



HAL
open science

Pixelwise instance segmentation of leaves in dense foliage

Jehan-Antoine Vayssade, Gawain Jones, Christelle Gée, Jean-Noël Paoli

► To cite this version:

Jehan-Antoine Vayssade, Gawain Jones, Christelle Gée, Jean-Noël Paoli. Pixelwise instance segmentation of leaves in dense foliage. *Computers and Electronics in Agriculture*, 2022, 195, pp.106797. 10.1016/j.compag.2022.106797 . hal-03641427

HAL Id: hal-03641427

<https://institut-agro-dijon.hal.science/hal-03641427v1>

Submitted on 22 Jul 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Pixelwise Instance Segmentation Of Leaves In Dense Foliage

Jehan-Antoine Vayssade^a, Gawain Jones^a, Christelle Gée^a, Jean-Noël Paoli^{a,*}

^a*Agroécologie, AgroSup Dijon, INRAE, Univ. Bourgogne-Franche-Comté, 26 Bd Dr Petitjean, F-21000 Dijon, France*

Abstract

Detecting and identifying plants using image analysis is a key step for many applications in precision agriculture (from phenotyping to site specific weed management). Instance segmentation is usually carried on to detect entire plants. However, the shape of the detected objects changes between individuals and growth stages. A relevant approach to reduce these variations is to narrow the detection on the leaf. Nevertheless, segmenting leaves is a difficult task, when images contain mixes of plant species, and when individuals overlap, particularly in an uncontrolled outdoor environment. To leverage this issue, this study based on recent Convolutional Neural Network mechanisms, proposes a pixelwise instance segmentation to detect leaves in dense foliage environment. It combines “deep contour aware” (to separate the inner of big leaves from its edges), “Leaf Segmentation through classification of edges” (to separate instances with a specific inner edges) and “Pyramid CNN for Dense Leaves” (to consider edges at different scales). But the segmentation output is also refined using a Watershed and a method to compute optimized vegetation indices (DeepIndices). The method is compared to others running the leaf segmentation challenge (provided by the International Network on Plant Phenotyping)

*Corresponding author

Email address: jean-noel.paoli@agrosupdijon.fr (Jean-Noël Paoli)

and applied on an external dataset of Komatsuna plants. In addition, a new multispectral dataset of 300 images of bean plants is introduced (with dense foliage, individuals overlapping, mixes of species and natural lighting conditions). The ground truth (e.g. the leaves boundaries) is defined by labelled polygons and can be used to train and assess the performance of various algorithms dedicated to leaf detection or crop/weed classification. On the usual datasets, the performances of the proposed method are similar to those of the usual methods involved in the leaf segmentation challenges. On the new dataset, their results are strongly better than those of the usual RCNN method. Remaining errors are bad fusion between neighboring areas and over segmentation of multi-foliate leaves. Structural analysis methods could be studied in order to overcome these deficiencies.

Keywords: precision agriculture, remote sensing, leaf segmentation, dense foliage, boundary detection, semantic segmentation, CNN, multispectral

1. Introduction

In precision agriculture, one of the hardest tasks is the detection of crops and weeds by imagery. Imaging systems will play a significant role in the new generation of agriculture, from the genetic selection in phenotyping (Omari et al., 2020) to site-specific weed management (Louargant et al., 2017). They are also used to record frequently and accurately the plant growth, crop yield, leaf area, etc (Gée et al., 2021). These data are then used to quantify and evaluate the quality of production.

In proximal detection, this work is mainly done at the plant level and gives important agronomic information once reported at the plot level (number of weeds, crops and weeds locations, diseases, stress ...). The plant level is also needed for the new agricultural revolution, such as robotic weed management. Thus, studies try to detect crop and weeds plants by using a wide variety of techniques, which have been reviewed by (Wang et al., 2019). The instance segmentation is a key-step used before the task of classifying plants as crop and weeds. In CNN field, it is mostly based on a major class approaches (Hafiz and Bhat, 2020), such as Mask-RCNN. However, detecting the entire plants has 3 main limitations (i) when plants are too numerous, it is hard to distinguish individuals when they overlap, one of them is undetected or both are merged which is due to the underlying non-max suppression algorithm (Bonneau et al., 2020), (ii) small and thin elements are undetected and (iii) the number of variations for a plant is infinite: the number of leaves, their orientations, their sizes and other differences that radically change between individuals and growth stages, implies a large amount of data for the training process. These limitations logically impact the detection rate and may cause a significant quantity of miss-detection. To solve this problem, one approach that should be relevant is to base the detection not on the whole plant, but on the leaf. For this purpose, a pixel-wise

