Self-supervised Deep Learning for Flower Image Segmentation

Conference Paper · November 2020

DOI: 10.1109/IIT50501.2020.9298979

Sudipan Saha
IIT Delhi

Biplab Banerjee
Indian Institute of Technology Bombay

Sumedh Pendurkar
Texas A&M University

Self-supervised Deep Learning for Flower Image Segmentation

Sudipan Saha
Fondazione Bruno Kessler
Trento, Italy
saha@fbk.eu

Nasrullah Sheikh
IBM Research - Almaden
California, USA
nasrullah.sheikh@ibm.com

Biplab Banerjee
Indian Institute of Technology Bombay
Mumbai, India
bbanerjee@iitb.ac.in

Sumedh Pendurkar
Texas A&M University
Texas, USA
sumedhpendurkar@tamu.edu

Abstract—**Segmentation plays an important role in image-based plant phenotyping applications. Deep learning has led to a dramatic improvement in segmentation performance. Most deep learning-based methods are supervised and require abundant application-specific training data. Considering the wide range of plant phenotyping applications, such data may not be always available. To mitigate this problem, we introduce a segmentation method that exploits the power of deep learning without using any prior training. In this paper, we specifically focus on flower segmentation. Recurrence of information inside a flower image is used to train an image-specific deep network that is subsequently used for segmentation. The proposed method is self-supervised as it exploits the internal statistics of input image without using any prior labeled data. To the best of our knowledge, this is the first unsupervised deep learning-based method proposed for single-image flower segmentation.**

Index Terms—**Image segmentation, self-supervised learning, unsupervised deep learning.**

I. Introduction

Image-based plant phenotyping is an important application of computer vision in agriculture [1]. It provides an inexpensive alternative to manual phenotyping [2]. Recently, machine learning and deep learning based plant phenotyping has gained attention of the research community [1] [2] [3] [4]. Deep learning-based plant phenotyping methods are generally supervised [4]. Such methods require a significantly large training dataset to effectively capture the plant traits. There are few works based on transfer learning [5] and those which attempt to reduce the number of parameters in the deep network [2].

Segmentation is a key task in image-based plant phenotyping and is often used as an intermediate step for other tasks [6]. Deep learning-based segmentation methods are generally supervised [7] [8] [9] and thus they need a significantly large training dataset having pixel-level labels. They exploit training pixels to learn a deep classifier model and afterward use it to get segmentation labels. However, obtaining labeled data (with pixel-level label) may not be possible for all applications, especially while covering a large variety of plants.

A number of alternatives exist in the deep learning literature to circumvent the absence of labeled samples [5] [10] [11]. Noteworthy among them are the generative adversarial network (GAN) [12], [13] based methods. However, GAN based methods [14] require a large amount of unlabeled training data to capture the underlying distribution of the data. In real-life applications, the assumption of the presence of a large/massive amount of unlabeled data is often not true.

To overcome the aforementioned difficulty in collecting labeled/unlabeled large amounts of training data, single-image methods have been proposed [15] that exploit the statistics of a single image and potential of deep learning without relying on any prior training example. Based on this, methods have been proposed for single-image deep segmentation [16] [17]. We take inspiration from them to propose a method for single-image deep unsupervised/self-supervised method for plant phenotyping applications. In particular, we focus on automatically segmenting/detecting flowers in the color images. It is a challenging problem considering the variety of flower classes and the variation within a particular flower class. Towards this, the proposed method processes the input flower image through a set of trainable deep layers, the weights of which are refined in iterations by using an objective function that can work without labeled data. In this training process, the deep learning-based model learns to assign the same label to pixels with homogeneous semantic characteristics (e.g., the same part of the plant). The segmentation map is finally processed through a saliency model to obtain the foreground segmentation mask, i.e., the flower or flowers. Though training is performed, since the training process is not associated with any external label, the complete process is unsupervised. The proposed method is tested on Oxford flower dataset [18] that shows the the proposed method to be effective.

The remainder of this paper is organized as follows. We briefly discuss the literature relevant to the proposed method in Section 2. The proposed method is outlined in Section 3. Experimental results are described in Section 4. The paper is concluded along with a discussion of future works in Section 5.

