

Soybean Leaf Coverage Estimation with Machine Learning and Thresholding Algorithms for Field Phenotyping

Kevin Keller³

kellerke@student.ethz.ch

Norbert Kirchgessner¹

norbert.kirchgessner@usys.ethz.ch

Raghav Khanna²

raghav.khanna@mavt.ethz.ch

Roland Siegwart²

rsiegwart@ethz.ch

Achim Walter¹

achim.walter@usys.ethz.ch

Helge Aasen¹

helge.aasen@usys.ethz.ch

¹ Crop Science

Universitätstrasse 2

Building LFW, C 55.1

8092 Zürich, Switzerland

² Autonomous Systems Lab

Leonhardstrasse 21

8092 Zürich, Switzerland

³ Information Technology and

Electrical Engineering

Gloriastrasse 35

8092 Zürich, Switzerland

Abstract

Our project compares leaf coverage estimation techniques using machine learning and thresholding algorithms on a set of soybean images captured in a field under natural illumination conditions. Using Amazon Mechanical Turk for image segmentation, we developed a labeled training data set of 285 early growth soybean plant images. These training masks and images were used to train and evaluate three segmentation techniques: thresholding, random forest classifiers, and a deep convolutional neural network. The color based thresholding performed the best on our dataset with a mean intersection over the union score of 87.52% while the random forest and the deep learning model scored 51.24% and 78.65% respectively.

1 Introduction

Plant phenotyping of in-field crop cultivars to determine optimal crop varieties for specific soil and climate conditions is a major focus in current plant research. Increasing crop yield and plant growth efficiency over the next decades will be necessary to meet current global predicted food demands. Plant phenotyping is used to compare crop varieties for growth rate, as well as to determine the best drought and climate-change resistant varieties of common crops [1]. Field phenotyping allows plant researchers to bridge the gap between plant genomics and phenotype in real-world experiments. Indoor testing of plant cultivars allow faster and more efficient testing, but do not always accurately predict growth patterns in outdoor environments. Plant growth between identical cultivars can vary between outdoor and

indoor growing environments due to differences in nutrients and water availability in an artificial growing environment, as well as environmental influences such as wind, air humidity, solar radiation, and cloud coverage [1]. Analysis of these traits in real-world conditions require in-field phenotyping methods on plants in outdoor research fields. Plants in a field do not grow in isolate as in pots, but form a complex canopy that increases the difficulty of leaf analysis due leaf occlusion and shadow differences on lower and upper leaves in the canopy.

To accurately and reliably study the in-field growth pattern of the research plant cultivars, high throughput field phenotyping methods are required in the research fields [2]. High throughput methods allow automated or semi-automated analysis to be applied on the growing plants. High throughput plant phenotyping relies on remote sensing methods such as near/far-infrared sensing, as well as RGB imaging systems. These sensors provide valuable information on plant growth, but are subject to imaging issues due to crop illumination from atmospheric changes such as clouds and any analysis must take this issue into account [3]. Multiple options are currently used to capture field phenotyping data including Unmanned Aerial Vehicles (UAV), ground robots, tractor attachments, or cameras suspended above research fields as shown in [4], [5].

One useful metric for plant phenotyping is leaf coverage, as it indicates plant growth rate as well as predicted yield [2] and can be used to compare experimental soybean cultivars. Our project focuses on leaf coverage estimation of early growth soybean plants using RGB images. We focus on RGB images as opposed to near/far infrared as this allows the segmentation methods we test to be applied to images captured by smaller camera systems such as those on UAVs. A major goal of plant phenotype and precision farming researchers is to develop software for use on widely available UAVs to allow easier data collection and analysis without large investments in imaging infrastructure [3].

Leaf coverage is often estimated using thresholding methods on images of the plants, or by training supervised machine learning models to semantically segment the leaves [4],[3], [5]. For our leaf coverage estimation we developed a training dataset of images and ground-truth masks, and tested various thresholding and machine learning algorithms focused on semantic segmentation. The ground-truth image masks in the training dataset were developed using crowdsourcing provided through Amazon Mechanical Turk.