28 instance segmentation is proposed. The idea is to separate the instances by
29 detecting the edges of the leaves whose projected shadow gives an interesting
30 gradient break, particularly in an uncontrolled outdoor environment.

31 *1.1. Related Works*

32 Some recent works on leaf segmentation were found in controlled illumina-
33 tion environments with limited occlusion between individuals. Especially on
34 an open dataset of Arabidopsis Thaliana (Scharr et al., 2016). Other studies
35 related to biomedical imagery and nuclei segmentation were also found with
36 pixel-wise instance segmentation. These studies show the important of defining
37 one or more edges classes and a relevant loss function. These two factors are
38 both related to the weight given to each sample of the training dataset in the
39 estimated error for the optimization algorithm. But the way of how parts of the
40 network are dedicated to the edge classification and thus on how the network
41 focuses on instance separation. Edge detection therefore plays a significant role
42 in pixelwise instance segmentation task.

43 The first studies were dedicated to the definition of specific class of edges to
44 separate instances. Thus Chen et al. (2016) proposed a novel approach named
45 “Deep Contour-Aware” based on the semantic segmentation of two classes, one
46 class is dedicated to the inside of glands, while the second is for the segmenta-
47 tion of glands edges. A bit later, Morris (2018) was the first to define a pixelwise
48 instance segmentation for dense leaf detection. They proposed the “Pyramid
49 CNN for Dense Leaves” architecture which is similar to U-Net (the most com-
50 mon CNN used for biomedical images segmentation). The network is dedicated
51 to the detection of leaf boundaries. To facilitate the learning of edges at differ-
52 ent scales, an auxiliary loss function is placed at each sub-scale of the pyramid.
53 Finally, the instances are retrieved by using a superpixel algorithm.

54 Cui et al. (2019) enhance the “Deep Contour-Aware” model proposed in

2016 by using a real U-Net architecture and data augmentation. Concerning
edge classes based instances segmentation, [Bell and Dee \(2019\)](#) study shows
the importance of separating edges into two classes. As the outer edges are
dominant in the samples, the corresponding error on contiguous edges are less
important. Thus, the edges of leaves are separated into two classes, one for the
outer edges and one for inner edges (when leaves are touching or overlapping).
The multi-scale loss function is still used and proposed by [Xie et al. \(2020\)](#)
for nucleus instance segmentation. They show that multi-scale loss helps to
regularize the network and narrow down the perceptual distances and enlarge
the semantic dissimilarity between individuals. In addition, a count ranking loss
is used on the last feature layer of a fully-connected layer. This count ranking
loss enforces the network to focus on the learning of samples containing crowded
nuclei. This technique results in an implicitly regularized trained network, to
be aware of individuals quantity.

1.2. Objectives

All these studies show that pixelwise instance segmentation technique is
viable but still limited. They also underline the importance of choosing a loss
function adapted to the defined semantic classes. Based on these related works,
this paper proposes to merge most recent advances in the field of pixelwise
instance segmentation.

First, the proposed method combines “deep contour aware” (to separate the
inner of big leaves from its edges), “Leaf Segmentation trough classification of
edges” (to separate instances with a specific inner edges) and “Pyramid CNN
for Dense Leaves” (to consider edges at different scales). Second, a new loss
function is also introduced to limit under and over-segmentation. Third output
is refined using a specific vegetation index based on previous work ([Vayssade
et al., 2021](#)) and a watershed algorithm.

82 This method is applied to dense leaf segmentation, on images containing
83 mixes of plant species and acquired in natural light. A specific multispectral
84 dataset has been acquired, it is presented in the next section 2.1 and released
85 publicly. The method is also evaluated on two common online leaf RGB image
86 databases (Scharr et al., 2016), presented in section 2.2.1 and 2.2.2.