II. Related Works

Considering the relevance to the proposed problem statement, we briefly review the supervised and unsupervised deep learning based segmentation methods.

The supervised deep learning based segmentation methods implicitly model segmentation as a pixel-level classification task [19] [7] [8] [20] [9] [21]. They rely on abundant amount of training pixels to train the segmentation model. Such methods usually rely on region proposals (bottom-up), which is used to supervise the training required for segmentation. Fully convolutional networks (FCNs) [7] is a simple model that is effective for supervised semantic segmentation. FCN networks can ingest input of any spatial size and generate a segmentation map of the same size. [8] presents U-Net to improve the capturing of spatial context. Moreover, U-Net uses a symmetric expanding path to improve the accuracy of localization. In comparison to previous models, U-Net has added trainable layers for upsampling. U-Net architecture has gained significant popularity in plant phenotyping [22] [23]. Another important supervised segmentation method is SegNet [9]. It stores the max-pooling indices and subsequently uses them to upsample the encoded features. Wang *et al.* proposed a variant of SegNet [24] for root image analysis. All the supervised methods need significant training samples.

Recently, we observe a development of interest in exploiting deep learning without training samples, i.e., in an unsupervised way [5] [25] . A few methods have been proposed for unsupervised deep image segmentation. W-shaped network is proposed for unsupervised segmentation in [26]. However, the method requires significant amount of unlabeled training data and the proposed network is complex consisting of large number of convolutional layers. Such complexity of network is not desired when number of training samples are few. Kanezaki [16] proposed an unsupervised deep segmentation method that works on single-image input and uses few convolutional layers. Another unsupervised single-image method is proposed in [17] that uses semantic guidance from a pre-trained network and builds up on the joint optimization of weights of the trainable convolutional layers along with pixel labels.

The proposed method is inspired from [17] and thus does not require labeled training data like the supervised methods [9] [21] [19] [7] [8] [20]. Proposed method works on single-image input [15].

III. Proposed algorithm

A. Problem summary

Let us consider the scenario where we have an image X that captures a scene relevant to plant phenotyping. X consists of N pixels $x_n (n = 1, ..., N)$. In this paper, we segregate $x_n (n = 1, ..., N)$ into two classes, one corresponding to the foreground (object or target of interest for plant phenotyping) and the other corresponding to the background.

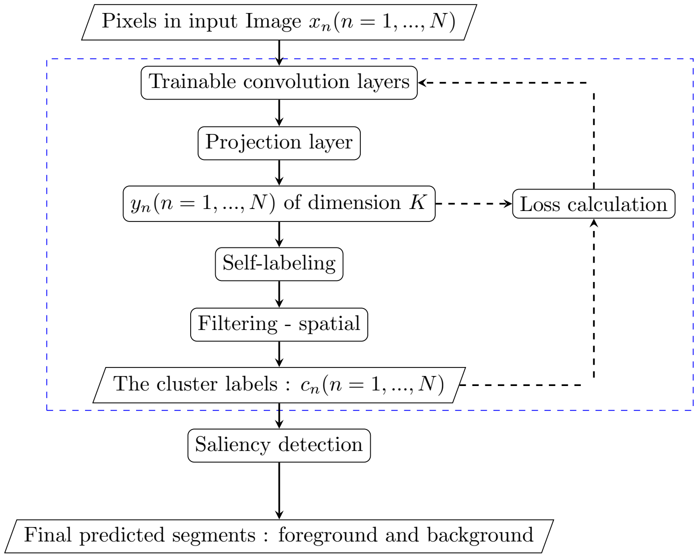
Pixels in input Image $x_n(n = 1, ..., N)$
Trainable convolution layers
Projection layer
$y_n(n = 1, ..., N)$ of dimension K
Loss calculation
Self-labeling
Filtering - spatial
The cluster labels : $c_n(n = 1, ..., N)$
Saliency detection
Final predicted segments : foreground and background

Fig. 1. The proposed method shown as a block diagram. The part shown in blue dashed box is executed over iterations.

