatically in leaf appearances [18, 21]. It is often required to re-train a model from scratch or fine-tune a pre-trained model on annotated datasets of unseen plant species to achieve satisfactory performance across plant species.

- Lastly, dramatic changes in the background and appearance of plant leaves caused by varying lighting conditions adds another dimension to the challenges faced by image-based plant phenotyping. Along with other factors such

as weather conditions or different times of the day, this will result in complex lighting conditions for plant image acquisition, and pose a huge challenge for leaf segmentation. Surprisingly, to the best of our knowledge, there has been limited research on investigating the effectiveness of leaf segmentation under different lighting conditions. Given that the use of artificial light has become very common in greenhouse cultivation to supplement natural sunlight, this is a significant omission as it precludes the use of many existing algorithms on plants grown under artificial lighting conditions.

To mitigate the aforementioned problems, there have been many attempts to generate new synthetic data samples by 3D plant modelling [22], Generative Adversary Networks (GANs) [23, 24], domain randomization [17, 18, 21], etc. However, it is difficult and often impossible to accurately simulate different plant characteristics, various environmental conditions, and complex interplay between genetic and environmental factors, which inevitably creates a gap between the real and synthetic data. In this work, we propose to surmount the above challenges by developing a self-supervised learning framework for leaf segmentation under complex lighting conditions without using any annotated data. Specifically, we make the following contributions:

1. We propose a novel self-supervised semantic segmentation model. It integrates the feature extraction power of Convolutional Neural Networks (CNNs) with the structured modeling capabilities of fully connected Conditional Random Fields (CRFs). It allows the pixels of the same semantic object to be jointly processed, thus significantly reducing the impact of complex backgrounds and variations within the leaf and non-leaf regions.

2. We propose a color-based leaf segmentation algorithm. It models the "greenness" of semantic objects in an image with the multivariate normal distribution in the HSV color space, and identifies the regions with admissible *absolute* and *relative* greenness as leaf regions.

3. We propose a self-supervised color correction model to rectify the "distorted"

color in an image caused by the use of artificial grow lights. In so doing, the color-corrected images can be segmented in the same way as for the images taken under "natural" daylight conditions.

4. We publish a dataset of top-view cannabis images captured in a greenhouse equipped with grow lights to facilitate the research in the area of image-based plant phenotyping.

The remainder of this manuscript is organized as follows. We first review the literature relevant to the proposed method in Section 2. The details of the proposed self-supervised leaf segmentation framework are presented in Section 3, followed by comprehensive experimental results in Section 4. The concluding remarks along with a discussion of future works are given in Section 5.

2. Related Works

Unsupervised leaf segmentation. Unsupervised image segmentation aims to partition an image into groups of perceptually or semantically similar pixels without resorting to the ground-truth annotations. In the specific case of leaf segmentation, traditional unsupervised clustering algorithms, e.g., expectation maximization (EM) algorithm [25], K-means [26], and fuzzy clustering [27], based on color [25, 26, 28], shape [28], texture [29] features have been widely adopted to distinguish the "leaf" pixels from the background. These methods are usually employed in conjunction with superpixel algorithms to enhance the spatial consistency and boundary adherence of the segmentation result. As can be expected, these methods inevitably inherit the shortcoming of being sensitive to parameters and outliers from the adopted traditional clustering algorithms. Consequently, tedious parameter tuning and ad-hoc post-processing are usually required to obtain satisfactory segmentation results on specific datasets. Moreover, the image features, particularly the shape and texture features, that these unsupervised leaf segmentation algorithms rely on are specially designed for a specific plant species and do not generalize well across a variety of plant

species. One may argue that the color feature is generalizable over different plant species because the leaves of most plants are green due to the presence of chlorophyll. Indeed, the color feature is arguably the most widely used feature in plant-related image analysis. However, the color-based leaf segmentation can be significantly influenced by the lighting conditions and the green-looking objects present in the background, e.g., mosses and weeds.

Supervised leaf segmentation. Supervised leaf segmentation aims to segment leaf pixels in an image with a model trained on annotated image datasets. While some early works [13, 30] attempted to accomplish this task by learning the distributions of leaf and non-leaf pixels in the color space, the past few years have seen intensive use of methods based on deep neural networks in various computer vision tasks, with no exception for image-based plant phenotyping. Deep neural network architectures, such as U-Net [31] and Mask R-CNN [20], have been successfully used for category-level [32–34] or instance-level [17, 18, 21, 35] leaf segmentation. To harness the full potential of deep neural networks, it is essential to train the networks on large-scale high-quality annotated datasets. However, the expense of the specialized facilities and equipment for growing and monitoring individual plants, have substantially hindered the collection and annotation of large representative datasets required in training deep learning models for image-based plant phenotyping. To mitigate the data scarcity issue, Ward *et al.* [17, 18] employed domain randomization to generate synthetic arabidopsis leaf images. With a pool of "inspiration" leaves with leaf geometries, leaf textures, and backgrounds collected from existing annotated leaf image datasets, a synthetic plant is generated by randomizing the background and various plant parameters, e.g., leaf shapes and textures, sampled from the pool of inspiration leaves. A similar idea was proposed by Kuznichov *et al.* [21], where a synthetic image is generated by applying geometric transformations with random parameters to individual leaves segmented from real leaf images and pasting them in random (*naïve collage*) or logical and structured (*structured collage*) locations over a background image randomly selected from the CVPPP LSC dataset [12]. Some other works [23, 24, 36] resort to generative

adversarial networks (GANs) [37] to generate synthetic plant images. While effective in generating realistic synthetic plant images, all the above-mentioned methods are still highly reliant on large amounts of annotated images.

Self-supervised leaf segmentation. Self-supervised learning aims to automatically generate some kind of supervisory signal, e.g., pseudo labels, from unlabeled data to solve tasks that are typically targeted by supervised learning. As the supervisory signal is automatically generated from the data itself or its transformed versions, self-supervised learning does not rely on human labeled data and thus can be considered as a subset of unsupervised learning. To differentiate it from traditionally unsupervised learning (e.g., K-Means and fuzzy clustering), in this work, we use the term "self-supervised learning" to refer to the techniques that explicitly and automatically generate supervisory signals for typical supervised learning tasks such as classification and regression. Through solving pretext tasks [38, 39], self-supervised learning has been widely used to pre-train deep neural networks for learning visual representations that can be transferred to downstream tasks, e.g., image classification [40], object detection [41], and semantic segmentation [40]. There has also been a recent emergence of self-supervised methods that directly output class labels for image clustering and semantic segmentation without the use of a pretext task. For instance, Ji *et al.* [42] trained a deep neural network by maximizing the mutual information between the network outputs of an image and its augmented versions to predict the image-level and pixel-level semantic labels for image clustering and image segmentation, respectively. In a similar vein, some works attempted to learn pixel-level representations for semantic segmentation from different views [43] or object mask proposals [44] of the input image. Another line of self-supervised semantic segmentation [45, 46] constructs supervisory signal by grouping spatially adjacent pixels with similar features and iteratively updating the network parameters until semantic label assignment converges. However, despite the growing popularity in machine learning community, self-supervised learning is still fairly unexplored in image-based plant phenotyping and, more broadly, in agricultural technology, with a few exceptions that leverage self-supervised

(a) Greenhouse setup (b) Camera (c) Sample top-view image

Figure 1: Image acquisition setup for our Cannabis dataset.

methods for pre-training agricultural image classification model, e.g., [47].

3. Methodology

We first describe the unique opportunity we had to design and deploy a cannabis growth monitoring system in a real-world greenhouse environment. As shown in Fig. 1, cameras are mounted directly above the pots to collect top-view images at an hourly interval over the course of the whole growth cycle. Depending on the growth stage of the cannabis, the growing environment can be customized by controlling the temperature, humidity and lighting conditions to optimize the cannabis yield. We would like to particularly stress that the lighting schedule (i.e., exposure duration) and quality (i.e., light spectrum) can directly impact the transition between growth stages and ultimately affect the yield of the plant. Thus, a high-quality artificial lighting environment is critical for the effective indoor cultivation of cannabis. In such a scenario, measuring the leaf area index of plant canopy, which can be approached via category-level leaf segmentation, is practically more feasible than counting the number of leaves for growth monitoring due to the heavy leaf occlusions that often occur at the later growth stages of the cannabis. The key challenges for leaf segmentation in this typical scenario are twofold. First, while data can be collected round the clock automatically, annotating the collected data for training leaf segmentation algorithms is labor-intensive and error-prone. Second, the use of artificial grow lights poses a great challenge for many segmentation algorithms as it dramatically changes the appearance of the plant in the image. In this

work, we propose a self-supervised leaf segmentation framework that provides a promising way towards effective and generalizable leaf segmentation under complex lighting conditions without the need for annotated data. Fig. 2 shows the overview of our proposed framework, which mainly consists of three components: self-supervised color correction, self-supervised semantic segmentation, and color-based leaf segmentation. In what follows, we will delve into the details of the self-supervised semantic segmentation and color-based leaf segmentation for the images acquired under "natural" or "normal" lighting conditions. We then introduce the self-supervised color correction model for correcting the color of the images taken under "unnatural" artificial lights so that the color-corrected images can be segmented in the same way as for "natural" images.