2 Training Data Set

To train the segmentation algorithms, we need training data that contains crop images and corresponding ground-truth black/white masks to indicate leaf area. A separate validation set is used to evaluate the trained models. For our training and validation data sets, we used a set of soybean crop images from the Swiss Federal Institute of Technology Eschikon Field Research Station. The soybean crop images were produced by the Field Phenotyping Platform (FIP) which captures high-resolution aerial images of the individual plants growing in the test beds [3]. Two different soybean cultivars, Gallec and Pollux, were used in the images. The crop images were captured in 16 bit TIFF format using a Canon EOS 5D Mark II camera with a 35mm lens during daylight hours in June. The set of FIP images show a time progression of each plant over the course of the growing season.

Segmentation of plant images pose a unique challenge in comparison to urban or interior scenes commonly used for image segmentation tasks. Strict leaf boundary identification is often challenging due to illumination changes, leaf occlusion, and geometric variability between individual leaves. The outdoor growing environment of the plants also pose a challenge for our segmentation task due to illumination condition variation between images [3].

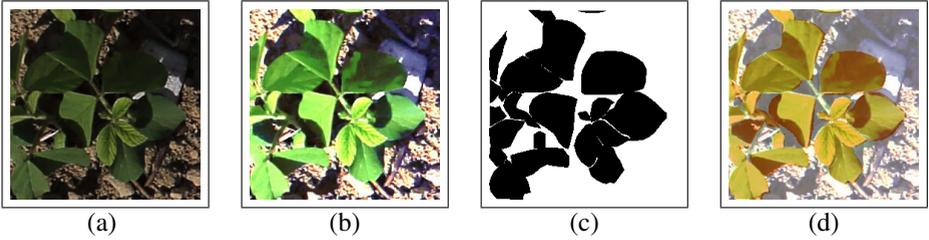


Figure 1: Example image and mask pair from data set: (a) Original image soybean image; (b) Color stretched image used for edge enhancement during manual segmentation; (c) Mechanical Turk segmentation mask; (d) Overlay of mask and image.

2.1 Image Selection

We used early-growth soybean plant images for the segmentation dataset. The early growth images were selected due to the ease of leaf boundary identification as later growth stage images often contain 100% leaf coverage with no visible soil. To aid in edge identification during manual segmentation, we applied a 20% decorrelation color stretch on the RGB images to enhance color differences. We then extracted a set of 256x256 pixel tiles that are centered around individual soybean leaf clusters as shown in Figure 1.

2.2 Training Masks

We used Amazon Mechanical Turk and the open-source LabelMe Annotation Tool [20] to manually develop a segmentation mask for each image tile. Mechanical Turk is a crowdsourcing platform to allow small, manual tasks to be quickly, and cheaply completed. Each image in our dataset was uploaded to the Mechanical Turk website, and a human worker was asked to manually annotate the boundary of each leaf in the image using the LabelMe Annotation software.

To reduce the noise in the final segmentation masks, each image was segmented three times by different workers to produce three unique masks. Using the three different masks for each image, we manually selected the most accurate image mask based on the overlay of the mask and the original image. For some sets of segmentation masks from the workers, there was no clear 'best' mask available. In this situation we used a pixel based majority voting mechanism to merge the three existing masks to one final segmentation mask. The final dataset consists of 285 training images with corresponding binary masks. Figure 1 shows an example of a training image and corresponding mask from our dataset. The original, non-colorstretched, image is used for model training.

2.3 Dataset Augmentation

Dataset augmentation is a common method to increase the size of a dataset as well as decrease overfitting during the training [15], [16]. For training and validation of the models, we randomly split the 285 images in our labeled dataset into 240 training and 45 validation image pairs, and then applied three different augmentation techniques to the images. Each image underwent a mirror, a flip, and a 90 degree rotation operation to produce a final dataset of 960 training, and 180 validation images. To test the efficacy of the augmentation, we used both the original and the augmented datasets in parallel in our experiments.