B. Gist of the proposed method

The proposed method iteratively learns a deep network from single-image input by self-labeling and representation learning. It processes the input image X (pixels $x_n (n = 1, ..., N)$) through a collection of trainable convolutional layers $f(x_n)$. The weights of the convolutional layers can be initialized by any suitable method [27]. Considering, the weights are fixed, the network is used to extract a set of deep features c_n (dimension K) from the final convolutional layer. The set of deep features is used for self-labeling [28] using *argmax* classification [17]. Following this, the labels and features from final convolutional layers are used to compute loss using the cross-entropy loss function. The weights of the deep network is adjusted using the computed loss, thus leading to better representation learning. Self-labeling and representation learning are two separate but related processes that are carried out alternately. After training for a fixed number of iterations, the segmentation labels $c_n (\forall n = 1..., N)$ is processed through a saliency detection framework [29] to segregate into the foreground and background class.

We show a sketch of the proposed method in Figure 1.

C. Algorithm details

1) Deep feature representation: Given input image X that captures a scene relevant to plant phenotyping, deep feature representation is obtained such that it captures the pixel-wise high-level semantics of X. For this task, a set of L trainable convolutional layers (weights: $\mathbb{W}^1, ..., \mathbb{W}^L$) are used. Stride value is set to 1, thus moving the convolution filters one pixel at a time. This technique is used to preserve the size of the input image. Convolutional layers are further supplemented with Rectified Linear Unit (ReLU) and normalization layer [17]. The input image is convolved with the convolutional filters from the first layer to capture low-level semantics (e.g., edges). Filters from the following layers are convolved with the feature obtained from the previous layers. In this way, deeper layers work upon the lower-level semantics captured by the shallower layers and capture higher-level semantics. The last (L) layer is a linear projection layer (layer having filters of the size 1×1) and is used to project the dimension in the kernel space to K. Here, K is a number that is much larger than the number of distinct segmentation labels. For flower segmentation, distinct segmentation labels can be two (flower and background) or more. For this work, we set $K = 100$. The feature extracted from the projection layer is further normalized in accordance with [17] to produce feature maps on a similar scale. Thus, this deep network produces pixelwise deep representation y_n of dimension K.

2) Self labeling: Since the image (and hence all pixels in it) is processed through the same set of convolutional layers, it can be assumed that pixels belonging to the same semantic class generate high activation value in the same set of deep features. Using this assumption, the deep representation y_n is processed to get the cluster label c_n for each pixel x_n. In more details, label c'_n corresponding to a specific input pixel x_n is obtained by using the *argmax* based classification. It is accomplished by choosing the feature (out of total K features) that has a maximum value in y_n [17]. Since spatial continuity is an expected property in plant image segmentation, it can be assumed that labels for adjacent pixels are similar. To ensure this spatial homogeneity, we further use an effective sliding window mode filtering [17] based image filtering to further process/refine the output map and obtain c_n. In this way, we can obtain label c_n for each pixel x_n. This process of obtaining a label is called self-labeling since the label is computed from the deep features computed from the image without using any external label.

After the self-labeling process, pixels can be assigned to any of the possible K distinct labels. However, in practice, for a typical plant phenotyping image, most pixels obtain *argmax* value in fewer distinct labels. Here, K works as an upper limit to the maximum possible number of clusters.

3) Deep representation learning: The set of L trainable convolutional layers (weights: $\mathbb{W}^1, ..., \mathbb{W}^L$) are trainable and the training process is performed without using any external label. To accomplish this, cross entropy loss is computed between the deep representation (y_n) obtained from the last layer and the labels c_n obtained after the self-labeling process. The cross entropy loss is computed over all the pixels in the image using mean reduction [30]. In this way, we obtain $\mathcal{L}$ as:

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^{N} \text{cross-entropy}(y_n, c_n) \tag{1}$$

The computed loss is used to adjust the weights of the trainable layers:

$$\mathbb{W}^1, ..., \mathbb{W}^L \leftarrow \text{update}(\mathcal{L}) \tag{2}$$

The training process is performed in an iterative fashion over $\mathcal{I}$ iterations. In this work, we used $\mathcal{I} = 500$. The value is chosen keeping consistency with previous works on self-supervised clustering [17].