3.1. Pseudo Label Generation for Self-Supervised Semantic Segmentation

At the core of mainstream self-supervised semantic segmentation approaches is the generation of supervisory signals, typically in the form of "pseudo labels" for samples of the same or different classes, by leveraging human prior knowledge lying in the data. Images and their augmentations are used for generating positive samples of the same class, while all other images are considered as negative samples. With the positive and negative samples generated from a large number of images, a convolutional neural network (CNN) can be trained to extract pixel-level embeddings or representations for predicting semantic labels. Taking a detour from this prevalent practice, we approach the self-supervision problem by letting the neural network itself determine whether two pixels (actually two local patches due to the spatial locality enforced by the convolutional operation) of the same image belong to the same class or not. The underlying assumption is that *semantically similar pixels should be mapped by an appropriately parameterized embedding learning network into representation embeddings that are close to each other in the embedding space, and therefore are more likely to be assigned with the same semantic label.*

Concretely, for a pixel-level embedding learning function $\Phi_\theta : \mathcal{X} \to \mathcal{Z}$ parameterized by a neural network with weights θ, it is expected to map two similar

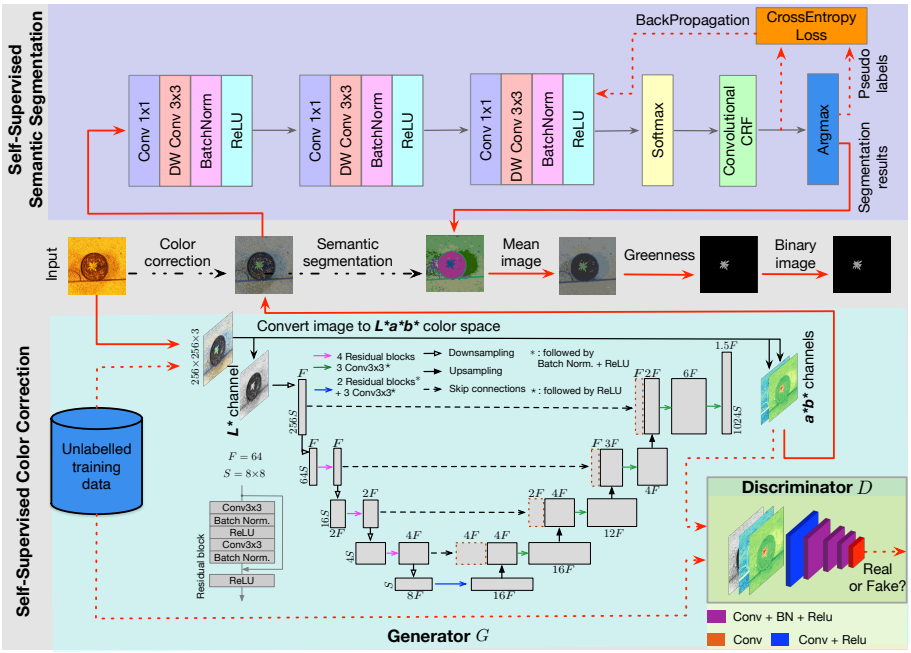

Figure 2: Overview of the proposed self-supervised leaf segmentation framework. It mainly consists of three components: self-supervised color correction, self-supervised semantic segmentation, and color-based leaf segmentation. The data flows during the training and testing phases are shown with red dashed and red solid lines, respectively. An image input is first passed through the trained color correction model to rectify the potential 'unnatural' color in the image. The color-corrected image is then input to the self-supervised semantic segmentation model to group the pixels of semantically similar objects, which will be jointly considered to identify the green leaf objects with our color-based leaf segmentation model.

image pixels $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ to two similar representation embeddings $\mathbf{z}, \mathbf{z}' \in \mathcal{Z}$. A shallow network with 2 or 3 convolutional layers is sufficient to extract discriminative features for our task of leaf segmentation, as shown in Fig. 2. Following [48], we replace the traditional 3×3 convolution with a 1×1 pointwise convolution and a 3×3 depthwise convolution, which reduces the number of parameters of the network and speeds up the embedding learning process. Similar to [42], we terminate the output of convolutional layers with a softmax layer, which allows us to model the uncertainty of the label assignment at the pixel i with a discrete probability distribution over K semantic labels, i.e., $\mathbf{z}_i = \Phi_\theta(\mathbf{x}_i) \in [0, 1]^K$.

As such, the supervisory signals can be generated by exploiting the information readily available in the discrete probability distributions. A straightforward way to obtain the "pseudo labels", as proposed in [45, 46], is to perform the *argmax* classification by assigning the index with the highest probability to each pixel, i.e., $l_i = \{j^\star | \mathbf{z}_{i,j^\star} = \max_j \mathbf{z}_{i,j}\}$, where l_i is the pseudo label for the pixel i and $\mathbf{z}_{i,j}$ is the jth element of $\mathbf{z}_i$. These pixel-level pseudo labels are, in turn, used in a supervised fashion to update the network parameters via backpropagation. This procedure is repeated iteratively until convergence or the maximum number of iterations is reached.

3.2. Challenges of Pseudo Label Refinement

As in supervised learning, the quality of the pseudo labels also has a significant impact on the performance of self-supervised learning approaches. However, the aforementioned way of pseudo label generation, in its primitive form, is prone to noisy labels possibly due to the intrinsic properties of convolutional neural networks, e.g., the sensitivity to small perturbations in the image [49] and the tendency to output blurry object boundaries [50]. Consequently, semantically similar pixels may be assigned with different labels, while pixels of different semantic objects are likely to be assigned with the same label. These problems are more prominent in the earlier stage of the iterative procedure when the network weights are primarily random values.

Two approaches have been pursued to impose additional constraints for the refinement of pseudo label assignment. The first approach [45] is to apply superpixel segmentation (e.g., SLIC [51]) beforehand and force the pixels in the same superpixel to have the same pseudo label. The second approach is to employ a spatial continuity loss [46] to encourage consistent pseudo label assignment for adjacent pixels. The drawback of the first approach is that superpixel segmentation itself is an ill-posed problem and the errors in superpixel segmentation, which often occur in object boundaries, may lead to inaccurate and misleading pseudo label assignment. Moreover, as the superpixel segmentation is only performed once prior to the iterative update of the network parameters, it does

not allow the local neighborhood information conveyed by superpixels to be updated in a dynamic way. Inevitably, little useful information can be provided for pseudo label refinement once the spatial consistency enforced by superpixels has been fulfilled. For the second approach, the spatial continuity loss encourages spatial consistency of label assignment by enforcing the extracted representation embeddings of adjacent pixels to be close to each other. Not only does it neglect the long-range dependencies between pixels, it also disregards the fact that adjacent pixels may belong to different semantic objects. Such a boundary-unaware label propagation may provide conflicting information for label refinement and often results in subtle segments within the same semantic object, which can be clearly observed in the visualization results presented in [46].

3.3. Fully-Connected CRFs for Structured Pseudo Label Refinement

In response to the above-mentioned limitations, we propose to integrate the fully connected conditional random field (CRF) [52] into the iterative label assignment procedure of our self-supervised semantic segmentation model. Given an image $\mathbf{X}$ consisting of N pixels, we model its segmentation as a random field defined over a set of variables $\boldsymbol{L} = \{l_1, l_2, ..., l_N\}$, where l_i represents the label assigned to the pixel i and can take any value from a set of K semantic labels $\mathcal{L} = \{1, 2, ..., K\}$. A conditional random field $(\mathbf{X}, \boldsymbol{L})$ can be characterized by a Gibbs distribution in the form of $P(\boldsymbol{L}{=}\boldsymbol{l}|\mathbf{X}) = \frac{1}{Z(\mathbf{X})}\exp(-E(\boldsymbol{l}|\mathbf{X}))$, where $E(\boldsymbol{l}|\mathbf{X})$ is the Gibbs energy of a labeling configuration $\boldsymbol{l} \in \mathcal{L}^N$ and $Z(\mathbf{X})$ is the partition function. For the fully connected CRF model in [52], the Gibbs energy is given by

$$E(\boldsymbol{l}|\mathbf{X}) = \sum_{i \leqslant N} \psi_u(l_i|\mathbf{X}) + \eta \sum_{i \neq j \leqslant N} \psi_p(l_i, l_j|\mathbf{X}), \tag{1}$$

where the unary potential $\psi_u(l_i|\mathbf{X})$ measures the cost of assigning label l_i to the pixel i and the pairwise potential $\psi_p(l_i, l_j|\mathbf{X})$ measures the cost of assigning labels l_i, l_j to pixels i, j simultaneously. η is a weighting factor adjusting the relative importance of the unary and pairwise potentials.