3 Segmentation Methods

We tested three different segmentation techniques using our dataset. The first option was a color based thresholding method as a baseline score [24], and then we tested a random forest classifier, and a deep learning convolutional neural network called DeepLab. Recent developments in deep learning networks have shown to provide better results on image segmentation tasks than traditional feature-based learning methods such as random forest classifiers [9], [9]. The DeepLab model was selected for our experiments based on the recommendation in [24] due to the model performance on RGB image datasets.

3.1 Evaluation Metrics

For the quantitative evaluation of our models, we used the Mean Intersection over the Union (mIOU) to compare the predicted image mask versus the validation set mask [19]. The IOU metric is the intersection of the prediction and validation masks over the union of the prediction and validation masks, with a perfect alignment returning 1. The final metric is the mean over the entire validation set.

3.2 HSV Segmentation

The baseline segmentation technique is based on color thresholding in the Hue Saturation Value (HSV) color space. The HSV color space is different than the standard Red Green Blue (RGB) since it separates the pixel intensity from the actual color of the image [24]. This is useful for our data set as illumination conditions between images change due to the outdoor conditions.

To preprocess the image, we used a Gaussian blur on the image with a σ value of 3 and then converted to the HSV color space. Using only the Hue channel from the HSV image, we applied an Otsu Threshold to extract the foreground from the background of the image [25]. The Otsu thresholding method builds a histogram of pixel intensities in the image, and attempts to find a threshold in this histogram that maximizes the inter-class variance. We experimentally selected the H channel to use for the thresholding level selection as it produced the strongest result during initial testing.

3.3 Random Forest

The random forest classifier [9] is a machine learning model that is an extension of a decision tree classifier. The specific model was implemented in python using the sklearn python package. The HSV pixel values, the RGB pixel values, and a set of Histogram of Oriented Gradient (HOG) features extracted from the Gaussian blurred HSV image were used as training input to the random forest classifier. HOG features represent the shape and orientation of objects in an image and are commonly used for object detection and segmentation [23], [8]. Each HOG feature consists of eight gradient values calculated on an 8x8 pixel patch.

Our model used 64 trees in the final estimator parameters which provided a good trade-off between training time and accuracy based on our initial testing. Beyond 64 trees, the marginal increase in mIOU results did not justify the extra time required for the training. We ran a set of experiments using the random forest classifier to determine what effect the preprocessing and feature combinations had on the evaluation results. The final best combination was then trained on the final 960 image dataset. We tested the network using the following model combinations on the original 240 image dataset:

- HSV pixel values
- HSV and RGB pixel values

- HSV pixel values and HOG features
- HSV and RGB pixel values and HOG features

3.4 DeepLabv3+

DeepLabv3+ is a Convolutional Neural Network model designed for pixel based semantic image segmentation [9], [10]. The model is open source and implemented in tensorflow [11]. Convolutional Neural Networks use a set of filters convolved with the input data to extract sets of features from the data. The features from each layer are combined into feature maps based on the training data that can then be used to make the output prediction.

DeepLabv3+ builds on the DeepLabv3 design and combines a spatial pyramid pooling structure with an encoder-decoder structure to achieve state of the art segmentation results. The original DeepLabv3 model [9] is used to encode the input information with atrous convolution layers, and the additional decoder module is used to extract the boundary information for the final segmentation. The advantage of using DeepLabv3 as the encoder is that the atrous convolution rate can be varied based on computation resources during training.

Atrous convolutions, also called dilated convolutions, are convolutions performed by increasing the size of the filter by padding with 0 between elements. This preserves the original filter and just changes the field of view of the filter kernel [12]. For example, an atrous convolution with an atrous rate of 2, would add a 0 element between each element in the original convolution filter to double the effective size of the kernel. This acts to increase the field of view of the filter. In the DeepLab model, multiple atrous convolutions are performed on the images at different rates to capture image features at multiple scales.