TABLE I
KERNEL SIZE AND STRIDE FOR THE CONVOLUTIONAL LAYERS

Layer	**Stride**	**Kernel size**
1	1	3×3
2	1	3×3
3	1	3×3
4	1	1×1

4) Final segmentation map: After training process is completed, the trained network is used to obtain segmentation labels c_n for all $x_n (n = 1, ..., N)$. The segmented image is further processed through a saliency detection framework [29]. The method in [29] distinguishes the spatial layout of segmented regions with respect to image boundaries to obtain two clusters, corresponding to the foreground (containing the object-of-interest for plant phenotyping) and the background.

IV. RESULT

A. Experimental setup

We use Oxford flower dataset [18] to validate the proposed method. For experimental evaluation, we use the set of 753 images as described in [2] and we compare the proposed method to the methods in [2]. There is no specific test set in the dataset and the supervised methods in [2] split the images into 80/20 for train/test. By using this mechanism, the supervised methods can exploit what they learn from training images (80%) and subsequently use that knowledge on the test images (20%). On the other hand, the proposed method is unsupervised and can work on single-image input, and hence it does not require the training set. Moreover, it does not have the advantage of transferring knowledge learned from one image (i.e., the images in the training set) to the other image (i.e., the images in the test set). Thus, for a fair comparison, for the unsupervised methods, we compute quantitative metrics using the top 20 percentile per class, arranged as per its performance. We show the quantitative result in terms of mean pixel accuracy [2]. We show two sets of results, one with input images resized to size 128×128 pixels and the other with size 224×224 pixels (as in [2]). A learning rate of 0.005 is used during the training process. The proposed method is implemented using PyTorch [30].

B. Complexity of the network

The proposed model is light-weight as we use $L = 4$, i.e., 4 convolutional layers. The number of parameters in the proposed model is 193,900 which is considerably smaller than the models in [2] where model with least parameter has 360,743 parameters. The kernel size and stride for the 4 layers are tabulated in the Table I.

C. Qualitative result

We show the sample results from the proposed method in Figure 2. It is clear that the proposed method is able to delineate flowers from the background. Moreover, the proposed method is able to work where flowers show significant variation in color and texture. However, in some cases, the proposed method is prone to over-segmentation, as evident in the case shown in the last row of Figure 2.