For our pseudo label refinement, we set the unary potential as $\psi_u(l_i|\mathbf{X}) = -\log z_{i,l_i}$, where z_{i,l_i} is the probability of assigning label l_i to pixel i as output by

the softmax layer of the embedding learning network Φ_θ. While for the pairwise potential $\psi_p(l_i, l_j|\mathbf{X})$, we adopt a Gaussian *appearance* kernel [52]:

$$\begin{aligned}\psi_p(l_i, l_j|\mathbf{X}) &= \mu(l_i, l_j)k(\mathbf{f}_i, \mathbf{f}_j) \\ &= \mu(l_i, l_j) \underbrace{\exp\left(-\frac{\|\mathbf{p}_i - \mathbf{p}_j\|_2^2}{2\sigma_\alpha^2} - \frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma_\beta^2}\right)}_{appearance\ kernel}.\end{aligned} \tag{2}$$

The label compatibility function $\mu(l_i, l_j)$ imposes a penalty when different labels l_i and l_j are assigned to adjacent pixels. While it is possible to specify different penalties for different pairs of labels or make $\mu(l_i, l_j)$ as parameters that can be learned from the data as in [52], it is unreasonable to do so in our self-supervised setting as the labels are randomly assigned for different images and there are no pre-determined semantic meanings for the labels. In other words, the leaf pixels in two different images may be represented by different labels. For this reason, we use the simple and most widely used Potts model given by $\mu(l_i, l_j) = [\![l_i \neq l_j]\!]$, where $[\![]\!]$ is the Iverson bracket.

The appearance kernel $k(\mathbf{f}_i, \mathbf{f}_j)$ in Eq. (2) depends on both pixel locations $(\mathbf{p}_i, \mathbf{p}_j)$ and the corresponding color vectors $(\mathbf{x}_i, \mathbf{x}_j)$ in the RGB color space. Intuitively, it tends to assign the same label for adjacent pixels with similar color, with the "scale" of spatial distance and color proximity controlled by the parameters σ_α and σ_β. As each pair of pixels i and j will contribute to the pairwise potential, regardless of their distance from each other, the fully connected CRF model allows to exploit long-range pixel dependencies for pseudo label refinement. Note that in the original model in [52], there is a *smoothness* kernel intended for removing small isolated regions. We discard it in our method as it will give rise to the chance of merging small, even though visually distinct, into the background, which could be detrimental for separating small plant leaves (e.g., at the seeding stage) from the background. Under such formalization, our pseudo label refinement for a given image $\mathbf{X}$ can be achieved by finding the most probable label assignment $\boldsymbol{l}^\star$ that gives the maximum a posteriori (MAP) labeling of the random field, i.e., $\boldsymbol{l}^\star = \arg\max_{\boldsymbol{l} \in \mathcal{L}^N} P(\boldsymbol{l}|\mathbf{X})$, or equivalently, the lowest Gibbs energy $E(\boldsymbol{l}^\star|\mathbf{X})$. However, CRFs are notoriously

Algorithm 1 Self-Supervised Semantic Segmentation

1: **for** $t = 1$ to T **do**
2: $\quad \{\mathbf{q}_i\}_{i=1}^N, \{\mathbf{z}_i\}_{i=1}^N \leftarrow \{\Phi_\theta(\mathbf{x}_i)\}_{i=1}^N$ ▷ Pixel-level softmax output
3: $\quad$ **repeat** m=5 **times**
4: $\quad\quad \{\tilde{q}_{i,l}\}_{i=1}^N \leftarrow \{\eta \sum_{j \neq i} k(\mathbf{f}_i, \mathbf{f}_j)\, q_{j,l}\}_{i=1}^N$ ▷ Message passing
5: $\quad\quad \{\hat{q}_{i,l}\}_{i=1}^N \leftarrow \{\sum_{l' \in \mathcal{L}} \mu(l, l')\, \tilde{q}_{i,l'}\}_{i=1}^N$ ▷ Compatibility transform
6: $\quad\quad \{\check{q}_{i,l}\}_{i=1}^N \leftarrow \{\hat{q}_{i,l} + \log z_{i,l}\}_{i=1}^N$ ▷ Adding unary potentials
7: $\quad\quad \{\mathbf{q}_i\}_{i=1}^N \leftarrow \{\text{Softmax}(\check{\mathbf{q}}_i)\}_{i=1}^N$ ▷ Softmax normalization
8: $\quad$ **end**
9: $\quad \{l_i\}_{i=1}^N \leftarrow \{\arg\max_j q_{i,j}\}_{i=1}^N$ ▷ Supervisory pseudo labels
10: $\quad \mathfrak{L} \leftarrow \text{CrossEntropyLoss}(\{\mathbf{q}_i, l_i\}_{i=1}^N)$ ▷ Cross-entropy loss
11: $\quad \Phi_\theta \leftarrow \text{Update}(\mathfrak{L}, \Phi_\theta)$ ▷ Network parameters update
12: **end for**
13: **return** $\{l_i\}_{i=1}^N$

hard to optimize [53, 54] and the exact maximization of $P(\boldsymbol{l}|\mathbf{X})$ is intractable even for low-resolution images. To circumvent this issue, a mean-field algorithm was proposed in [52] for approximate MAP marginal inference of $P(\boldsymbol{l}|\mathbf{X})$. The basic idea is to approximate the distribution $P(\boldsymbol{l}|\mathbf{X})$ with a simpler distribution $Q(\boldsymbol{l}|\mathbf{X})$ that can be expressed as a product of independent marginals. The details of the mean-field algorithm are summarized in Steps 2-8 of Algorithm 1, where the distribution $Q(\boldsymbol{l}|\mathbf{X})$ is initialized as the pixel-level softmax output $\{\mathbf{q}_i\}_{i=1}^N$ of the embedding learning network Φ_θ.

The efficient implementation of the mean-field algorithm is important for our label refinement as it may be executed T (a few hundreds) times until the neural network Φ_θ has been trained to extract meaningful embeddings for semantic segmentation. Fortunately, it was shown in [55] that all steps of the mean-filed algorithm (i.e., the Steps 2-8 of Algorithm 1) can be efficiently implemented on GPUs. Of particular note is the message passing (the Step 4 of Algorithm 1), which can be implemented as a Gaussian filter with the coefficients calculated using the Gaussian appearance kernel in Eq. (2). The fully connected

CRF, in its original form, allows for modeling the dependency between any pair of pixels in an image, resulting in a Gaussian filter that potentially spans the whole image. As suggested in [56], this issue can be circumvented by assuming that the label distributions of two pixels are conditionally independent if their Manhattan distance is greater than k. Such a conditional independence assumption allows to efficiently implement the message passing with a $k \times k$ convolutional filter while still retaining the capability and flexibility of modeling long-range pixel dependencies. As shown in Algorithm 1, the mean-filed algorithm is repeated $m = 5$ times, as suggested in [19, 56], to obtain the refined label assignment. Afterwards in the Steps 9-11 of Algorithm 1, the supervisory pseudo labels, generated by applying *argmax* classification to the refined label assignment distribution of each pixel, are used to calculate the multi-class cross-entropy loss for updating the embedding network with backpropagation. The entire procedure is repeated T times until the network Φ_θ is capable of extracting meaningful embeddings.

3.4. Color-Based Leaf Segmentation

Most existing semantic segmentation algorithms, including many self-supervised methods, require large-scale image datasets for training the network to group pixels into a pre-defined set of semantic classes. In contrast, our proposed self-supervised semantic segmentation algorithm learns to assign the same label to the semantically similar pixels with the self-contained information in a single image. While this precludes the use of external data, the side effect is that additional efforts are required to distinguish the leaves from other objects. Prior works [13, 25] have shown the potential of color-based features for leaf segmentation, albeit for images with homogeneous backgrounds or in the supervised setting. With the results output by our self-supervised semantic segmentation algorithm, we are allowed to jointly process similar pixels of the same semantic label and extract more reliable color information that is less susceptible to the cluttered backgrounds or the subtle changes in leaf pixels. Towards this end, we propose a leaf segmentation algorithm based on the "greenness" of the pixels.