For testing on our soybean dataset we started with a model checkpoint provided by the DeepLab team to initialize the network. This checkpoint was initialized using the PASCAL VOC dataset. During training, we used the following parameters:

output stride = 16; atrous rates = [6,12,18]; batch size = 4; model variant = xception_65; initial checkpoint = deeplab3_pascal_train_aug_2018_01_04

Model convergence was defined based on the rate of mIOU score improvement. During training, the mIOU was calculated at every 50 iterations. The stopping criteria was set to be when the mIOU score improvement was less than 0.0001 in 100 iterations.

We ran four different experiments using the DeepLabv3+ model to determine what effect the input data size and preprocessing had on the evaluation results.

- Original RGB dataset with 240 images with default initialized model.
- Augmented RGB dataset with 960 images with default initialized model.
- Augmented HSV dataset with 960 images using the previously trained Augmented RGB model as the initial training model.
- Augmented HSV dataset with 960 images with default initialized model.

4 Results

For the segmentation experiments, we tried three main techniques: color based segmentation, random forest segmentation, and a deep learning model. We used the original 240 image dataset, as well as the augmented and HSV datasets to evaluate the various models using the mean IOU score on the validation sets.

4.1 HSV Thresholding Results

The HSV color space thresholding results using the original and augmented datasets are shown in Table 1. Since there is no learning using this method, the difference between the

original and augmented results is unexpected but is caused by the otsu thresholding.

Figure 2 shows the cause of the mIOU difference between the augmented and the original dataset. The augmentation process of mirror, flip, and rotate results in differences in the otsu thresholded image. The four predicted masks contain slight differences when examined closely. For example, the white clusters in the center are shaped differently between all four predicted masks leading to differences in the IOU score between augmented images when compared to the target mask.

	Original	Augmented
HSV Threshold	0.85746	0.8752

Table 1: mIOU scores for HSV thresholding on the original training/validation dataset.



Figure 2: Otsu threshold applied on single image after augmentation process: original, mirrored, flipped, rotated. Note the differences in noise level and color in the center of mask.

4.2 Random Forest Results

The comparison of different random forest feature vectors were all tested on the original training data set with 240 training and 45 validation images. The models were trained using the ETH Leonhard GPU cluster. Table 2 shows the results of the different random forest experiments. Using only the HSV pixel values as the feature vector produces the strongest results using the original dataset.

Feature Vector	Original
HSV	0.4649
RGB	0.3746
HSV + HOG	0.4079
HSV + RGB + HOG	0.40396

Table 2: mIOU scores for random forest classifier on the original training/validation dataset.

After the initial round of random forest classifier experiments, we took the best method, the HSV pixel values, and retrained the classifier using the larger augmented data set. Table 3 shows the final results of the HSV pixel value random forest classifier on the datasets.

4.3 DeepLab Results

The DeepLab model was trained on four different parameter combinations as described in section 3.4 and the mIOU results are shown in Table 4. The results show a strong improvement of 4.1% between the RGB augmented dataset and the RGB original dataset indicating that the augmentation method was beneficial for the final performance. The best performance of the deeplab model was reached using the augmented dataset after converting to the HSV color space. Since the HSV color space is designed to remove issues with illumination changes, this indicates that the DeepLab model was not fully accounting for these changes.

Feature Vector	Original	Augmented
HSV	0.4649	0.5124

Table 3: mIOU scores for random forest classifier using HSV pixel values on datasets.

	mIOU	DataSet	iterations
RBG+HSV	0.7196	augmented	3000
RGB	0.7392	original	4500
RGB	0.7802	augmented	8000
HSV	0.7865	augmented	4500

Table 4: mIOU scores for DeepLab experiments. Note that for the RGB+ HSV model, we used the fully trained Augmented RGB model as the initial checkpoint to start the HSV training. This decreased the total training time required, but did not produce good results.