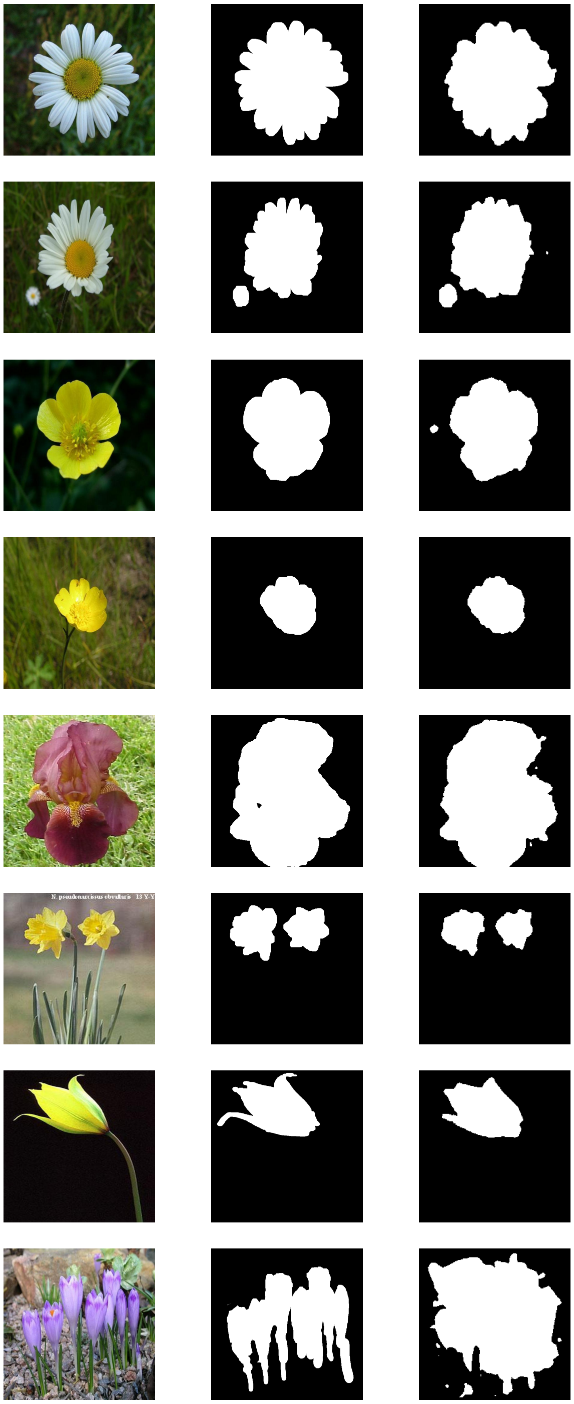

Fig. 2. Result on the Oxford flower dataset. Input images are shown in 1st column, groundtruth segmentation images are shown in column 2, results obtained with the proposed method are shown in column 3. While top 7 rows show cases where the proposed method successfully segments the flowers, last row demonstrates a case of over-segmentation.

TABLE II
PIXEL ACCURACIES ON THE OXFORD FLOWER DATASET. WE SEGMENT THE FLOWERS INTO TWO CLASSES - FLOWERS AND BACKGROUND. PERFORMANCE OF PROPOSED UNSUPERVISED METHOD IS COMPARABLE TO THE SUPERVISED METHODS.

Method	Supervision	**Accuracy (%)**
FCN-VGG16	Yes	97.23
FCN-VGG16 Lite-0.25	Yes	96.30
ResNet18	Yes	95.75
ResNet18 Lite-0.25	Yes	95.09
Proposed (Input size: 128×128**)**	No	95.03
Proposed (Input size: 224×224**)**	No	96.01
SLIC + saliency detection	No	94.66

D. *Quantitative result*

Quantitative comparison of the proposed method to the supervised methods in [2] is shown in Table II.

Fully convolutional network (FCN) [7] for semantic segmentation is adopted in FCN-VGG16. FCN-VGG16 Lite-0.25 is a modified version of FCN-VGG16 with a reduction in the number of parameters [2]. ResNet18 is deep residual learning-based network [31] and ResNet18 Lite-0.25 is a modified version it with a reduction in parameters [2]. The Lite models have reduced computational complexity in training due to the reduction in parameter size [2].

Due to the stochastic nature of the training process in the proposed method, there can be slight variations in quantitative results over different executions. Here, we show results averaged over three executions. The proposed method obtains superior result when performed with a larger input size, which is intuitive as there is information loss while down-sampling image. In spite of being unsupervised, the proposed method is able to obtain satisfactory mean pixel accuracy, which is almost at par with the supervised methods. Note that the proposed method is a first attempt to self-supervised deep segmentation for plant phenotyping applications and hence the comparison to deep learning based methods in this paradigm is not possible. However, we compare the proposed method to an unsupervised baseline using simple linear iterative clustering (SLIC) clustering [32] followed by saliency detection. The proposed method outperforms this baseline.

E. *Timing requirement*

For 128×128 pixel input, the method takes approximately 9 seconds per image (on a computer having NVidia Geforce RTX 2080 Ti GPU). For 224×224 pixel input, the method takes approximately 26 seconds per image.

V. CONCLUSIONS

In this paper, we propose a self-supervised image segmentation method for flower segmentation. The proposed method can work in absence of labeled training data. In spite of being unsupervised/self-supervised, the proposed method obtains similar results to the supervised methods for a large fraction of the images in the dataset. The proposed method can be used as a pre-processing step for other plant phenotyping tasks. The proposed method can also be used to obtain weak labels for further training a supervised network for phenotyping. While the proposed method is certainly not a replacement for supervised methods, it opens up a research direction towards a better understanding of image-based plant phenotyping. In the future, we plan to cluster plant images into a larger number of classes. We also plan to extend the method for other plant phenotyping datasets.

References

[1] S. Aich and I. Stavness, "Leaf counting with deep convolutional and deconvolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 2080–2089, 2017.