Specifically, we first replace each pixel color $\mathbf{x}_i$ with the mean color of the pixels with the same label in its *connected* region, i.e., $\bar{\mathbf{x}}_i = \frac{1}{|\mathcal{X}_{l_i}|}\sum_{j\in\mathcal{X}_{l_i}}\mathbf{x}_j$, where $\mathcal{X}_{l_i}$ is the set of pixels with the label l_i in the *connected* region of pixel i. The use of *connected* regions enforces the calculation of the mean color is performed locally, preventing two remotely located objects in the image from influencing each other. Next, we convert the image from RGB to the HSV color space (i.e., $\mathbf{v}_i = \text{rgb2hsl}(\bar{\mathbf{x}}_i)$) and measure the "greenness" of each pixel with the following multivariate normal distribution in the HSV color space:

$$g(\mathbf{v}_i) = \frac{1}{\sqrt{(2\pi)^3\ det(\mathbf{\Sigma})}}\exp\Big(-\frac{1}{2}(\mathbf{v}_i-\boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(\mathbf{v}_i-\boldsymbol{\mu})\Big), \tag{3}$$

where $\boldsymbol{\mu}\in\mathbb{R}^3$ and $\mathbf{\Sigma}$ are the user-specified mean color vector and diagonal covariance matrix. Finally, a binary leaf segmentation mask is generated by applying the following thresholding operation:

$$\hat{u}_i = \begin{cases} 1, & \tilde{g}(\mathbf{v}_i) > \gamma_1 \ \text{ AND } \ \breve{g}(\mathbf{v}_i) > \gamma_2 \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

where

$$\begin{cases} \tilde{g}(\mathbf{v}_i) = \dfrac{g(\mathbf{v}_i)}{g(\boldsymbol{\mu})} \in [0,1] \\ \breve{g}(\mathbf{v}_i) = \dfrac{g(\mathbf{v}_i) - \min_i g(\mathbf{v}_i)}{\max_i g(\mathbf{v}_i) - \min_i g(\mathbf{v}_i)} \in [0,1] \end{cases} \tag{5}$$

$\tilde{g}(\mathbf{v}_i)$ and $\breve{g}(\mathbf{v}_i)$ measure the *absolute* and the *relative* "greenness", respectively. Intuitively, the *relative* greenness measures the relative degree of greenness of an object by comparing to the highest ($\max_i g(\mathbf{v}_i)$) and the lowest *absolute* greenness ($\min_i g(\mathbf{v}_i)$) in the same image. It allows us to select only the most green objects in an image, which could be particularly useful for reducing false positives when non-leaf but green-looking objects (e.g., mosses) appear in the background.

3.5. Self-Supervised Color Correction

The last building block of our leaf segmentation framework is the self-supervised color correction model. As shown in Fig. 2, our color correction

model follows the GAN-based pixel2pixel image translation network architecture [57], with a generator responsible for generating color-corrected images and a discriminator responsible for discriminating "real" images taken under good lighting conditions and "fake" images generated by the generator. The generator G, mainly consisting of a series of Convolution-BatchNorm-ReLU modules and residual blocks, progressively downsamples the input image to obtain high-level features and then gradually upsamples the features to generate the target images. To compensate for the low-level information lost due to downsampling operations, skip connections are used as "shortcuts" to allow for the direct information flow between the downsampling and upsampling branches. The discriminator D is a binary classifier and is constructed by simply stacking a few blocks of Convolution-BatchNorm-ReLU.

The training of the color correction model only involves unlabeled natural images captured under good lighting conditions. For each of n natural image in the training set $\{\mathbf{I}_i\}_{i=1}^n$, it is converted into the $L^*a^*b^*$ color space, with the lightness/grayscale values stored in the L^* channel $\{\mathbf{I}_i^{L^*}\}_{i=1}^n$ and the color values stored in the a^*b^* channels $\{\mathbf{I}_i^{a^*b^*}\}_{i=1}^n$. By taking the L^* channel as input and the a^*b^* channels as output, the generator is trained to recover the color channels from the grayscale channel image. Once the training is done, the generator is expected to take the L^* channel of a "color-corrupted" image, e.g., taken under artificial lights or poor weather conditions, and produce a natural-looking image as if it was taken under good lighting conditions to achieve the purpose of color correction. The training data for the Discriminator D is the "real" original images $\{\mathbf{I}_i\}_{i=1}^n$ and the "fake" images formed by concatenating the L^* images and the corresponding a^*b^* images generated by the generator, i.e., $\{G(\mathbf{I}_i^{L^*}), \mathbf{I}_i^{L^*}\}_{i=1}^n$. For the training loss $\mathfrak{L}(G, D)$, we use the combination of

Table 1: The details of our Cannabis (Cnbs) dataset and the CVPPP LSC dataset.

Dataset	Res. (pixels)	# training images	# test images (Natural)	# test images (Yellow)	# test images (Purple)	Plant species
Cnbs	768×768	300	40	40	40	Cannabis
A1	500×530	128	33	33	33	Arabidopsis
A2	530×565	31	9	9	9	Arabidopsis
A3	2448×2048	27	65	65	65	Tobacco
A4	441×441	624	168	168	168	Arabidopsis

GAN loss $\mathfrak{L}_{GAN}(G, D)$ and $L1$ loss $\mathfrak{L}_{L1}(G)$ balanced by a weighting factor λ:

$$\begin{aligned}\mathfrak{L}(G, D) &= \mathfrak{L}_{GAN}(G, D) + \lambda \mathfrak{L}_{L1}(G) \\ &= \sum_{i=1}^{n} \log(D(\mathbf{I}_i)) + \log(1 - D(\{G(\mathbf{I}_i^{L^*}), \mathbf{I}_i^{L^*}\})) \\ &+ \lambda \sum_{i=1}^{n} \|\mathbf{I}_i^{a^* b^*} - G(\mathbf{I}_i^{L^*})\|_1 \end{aligned} \tag{6}$$

The generator G and discriminator D are trained alternatively in an adversarial manner to obtain the final color correction model $G^\star = \arg\min_G \max_D \mathfrak{L}(G, D)$.

4. Experiments

4.1. Datasets

We conduct the experiments on two datasets: our Cannabis dataset and the Computer Vision Problem in Plant Phenotype (CVPPP) Leaf Segmentation Challenge (LSC) dataset [12]. Table 1 summarizes the details of these two datasets. With the image acquisition setup shown in Fig. 1, we collect our Cannabis dataset at different growth stages of cannabis plants under "Natural", "Yellow", and "Purple" lighting conditions, which are controlled by turning off or tuning the grow lights to yellow or purple color. We collect 300 images under the "Natural" lighting condition as the unlabeled training set for training our

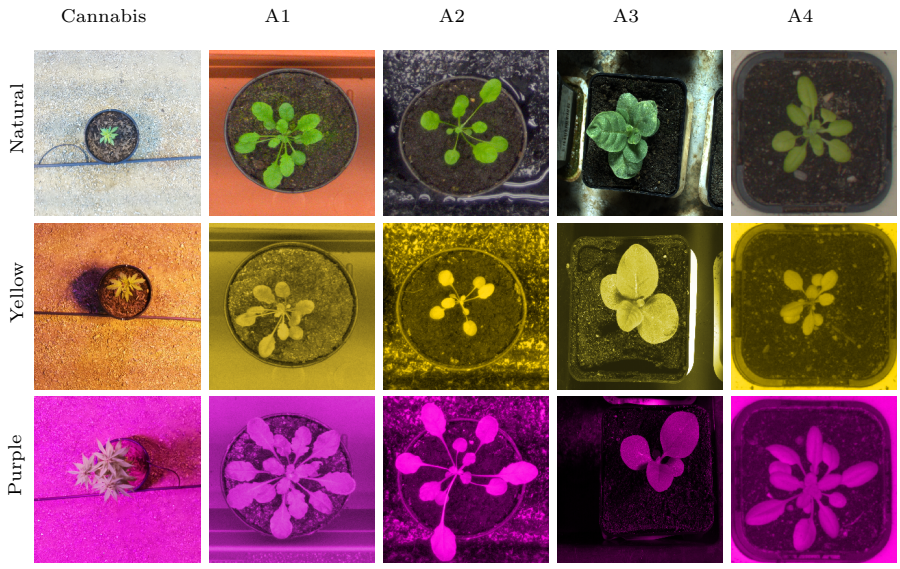

Figure 3: Examples of "Natural", "Yellow" and "Purple" plant images in our Cannabis dataset and the CVPPP LSC dataset.

color correction model. Besides, we also collect 40 images under each of the three lighting conditions and manually annotate the leaf segmentation masks for evaluating the leaf segmentation performance. The LSC dataset consists of 4 subsets: A1, A2, A3, and A4, with each subset containing the images of a different plant species. We refer to the original LSC dataset that was acquired under good lighting condition as "Natural". To simulate the yellow and purple lighting conditions for the LSC dataset, we generate two more versions, "Yellow" and "Purple", for each image by randomly adjusting the normalized hue value of each pixel in the ranges of $[0.13, 0.15]$ and $[0.83, 0.86]$, respectively. For both versions, we also randomly adjust the saturation and lightness values in the ranges of $[0.6, 1]$ and $[0.75, 1]$, respectively, to introduce more diversity to the generated images. Sample images under different lighting conditions in our Cannabis dataset and the LSC dataset are shown in Fig. 3. Note that the training images (without segmentation annotations) in both datasets are only used for the self-supervised training of the color correction model.