5 Discussion

Between the different segmentation methods tested, Table 5 shows the final mIOU results for the top performing methods in each category. The final mIOU scores show the simple HSV thresholding performed the best on our dataset. The DeepLab method had surprisingly poor results given how well the model performs on alternative data sets.

	mIOU	Data Set	Runtime
HSV RF	0.5124	augmented	8 hours
HSV Threshold	0.8752	augmented	2 minutes
DeepLab HSV	0.7865	augmented	40 hours

Table 5: mIOU scores for best performance between each segmentation method along with approximate training time. The DeepLab HSV model trained in less time than the RBG.

5.1 Predicted Masks

Despite the higher mIOU score on the HSV thresholding, the DeepLab prediction results through the whole dataset are more consistent, and reliable than the HSV thresholding as shown in Figure 3. Each predicted mask visually appears similar to the desired target mask, but lacks the fine details that are required. For example for clusters of occluded leaves with shadow as seen in image (e), the DeepLab model is not able to segment the leaf boundary reliably. The predicted mask image for (a) is missing the small leaves present on the edge of the image that appear in the target mask. Leaf boundary segmentation issues possibly stem from noise in the original training set caused by the challenging manual segmentation as discussed in Section 2.2.

The majority of the HSV thresholding results match the expected target well, but like the DeepLab results, the predicted masks lack sharply defined leaf boundaries as seen in image (d), and (e). Unlike the DeepLab validation results however, the HSV threshold predictions on the dataset contains a number of outliers. Image (a) shows an example of a thresholding mistake where the otsu thresholding algorithm was not able to detect the correct thresholding level and removes most of the leaf area. Comparing the predicted mask with the original image (a) shows that the predicted mask area lies in the dark shadow in the upper right. This indicates that this outlier is caused by the same issue as discussed in Section 3.2. These outliers in the final dataset indicate that despite the higher mIOU score, the HSV thresholding method is not as well suited to leaf segmentation as the DeepLab approach.

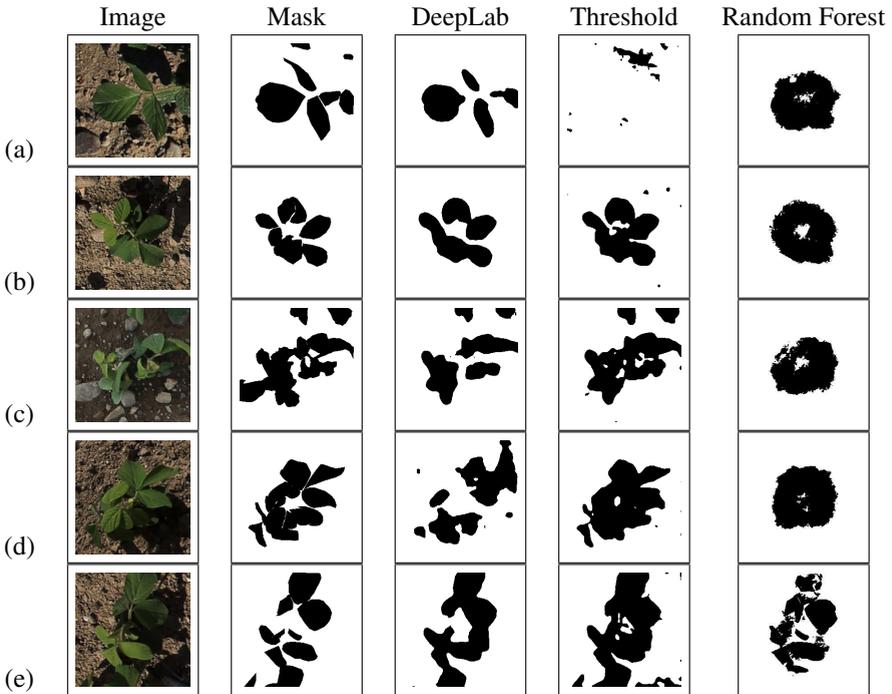


Figure 3: Validation image, ground-truth mask, and three predicted masks using the three segmentation techniques evaluated in this section.