[2] J. Atanbori, F. Chen, A. P. French, and T. P. Pridmore, "Towards low-cost image-based plant phenotyping using reduced-parameter CNN," in *CVPPP 2018: Workshop on Computer Vision Problems in Plant Phenotyping*, 2018.

[3] M. V. Giuffrida, M. Minervini, and S. Tsaftaris, "Learning to count leaves in rosette plants," in *Proceedings of the Computer Vision Problems in Plant Phenotyping (CVPPP)*, pp. 1.1–1.13, BMVA Press, September 2015.

[4] M. P. Pound, J. A. Atkinson, A. J. Townsend, M. H. Wilson, M. Griffiths, A. S. Jackson, A. Bulat, G. Tzimiropoulos, D. M. Wells, E. H. Murchie, *et al.*, "Deep machine learning provides state-of-the-art performance in image-based plant phenotyping," *Gigascience*, vol. 6, no. 10, p. gix083, 2017.

[5] E. Cai, S. Baireddy, C. Yang, M. Crawford, and E. J. Delp, "Deep transfer learning for plant center localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 62–63, 2020.

[6] P. Sodhi, S. Vijayarangan, and D. Wettergreen, "In-field segmentation and identification of plant structures using 3d imaging," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5180–5187, IEEE, 2017.

[7] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

[8] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[9] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[10] J. Chen and X. Shi, "A sparse convolutional predictor with denoising autoencoders for phenotype prediction," in *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pp. 217–222, 2019.

[11] Y. Zhu, M. Aoun, M. Krijn, J. Vanschoren, and H. T. Campus, "Data augmentation using conditional generative adversarial networks for leaf counting in arabidopsis plants.," in *BMVC*, p. 324, 2018.

[12] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[13] S. Saha, F. Bovolo, and L. Bruzzone, "Building change detection in VHR SAR images via unsupervised deep transcoding," *IEEE Transactions on Geoscience and Remote Sensing*, 2020.

[14] S. Saha, F. Bovolo, and L. Bruzzone, "Unsupervised multiple-change detection in vhr multisensor images via deep-learning based adaptation," in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, pp. 5033–5036, IEEE, 2019.

[15] A. Shocher, N. Cohen, and M. Irani, "“zero-shot" super-resolution using deep internal learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3118–3126, 2018.

[16] A. Kanezaki, "Unsupervised image segmentation by backpropagation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1543–1547, IEEE, 2018.

[17] S. Saha, S. Sudhakaran, B. Banerjee, and S. Pendurkar, "Semantic guided deep unsupervised image segmentation," in *International Conference on Image Analysis and Processing*, pp. 499–510, Springer, 2019.

[18] M.-E. Nilsback and A. Zisserman, "Delving deeper into the whorl of flower segmentation," *Image and Vision Computing*, vol. 28, no. 6, pp. 1049–1062, 2010.

[19] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pp. 580–587, IEEE, 2014.

[20] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," *arXiv preprint arXiv:1606.02147*, 2016.

[21] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[22] V. Kulikov, V. Yurchenko, and V. Lempitsky, "Instance segmentation by deep coloring," *arXiv preprint arXiv:1807.10007*, 2018.

[23] A. G. Smith, J. Petersen, R. Selvan, and C. R. Rasmussen, "Segmentation of roots in soil with u-net," *Plant Methods*, vol. 16, no. 1, pp. 1–15, 2020.

[24] T. Wang, M. Rostamza, Z. Song, L. Wang, G. McNickle, A. S. Iyer-Pascuzzi, Z. Qiu, and J. Jin, "Segroot: A high throughput segmentation method for root image analysis," *Computers and Electronics in Agriculture*, vol. 162, pp. 845–854, 2019.

[25] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 806–813, 2014.

[26] X. Xia and B. Kulis, "W-net: A deep model for fully unsupervised image segmentation," *arXiv preprint arXiv:1711.08506*, 2017.

[27] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

[28] Y. M. Asano, C. Rupprecht, and A. Vedaldi, "Self-labelling via simultaneous clustering and representation learning," *arXiv preprint arXiv:1911.05371*, 2019.

[29] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2814–2821, 2014.

[30] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[32] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.