4.2. Training Setup

For our self-supervised semantic segmentation model, it does not require any external training data and iteratively updates the segmentation result by resorting to the self-contained information in the same image. For all the experiments, we set the number of semantic labels $K = 64$, the maximum iteration number $T = 300$ (Algorithm 1), and the weighting factor $\eta = 10$ for the pairwise potential in Eq. (1). While for the color-based leaf segmentation described in Section 3.4, we empirically set the mean vector $\boldsymbol{\mu} = [0.3, 0.6, 0.8]$ and covariance matrix $\boldsymbol{\Sigma} = \mathrm{diag}([0.1, 0.3, 0.5])$ of the greenness measurement distribution in Eq. (3). The thresholds for the absolute and relative greenness in Eq. (4) are set to $\gamma_1 = 0.2$ and $\gamma_2 = 0.5$, respectively.

For the self-supervised color correction model, we train two models separately on our Cannabis dataset and the LSC dataset. We divide each image in the training set (without annotation) into blocks of 256×256 px with 50% overlapping to alleviate the data scarcity issue. We set the weighting factor $\lambda = 100$ in Eq. (6) and train the generator G and the discriminator D alternatively for 50 epoches with batches of size 16. The model weights are updated through Adam optimizer with a learning rate of 0.0002. Due to the large disparity in model complexity between the generator G and the discriminator D in Fig. 2, training them from the same starting point could easily lead to the earlier convergence of the discriminator. To balance the learning speed of the generator and the discriminator, we pre-train the generator with the $L1$ loss for 20 epoches before alternatively training the two networks.

4.3. Evaluation Metrics

We evaluate the leaf segmentation performance with the commonly used metric of Foreground-Background Dice (FBD) coefficient, which is calculated as FBD=$\frac{1}{n}\sum_{i=1}^{n}\frac{2TP_i}{2TP_i+FP_i+FN_i} \in [0,1]$ where TP_i, FP_i, and FN_i are, respectively, the numbers of true positive, false positive, and false negative pixels of the ith image. For the performance evaluation of color correction, we use the metrics of Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM).

A higher PSNR/SSIM value indicates better color correction performance. We tailor the calculation of these two metrics for our color correction task by converting the images to the YUV color space and only measuring PSNR and SSIM of the U and V color channels. Besides, we also employ the Learned Perceptual Image Patch Similarity (LPIPS) metric [58] to measure the perceptual similarity between the original and the color-corrected images. LPIPS between two images is calculated as the distance between their deep embeddings obtained from classic neural networks, e.g., VGG and AlexNet, trained on ImageNet [59].

4.4. Ablation Studies

In this subsection, we will examine the influence of different components of our self-supervised leaf segmentation framework using the LSC dataset.

Leaf segmentation. We first investigate the effects of the smooth term of the fully-connected CRF [52], the absolute greenness and the relative greenness on the performance of leaf segmentation. The results on the "Natural" *testing set* of the LSC dataset are shown in Table 2. The first observation is that including the smooth term negatively contributes to the segmentation results as it gives rise to the chance of removing small leaf regions. The second observation is that using the relative greenness improves the segmentation result on images with green background, e.g., 94.7% vs 90.4% on the subset 'A1' where some images contain green moss in the background. Note that because the relative greenness is a "relative" measurement, the non-green regions in an image may have high relative greenness if no green objects present in the image. For this reason, we jointly use the absolute and the relative greenness to reduce false positives caused by only using the relative greenness.

Color correction. For our color correction model, we investigate the effects of three training strategies, including 1) **Separated**: training on 4 subsets 'A1', 'A2', 'A3', and 'A4', separately, 2) **Combined**: training on the combined dataset of the 4 subsets, and 3) **Augmented**: training with data augmentations including random flipping, sharpness adjustment, Gaussian blurring, affine transform, and color jittering, to enhance the size and quality of the dataset.

Table 2: Analysis of the components of the proposed leaf segmentation algorithm. The highest FBD coefficient is shown in **bold**. The values in parenthesis are obtained on the subset 'A1'.

Smoothness term	Absolute greenness	Relative greenness	FBD(%)↑
✔	✔	✔	94.2
✗	✗	✔	94.6 (A1: 94.7)
✗	✔	✗	95.1 (A1: 90.4)
✗	✔	✔	**95.6** (A1: 94.7)

Note that for the color jittering augmentation, the hue value of each image remains unchanged to avoid unintended impacts on the color correction task. We train the color correction model on the "Natural" (i.e., original) *training set* of the LSC dataset with different training strategies and test the performance on the "Yellow" and "Purple" *testing sets* of the LSC dataset. The metrics averaged over 4 subsets (A1, A2, A3, and A4) on the 'Yellow" and "Purple" *testing sets* are reported in Table 3. We can see that data augmentations are effective in boosting the color correction performance, while combining the images from different subsets for training provides very limited benefit or may even worsen the performance, e.g., in the case of no augmentations. The worse performance on the combined dataset is probably due to the mutual interference between the notably different image backgrounds and plant traits in different subsets.

4.5. Performance Analyses

We compare our proposed self-supervised leaf segmentation framework with a wide range of methods covering the categories of unsupervised, supervised and self-supervised methods. The segmentation results on the "Natural" *testing sets* of the LSC dataset and our Cannabis dataset are shown in Table 4 and the visualization results on some example leaf images are provided in Fig. 4.

For unsupervised leaf segmentation methods, EM [25] performs poorly on both datasets as it assumes that the foreground and the background pixels can be modeled with two well-separated Gaussian distributions in the HSV color space, which does not hold for the images in the LSC dataset and our Cannabis

Table 3: Analysis of different training strategies for the color correction model. '↑' and '↓' respectively indicate that higher or lower values represent better performance. The best results are highlighted with **bold yellow** (for the "Yellow" testing set) and **bold purple** (for the "Purple" testing set) colors.

Separated	Combined	Augmented	PSNR[dB]↑	SSIM↑	LPIPS↓
✔	✗	✗	31.48(Y')	**0.841**(Y')	0.431(Y')
			33.43(P')	0.860(P')	0.244(P')
✔	✗	✔	31.33(Y')	0.836(Y')	**0.426**(Y')
			34.88(P')	**0.872**(P')	0.239(P')
✗	✔	✗	29.54(Y')	0.831(Y')	0.469(Y')
			32.46(P')	0.862(P')	0.263(P')
✗	✔	✔	**32.22**(Y')	0.831(Y')	0.427(Y')
			34.42(P')	0.860(P')	**0.237**(P')

Table 4: Leaf segmentation results in terms of FBD(%) on the "Natural" testing sets of the LSC dataset and our Cannabis dataset for unsupervised (EM [25], MCS [60], and Nottingham [14]), supervised (DC [16], SYN [17], and UPG [18]), and self-supervised (SSSLIC [45], SSCL [46], and our proposed SSCRF) methods. '⋆' indicates that the pre-trained model is fine-tuned on the *training set* of the LSC dataset. The highest FBD coefficient on each dataset/subset is highlighted in **bold**.

Dataset	Unsupervised			Supervised				Self-supervised		
	EM	MCS	Nott.	DC⋆	SYN	UPG	UPG⋆	SSSLIC	SSCL	SSCRF
Cnbs	16.1	70.6	90.1	–	62.2	23.0	–	80.7	87.8	**94.8**
A1	38.5	73.6	95.3	93.3	90.3	49.2	90.4	91.5	94.3	**94.7**
A2	65.6	80.4	**93.0**	80.3	79.3	30.8	91.0	55.8	82.4	92.0
A3	34.6	39.2	90.7	68.4	72.0	36.4	92.6	91.7	93.9	**95.2**
A4	50.2	79.2	90.2	74.7	76.8	26.4	93.2	76.2	84.7	**96.1**

Table 5: Cross-dataset performance evaluation in terms of FBD(%) on the "Natural" set of our Cannabis dataset for DC⋆[16] and UPG⋆[18].

Fine-tuning dataset	DC⋆[16]				UPG⋆[18]			
	A1	A2	A3	A4	A1	A2	A3	A4
FBD(%)	42.6	81.3	76.5	66.5	74.2	81.6	53.2	83.2

dataset. MCS [60] incorporates multiple cues, including color, texture, shape, and structure (i.e., leaf vein) information, to facilitate the leaf segmentation in complex backgrounds. However, it lacks the capability of discriminating the leaf from the scattered moss regions and is sensitive to illumination changes, leading to the poor segmentation performance on the subsets 'A1' and 'A3' of the LSC dataset. Nottingham [14] first over-segments the image in the $L^*a^*b^*$ color space using the SLIC [51] superpixel algorithm and extracts leaf regions with a simple seeded region growing algorithm in the superpixel space. Despite its great performance, we found that due to the pixel intensity variations among the superpixels within the leaf regions, it is tricky to select an appropriate threshold for separating the leaf regions from the background. Different from Nottingham [14], our proposed method performs leaf segmentation at a higher granularity level of semantic objects, thus reducing the intensity inhomogeneity within leaf and non-leaf regions and providing greater flexibility in threshold selection.