The random forest results are strange. Image (a)-(d) represent the common segmentation pattern in the predicted mask. One hypothesis for this strange result is that the mask prediction is biased towards leaves in the center of the image. The training dataset was created using images that center on a leaf cluster, so perhaps a more effective random forest training would be to train on smaller tile from each image to avoid the bias towards the center pixels. Image (e) for the predicted masks show an outlier in the random forest predictions where the predicted mask starts to capture some of the geometrical features in the image. This poor performance could be due to poorly selected features for the training, the small size of the dataset, or incorrectly selected hyperparameters for the classifier training. Preprocessing the input data using dimension reduction techniques or better feature selection could improve the final results.

5.2 Run Time

The approximate runtime for each segmentation method is presented in Table 5. The HSV thresholding is the fastest method at only two minutes as it requires no training time for the final prediction. The random forest required between 1 to 8 hours on the GPU cluster. The DeepLab model required 40 hours for HSV Augmented dataset convergence, and 80 hours for the RGB augmented dataset indicating that the HSV preprocessing was efficacious.

6 Conclusion

This project focused on soybean leaf coverage estimation using three different segmentation methods. We developed a custom training and validation dataset from soybean crop images from outdoor research facilities using crowdsourcing to develop a set of binary masks for the leaf coverage in the image. The dataset size and quality were lower than ideal, but our

overall results were acceptable. Further work to increase the dataset size and quality would lead to greater performance of our models.

Using this training dataset we tested three segmentation techniques; random forest classifiers, color thresholding, and a deep learning neural network. The color thresholding method achieved the highest results using the Mean Intersection over the Union evaluation metric. Despite the high performance achieved by deep learning neural networks on other segmentation datasets, our implementation did not perform beyond the relatively high baseline set by the HSV color thresholding segmentation. However, despite the higher mIOU score on the HSV thresholding, the DeepLab predictions were more consistent and did not contain outliers as seen in the HSV thresholding predictions.

The two main conclusions from these segmentation experiments are that the dataset sizing is critical in the deep learning methods, and that preprocessing the data to the correct color space improves the results for all methods. Our augmented dataset is only 960 training images in comparison to the commonly used PASCAL Visual Object Classes dataset that contains 11,000 images. Preprocessing the images with the conversion from RGB to HSV had a large impact on the performance of the final mIOU evaluation. This conversion decreased the issues caused by uneven and inconsistent lighting conditions between the plant images. Soybean leaves hidden in the shadows were difficult for the segmentation methods to extract.

6.1 Future Work

Future work in this area can build on the results of these experiments and achieve greater performance. Based on the analysis of the tested segmentation methods, increasing the size of the dataset, and stronger shadow and color correction preprocessing would lead to the greatest improvement in the evaluation. This project trained and validated on small image tiles, but the end goal is to classify full sized images. Future work will be to expand the classification methods to larger images captured from the field phenotyping platform.

Alternative augmentation methods would be the easiest to test without requiring more data collection. Augmentation using additive noise and blurring, warping and scaling, or using randomly extracted patches from image/mask pairs would be beneficial. There are many possible augmentation methods, and for this set of experiments, we did not have the time to evaluate which augmentation method was most efficacious. The initial size, 256x256 pixel, of the image might also have an effect on the output prediction performance due to the atrous convolutional upsampling during the training phase. Future work could experiment with larger images, or different atrous rates to determine the effect on the mIOU. Future work will use model loss, and not mIOU performance, as stopping criteria since a larger mIOU does not always indicate a fully trained model.

Another option to increase performance of the model would be to add multiple pre-computed features to the input data. The DeepLab implementation supports png or jpg input images, but modifications to the network would allow larger input feature maps that contain HSV or additional feature information to be used for the training. Additionally, decreasing the number of hidden layers or using a pretrained model could improve performance and decrease training time.

References

- [1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.
- [2] José Luis Araus and Jill E Cairns. Field high-throughput phenotyping: the new crop breeding frontier. *Trends in plant science*, 19(1):52–61, 2014.
- [3] L Breiman. Random forests machine learning. 45: 5–32. *View Article PubMed/NCBI Google Scholar*, 2001.
- [4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Re-thinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.
- [5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.
- [6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *arXiv:1802.02611*, 2018.
- [7] Dan Ciresan, Alessandro Giusti, Luca M. Gambardella, and Jürgen Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2843–2851. Curran Associates, Inc., 2012.
- [8] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.
- [9] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1915–1929, 2013.
- [10] Fabio Fiorani and Ulrich Schurr. Future scenarios for plant phenotyping. *Annual review of plant biology*, 64:267–291, 2013.
- [11] Robert T Furbank and Mark Tester. Phenomics—technologies to relieve the phenotyping bottleneck. *Trends in plant science*, 16(12):635–644, 2011.
- [12] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and Jose Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*, 2017.
- [13] S. Haug, A. Michaels, P. Biber, and J. Ostermann. Plant classification system for crop /weed discrimination without segmentation. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1142–1149, March 2014.

- [14] Norbert Kirchgessner, Frank Liebisch, Kang Yu, Johannes Pfeifer, Michael Friedli, Andreas Hund, and Achim Walter. The eth field phenotyping platform fip: a cable-suspended multi-sensor system. *Functional Plant Biology*, 44(1):154–168, 2017.
- [15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [16] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [17] Jianguo Liu and Elizabeth Pattey. Retrieval of leaf area index from top-of-canopy digital photography over agricultural crops. *Agricultural and Forest Meteorology*, 150(11):1485–1490, 2010.
- [18] M. Minervini, H. Scharf, and S. A. Tsaftaris. Image analysis: The new bottleneck in plant phenotyping [applications corner]. *IEEE Signal Processing Magazine*, 32(4):126–131, July 2015. ISSN 1053-5888. doi: 10.1109/MSP.2015.2405111.
- [19] Md Atiqur Rahman and Yang Wang. Optimizing intersection-over-union in deep neural networks for image segmentation. In *International Symposium on Visual Computing*, pages 234–244. Springer, 2016.
- [20] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Labelme: A database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1):157–173, May 2008.
- [21] Pouria Sadeghi-Tehran, Nicolas Virlet, Kasra Sabermanesh, and Malcolm J Hawkesford. Multi-feature machine learning model for automatic segmentation of green fractional vegetation cover for high-throughput field phenotyping. *Plant methods*, 13(1):103, 2017.
- [22] Hanno Scharf, Massimo Minervini, Andrew P. French, Christian Klukas, David M. Kramer, Xiaoming Liu, Imanol Luengo, Jean-Michel Pape, Gerrit Polder, Danijela Vukadinovic, Xi Yin, and Sotirios A. Tsaftaris. Leaf segmentation in plant phenotyping: a collation study. *Machine Vision and Applications*, 27(4):585–606, May 2016.
- [23] Florian Schroff, Antonio Criminisi, and Andrew Zisserman. Object class segmentation using random forests. In *BMVC*, pages 1–10, 2008.
- [24] Shamik Sural, Gang Qian, and Sakti Pramanik. Segmentation and histogram generation using the hsv color space for image retrieval. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 2, pages II–II. IEEE, 2002.
- [25] Miss Hetal J Vala and Astha Baxi. A review on otsu image segmentation algorithm. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 2(2):pp–387, 2013.
- [26] Nicolas Virlet, Kasra Sabermanesh, Pouria Sadeghi-Tehran, and Malcolm J Hawkesford. Field scanalyzer: an automated robotic field phenotyping platform for detailed crop monitoring. *Functional plant biology*, 44(1):143–153, 2017.

- [27] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.
- [28] Kang Yu, Norbert Kirchgessner, Christoph Grieder, Achim Walter, and Andreas Hund. An image analysis pipeline for automated classification of imaging light conditions and for quantification of wheat canopy cover time series in field phenotyping. *Plant methods*, 13(1):15, 2017.