We investigate several supervised leaf segmentation methods based on two mainstream segmentation networks, U-Net [31] and Mask-RCNN [20]. By formulating instance-level segmentation as a coloring problem with a fixed number of colors, Deep Coloring (DC) [16] allows to train a semantic segmentation network based on U-Net [31] for instance-level segmentation with standard semantic segmentation objectives. With the pre-trained model provided by the authors of [16], we fine-tune it on each subset of the *training set* of the LSC dataset and test the fine-tuned model on the corresponding subset of the *testing set* of the LSC dataset. As the first color is reserved to represent the background, we conveniently extract the first network output channel as the leaf segmentation

mask. From the visualization results in Fig. 3, we can see that DC [16] gives reasonably good leaf segmentation performance, but it tends to mis-identify salient non-leaf regions in the background as leaf regions. SYN [17] and UPG [18] are methods based on Mask-RCNN [20] and make use of large-scale synthetic training data to achieve state-of-the-art leaf segmentation performance. For both methods, we use the union of the instance segmentation mask as the final leaf segmentation result. For SYN [17], we only report the results obtained with the pre-trained model as the implementation of model training has been removed in the source code published by the authors. While for UPG [18], we report the results obtained with the pre-trained model and the model fine-tuned on each subset of the *training set* of the LSC dataset. We use '$\star$' to indicate that a model has been fine-tuned on target datasets. Not surprisingly, UPG* outperforms UPG and SYN by a wide margin, which implies that fine-tuning the pre-trained model on annotated target datasets is critical for achieving the best possible segmentation performance. The generalization gap of deep learning models across different datasets becomes more evident in Table 5, where the highest FBD is only around 83% if the model fined-tuned on the LSC dataset is tested on our Cannabis dataset. These results re-confirm the difficult of training models that are generalizable across different plant species without fine-tuning on target datasets, which highlights the necessity and importance of developing self-supervised leaf segmentation methods.

As for the self-supervised segmentation methods, we evaluate three most related methods, SSSLIC [45] based on the SLIC [51] superpixel algorithm, SSCL [46] based on the continuity loss, and our proposed self-supervised segmentation method based on the fully-connected CRF model (SSCRF). We set the superpixel number to 10000 for SSSLIC and the weighting factor of the continuity loss to 5 for SSCL in our experiments. With the semantic segmentation results output by these three methods, we apply the same color-based method to obtain the final leaf segmentation results. We can see from Table 4 that our proposed SSCRF not only consistently outperforms the other two self-supervised methods but also achieves overall better performance than the state-of-the-art

Table 6: Leaf segmentation results in terms of FBD(%) on the "Yellow" testing sets of the LSC dataset and our Cannabis dataset. '⋆' indicates that the pre-trained model is fine-tuned on the *training set* of the LSC dataset.

Dataset	Unsupervised			Supervised				Self-supervised		
	EM	MCS	Nott.	DC⋆	SYN	UPG	UPG⋆	SSSLIC	SSCL	SSCRF
Cnbs	13.8	81.6	86.8	–	62.3	28.2	–	76.3	82.1	**87.1**
A1	38.7	73.6	87.8	87.4	88.8	59.3	**89.3**	87.2	85.0	88.7
A2	56.3	79.1	88.2	77.2	87.0	12.5	90.7	56.7	77.0	**92.5**
A3	27.4	19.1	74.1	65.0	67.5	26.4	90.3	91.3	91.6	**93.9**
A4	51.8	80.0	89.1	63.0	77.3	35.4	91.9	77.6	83.8	**92.3**

unsupervised and supervised methods. Further investigation on the visualization results in Fig. 3 shows that SSSLIC and SSCL tend to merge small leaves into the background, thus leading to the significant performance decline on the subsets 'A2' and 'A4' of the LSC dataset. The inferior performance of SSSLIC and SSCL can be attributed to the fact that, they only assign the same label to spatially adjacent pixels but lack an effective mechanism to prevent the occurrence of assigning the same label to distinctly different pixels/superpixels. While in our proposed SSCRF, such mechanism is realized via dynamically modeling the pairwise pixel affinities and penalizing inappropriate label assignments to neighboring pixels with large color differences.

To evaluate the performance of the proposed color correction model, we apply color correction to the images in the "Yellow" and "Purple" *testing sets* of the LSC dataset and our Cannabis dataset with the color correction models trained on the corresponding "Natural" *training sets*. We then repeat the above leaf segmentation experiments for all compared methods on the color-corrected images of the "Yellow" and "Purple" *testing sets*. The quantitative results are reported in Table 6 and Table 7, while some qualitative results can be found in Fig. 5, where we also show the color-corrected images in the second column

Table 7: Leaf segmentation results in terms of FBD(%) on the "Purple" testing sets of the LSC dataset and our Cannabis dataset. '⋆' indicates that the pre-trained model is fine-tuned on the *training set* of the LSC dataset.

Dataset	Unsupervised			Supervised				Self-supervised		
	EM	MCS	Nott.	DC⋆	SYN	UPG	UPG⋆	SSSLIC	SSCL	SSCRF
Cnbs	28.2	82.2	**84.8**	–	51.3	32.4	–	75.7	80.7	83.9
A1	37.3	72.9	87.8	91.7	90.0	48.5	89.7	91.4	92.5	**94.7**
A2	61.3	77.4	88.2	80.7	77.2	24.9	88.7	63.2	69.6	**92.6**
A3	22.9	18.7	81.6	68.2	67.1	45.4	89.9	87.3	91.9	**94.8**
A4	56.3	74.4	82.6	75.5	77.4	28.7	**88.4**	71.3	74.8	83.8

to visualize the color correction performance. We exclude EM [25] and UPG [18] in Fig. 5 because of their poor performance. We would like to make a few remarks for these results: 1) While trained on the same "Natural" *training sets*, the color correction model exhibits somewhat performance variations on images taken under different lighting conditions. It is generally easier to correct color for "Yellow" images than "Purple" images, probably because the yellow color is statistically distributed closer to the green color in the $L^*a^*b^*$ color space. 2) For the leaf segmentation task, it is important to include in the training set the images collected at various growth stages covering the whole life cycle of plants. In the training set of our Cannabis dataset, the majority of the images were collected at early growth stages including 72 images that only contain empty plant pots, which, to some extent, compromise the color correction performance on our Cannabis dataset. 3) Our proposed self-supervised leaf segmentation method still achieves overall better performance than other methods on the color-corrected images across different datasets, which highlights the potential of our method in achieving effective and generalizable leaf segmentation.

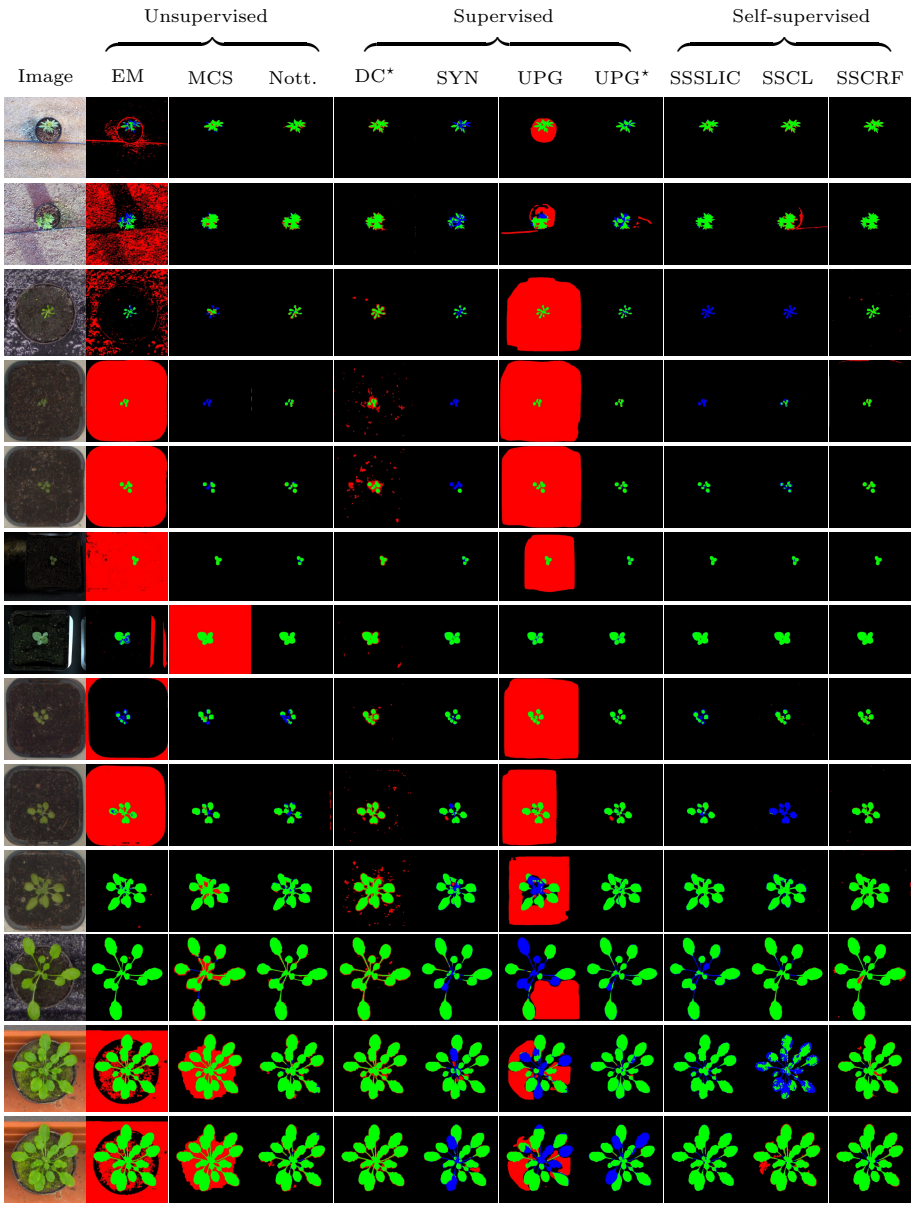

Figure 4: Example leaf segmentation results. For the results of DC*[16] and UPG*[18] on the Cannabis dataset, we show the visualization results obtained with the models fine-tuned on the subsets 'A2' and 'A4', respectively, which offer the highest achievable FBD(%) metrics according to Table 5. Color coding: *green*: detected leaf regions (true positives); *red*: detected non-leaf regions (false positives); *blue*: mis-detected leaf regions (false negatives).

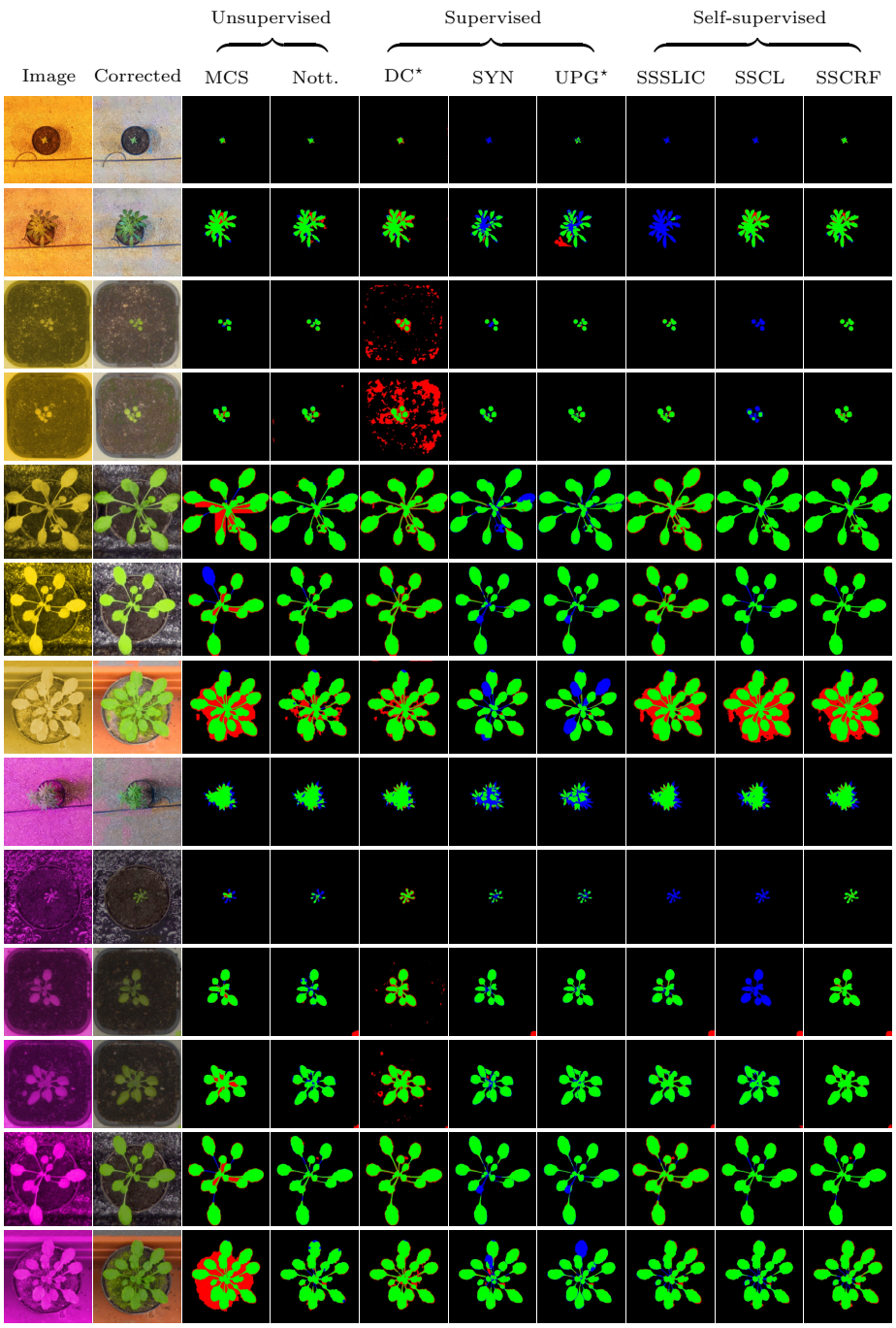

Figure 5: Example leaf segmentation results on "Yellow" and "Purple" images. Color coding: *green*: detected leaf regions (true positives); *red*: detected non-leaf regions (false positives); *blue*: mis-detected leaf regions (false negatives).

5. Conclusion and Future Work

In this work, we presented a self-supervised leaf segmentation framework that is capable of segmenting leaf regions from the background under complex illumination conditions without annotated training data. Comprehensive experiments on the CVPPP LSC dataset and our Cannabis dataset demonstrated that the proposed method achieves state-of-the-art performance. Despite its effectiveness in segmenting leaf regions under varying lighting conditions and the generalizability across different plant species, there is still some room for improvement. In the proposed self-supervised semantic segmentation model, the pixel-wise label assignment is updated and refined in an iterative manner, which may require hundreds of iterations to obtain sensible results. For an iteration number of T=300, the segmentation takes 20-30 seconds for an image of size 512×512 pixels on our desktop PC with a Nvidia GTX 2080Ti 11GB GPU. While this meets the requirements of our plant growth monitoring project, we will seek to improve the efficiency of our method by using heuristic early stopping criteria and initializing the pixel-level embeddings learning network with pre-trained weights. Given the promising results of our self-supervised semantic segmentation, another interesting line of research for future work is to explore the possibility of self-supervised instance-level leaf segmentation by bridging the gap between semantic segmentation and instance segmentation.

References

[1] A. Walter, F. Liebisch, A. Hund, Plant phenotyping: from bean weighing to image analysis, Plant Methods 11 (1) (2015) 1–11.

[2] A. Walter, U. Schurr, Dynamics of leaf and root growth: endogenous control versus environmental impact, Annals of Botany 95 (6) (2005) 891–900.

[3] A. J. Monforte, A. Diaz, A. Caño-Delgado, E. Van Der Knaap, The genetic basis of fruit morphology in horticultural crops: lessons from tomato and melon, Journal of Experimental Botany 65 (16) (2013) 4625–4637.

[4] S. Arvidsson, P. Pérez-Rodríguez, B. Mueller-Roeber, A growth phenotyping pipeline for arabidopsis thaliana integrating image analysis and rosette area mod-

eling for robust quantification of genotype effects, New Phytologist 191 (3) (2011) 895–907.

[5] W.-S. Lee, V. Alchanatis, C. Yang, M. Hirafuji, D. Moshou, C. Li, Sensing technologies for precision specialty crop production, Computers and Electronics in Agriculture 74 (1) (2010) 2–33.

[6] V. Saiz-Rubio, F. Rovira-Más, From smart farming towards agriculture 5.0: A review on crop data management, Agronomy 10 (2) (2020) 207.

[7] A. M. Mutka, R. S. Bart, Image-based phenotyping of plant disease symptoms, Frontiers in Plant Science 5 (2015) 734.

[8] A. M. Mutka, S. J. Fentress, J. W. Sher, J. C. Berry, C. Pretz, D. A. Nusinow, R. Bart, Quantitative, image-based phenotyping methods provide insight into spatial and temporal dimensions of plant disease, Plant physiology 172 (2) (2016) 650–660.

[9] S. P. Ojolo, S. Cao, S. Priyadarshani, W. Li, M. Yan, M. Aslam, H. Zhao, Y. Qin, Regulation of plant growth and development: a review from a chromatin remodeling perspective, Frontiers in Plant Science 9 (2018) 1232.

[10] C. Li, R. Adhikari, Y. Yao, A. G. Miller, K. Kalbaugh, D. Li, K. Nemali, Measuring plant growth characteristics using smartphone based image analysis technique in controlled environment agriculture, Computers and Electronics in Agriculture 168 (2020) 105123.

[11] T. Van Klompenburg, A. Kassahun, C. Catal, Crop yield prediction using machine learning: A systematic literature review, Computers and Electronics in Agriculture 177 (2020) 105709.

[12] M. Minervini, A. Fischbach, H. Scharr, S. A. Tsaftaris, Finely-grained annotated datasets for image-based plant phenotyping, Pattern recognition letters 81 (2016) 80–89.

[13] J.-M. Pape, C. Klukas, 3-d histogram-based segmentation and leaf detection for rosette plants, in: European Conference on Computer Vision, Springer, 2014, pp. 61–74.

[14] H. Scharr, M. Minervini, A. P. French, C. Klukas, D. M. Kramer, X. Liu, I. Luengo, J.-M. Pape, G. Polder, D. Vukadinovic, et al., Leaf segmentation in plant phenotyping: a collation study, Machine Vision and Applications 27 (4) (2016) 585–606.

[15] J.-M. Pape, C. Klukas, Utilizing machine learning approaches to improve the prediction of leaf counts and individual leaf segmentation of rosette plant images, Computer Vision Problems in Plant Phenotyping (CVPPP) (2015) 1–12.

[16] V. Kulikov, V. Yurchenko, V. Lempitsky, Instance segmentation by deep coloring, arXiv preprint arXiv:1807.10007 (2018).

[17] D. Ward, P. Moghadam, N. Hudson, Deep leaf segmentation using synthetic data, arXiv preprint arXiv:1807.10931 (2018).

[18] D. Ward, P. Moghadam, Scalable learning for bridging the species gap in image-based plant phenotyping, Computer Vision and Image Understanding 197 (2020) 103009.

[19] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer Assisted Intervention, Springer, 2015, pp. 234–241.

[20] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: IEEE International Conference on Computer Vision, 2017, pp. 2961–2969.

[21] D. Kuznichov, A. Zvirin, Y. Honen, R. Kimmel, Data augmentation for leaf segmentation and counting tasks in rosette plants, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2019, pp. 1–10.

[22] J. Ubbens, M. Cieslak, P. Prusinkiewicz, I. Stavness, The use of plant models in deep learning: an application to leaf counting in rosette plants, Plant Methods 14 (1) (2018) 1–10.

[23] M. Valerio Giuffrida, H. Scharr, S. A. Tsaftaris, Arigan: Synthetic arabidopsis plants using generative adversarial network, in: IEEE International Conference on Computer Vision Workshops, 2017, pp. 2064–2071.

[24] Y. Zhu, M. Aoun, M. Krijn, J. Vanschoren, H. T. Campus, Data augmentation using conditional generative adversarial networks for leaf counting in arabidopsis plants., in: British Machine Vision Conference, 2018, p. 324.

[25] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. C. Lopez, J. V. Soares, Leafsnap: A computer vision system for automatic plant species identification, in: European Conference on Computer Vision, Springer, 2012, pp. 502–516.

[26] S. Zhang, Z. You, X. Wu, Plant disease leaf image segmentation based on super-

pixel clustering and em algorithm, Neural Computing and Applications 31 (2) (2019) 1225–1232.

[27] X. Bai, X. Li, Z. Fu, X. Lv, L. Zhang, A fuzzy clustering segmentation method based on neighborhood grayscale information for defining cucumber leaf spot disease images, Computers and Electronics in Agriculture 136 (2017) 157–165.

[28] G. Cerutti, L. Tougne, J. Mille, A. Vacavant, D. Coquin, Understanding leaves in natural images–a model-based approach for tree species identification, Computer Vision and Image Understanding 117 (10) (2013) 1482–1501.

[29] K. Zou, L. Ge, C. Zhang, T. Yuan, W. Li, Broccoli seedling segmentation based on support vector machine combined with color texture features, IEEE Access 7 (2019) 168565–168574.

[30] L. Zheng, D. Shi, J. Zhang, Segmentation of green vegetation of crop canopy images based on mean shift and fisher linear discriminant, Pattern Recognition Letters 31 (9) (2010) 920–925.

[31] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer Assisted Intervention, Springer, 2015, pp. 234–241.

[32] H. S. Baweja, T. Parhar, O. Mirbod, S. Nuske, Stalknet: A deep learning pipeline for high-throughput measurement of plant stalk count and stalk width, in: Field and Service Robotics, Springer, 2018, pp. 271–284.

[33] M. Fawakherji, A. Youssef, D. Bloisi, A. Pretto, D. Nardi, Crop and weeds classification for precision agriculture using context-independent pixel-wise segmentation, in: IEEE International Conference on Robotic Computing, IEEE, 2019, pp. 146–152.

[34] M.-D. Yang, H.-H. Tseng, Y.-C. Hsu, H. P. Tsai, Semantic segmentation using deep learning with vegetation indices for rice lodging identification in multi-date uav visible images, Remote Sensing 12 (4) (2020) 633.

[35] K. Yang, W. Zhong, F. Li, Leaf segmentation and classification with a complicated background using deep learning, Agronomy 10 (11) (2020) 1721.

[36] R. Barth, J. Hemming, E. J. Van Henten, Optimising realism of synthetic images using cycle generative adversarial networks for improved part segmentation, Computers and Electronics in Agriculture 173 (2020) 105378.

[37] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair,

A. Courville, Y. Bengio, Generative adversarial nets, Advances in Neural Information Processing Systems 27 (2014).

[38] G. Larsson, M. Maire, G. Shakhnarovich, Colorization as a proxy task for visual understanding, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6874–6883.

[39] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, A. A. Efros, Context encoders: Feature learning by inpainting, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2536–2544.

[40] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, L. Van Gool, Scan: Learning to classify images without labels, in: European Conference on Computer Vision, Springer, 2020, pp. 268–285.

[41] I. Misra, L. v. d. Maaten, Self-supervised learning of pretext-invariant representations, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 6707–6717.

[42] X. Ji, J. F. Henriques, A. Vedaldi, Invariant information clustering for unsupervised image classification and segmentation, in: IEEE/CVF International Conference on Computer Vision, 2019, pp. 9865–9874.

[43] Y. Ouali, C. Hudelot, M. Tami, Autoregressive unsupervised image segmentation, in: European Conference on Computer Vision, Springer, 2020, pp. 142–158.

[44] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, L. Van Gool, Unsupervised semantic segmentation by contrasting object mask proposals, arXiv preprint arXiv:2102.06191 (2021).

[45] A. Kanezaki, Unsupervised image segmentation by backpropagation, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2018, pp. 1543–1547.

[46] W. Kim, A. Kanezaki, M. Tanaka, Unsupervised learning of image segmentation based on differentiable feature clustering, IEEE Transactions on Image Processing 29 (2020) 8055–8068.

[47] R. Güldenring, L. Nalpantidis, Self-supervised contrastive learning on agricultural images, Computers and Electronics in Agriculture 191 (2021) 106510.

[48] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861 (2017).

[49] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, arXiv preprint arXiv:1312.6199 (2013).

[50] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille, Semantic image segmentation with deep convolutional nets and fully connected crfs, arXiv preprint arXiv:1412.7062 (2014).

[51] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Süsstrunk, Slic superpixels compared to state-of-the-art superpixel methods, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (11) (2012) 2274–2282.

[52] P. Krähenbühl, V. Koltun, Efficient inference in fully connected crfs with gaussian edge potentials, Advances in Neural Information Processing Systems 24 (2011) 109–117.

[53] R. Wilson, C.-T. Li, A class of discrete multiresolution random fields and its application to image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (1) (2003) 42–56.

[54] C.-T. Li, Y. Yuan, R. Wilson, An unsupervised conditional random fields approach for clustering gene expression time series, Bioinformatics 24 (21) (2008) 2467–2473.

[55] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, P. H. Torr, Conditional random fields as recurrent neural networks, in: IEEE International Conference on Computer Vision, 2015, pp. 1529–1537.

[56] M. T. Teichmann, R. Cipolla, Convolutional crfs for semantic segmentation, arXiv preprint arXiv:1805.04777 (2018).

[57] P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros, Image-to-image translation with conditional adversarial networks, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1125–1134.

[58] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, O. Wang, The unreasonable effectiveness of deep features as a perceptual metric, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 586–595.

[59] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, International Journal of Computer Vision 115 (3) (2015) 211–252.

[60] N. Anantrasirichai, S. Hannuna, N. Canagarajah, Automatic leaf extraction from outdoor images, arXiv preprint arXiv:1709.06437 (2017).