87 The proposed method is much more robust compared to previous pixel-
88 wise instance segmentation methods and solves the issue of methods based on
89 bounding-box regression : all small and thin elements are detected.

90 2. Material and data

91 2.1. Specific multispectral dataset

92 2.1.1. Experimental plot

93 The data were acquired at the site of INRAE (Figure 1) in Montoldre (Allier,
94 France, at 46°20'30.3"N 3°26'03.6"E) within the framework of the "RoSE chal-
95 lenge" founded by the French National Research Agency (ANR) in 2019. The
96 aim of the Challenge is to objectively compare the solutions proposed by partic-
97 ipants for intra-row weed control (Avrin et al. (2020)). Within this context, the
98 challenge provides to contestants an evaluation plan and a set of experimental
99 plots of bean and maize plants. In addition various natural weeds (yarrows,
100 amaranth, geranium, plantago, etc) and sown ones (mustards, goosefoots, may-
101 weed and ryegrass) are managed to compare performances.



Figure 1: Aerial view of the experimentation plot located in Montoldre (now INRAE)

102 *2.1.2. multispectral camera*

103 The images were acquired with the Airphen (Hyphen, Avignon, France) six-
 104 band multispectral camera (visible on the upper-left of the Figure 2). This is
 105 a scientific camera developed by agronomists for agricultural applications. The
 106 camera embeds six sensors using six bandpass (450/570/675/710/730/850 nm)
 107 filters with a 10 nm FWHM each. The focal length of each lens is 8 mm. The
 108 raw resolutions for each spectral band are 1280×960 px with 12 bit precision.
 109 Due to the conception of the camera, spectral images are not aligned. Based
 110 on previous work (Vayssade et al., 2020), a method for registration has been
 111 developed with a registration accuracy down to sub-pixel. After the registration,
 112 all spectral images are cropped to 1200×800 px and concatenated to channel-wise
 113 where each dimension refers to a spectral band.

114 *2.1.3. Image acquisition and annotation*

115 From the presented experimental plots, a set of images were acquired. The
 116 camera is attached in front of an hybrid autonomous tractor called "TREK-
 117 TOR" launched by SITIA Company (Bouguenais, France) in 2019. The camera
 118 is setup to have a top-down view of crop rows, thus it is mounted on a pole in

119 front of the platform allowing to remove visible part of the robot and at 1.8 m
120 from the ground. The Figure 2 below shows the arrangement of the elements.
121 Crops and weeds were between phenological state 3 and 4 which means they
122 have between 2 and 6 leaves. The ground truth is defined on images by ex-
123 perts with polygons around each leaf boundary. In addition, polygons contain
124 a label for their classification between crop and weed (not used in this study).
125 The annotation was defined using the VIA annotation software (Dutta and Zis-
126 serman, 2019) and a total of 300 images of bean were annotated, 170 from
127 acquisition made in June and 130 in October. These dataset is freely available
128 at <https://doi.org/10.15454/JMKP9S>.



Figure 2: The experimental set-up : the multispectral camera and the robotic platform

129 *2.2. Online image databases*

130 *2.2.1. Additional data from the Computer Vision Problems in Plant Phenotyp-*
131 *ing dataset (CVPPP)*

132 To compare the method proposed in this study with others, an additional
133 dataset was used. This dataset is proposed for the Leaf Segmentation Challenge
134 (LSC [Scharr et al. \(2017\)](#)) provided by the International Network on Plant
135 Phenotyping (INPP), a very popular challenge for data scientists. It is composed
136 of RGB images of Arabidopsis Thaliana (783 images) and Rosette (27 images)
137 plants segmented into several leaves. The authors state that the images were
138 collected from multiple locations in a growth chamber experiment and divided
139 into four groups, named A1 through A4. In addition, the dataset is composed
140 of various image sizes (respectively 530×530 , 565×565 , 2048×2048 , 441×441
141 for each sub-dataset), which have been resize to $512 \times 512 \times 3$.



Figure 3: Example of images from the CVPPP dataset (top) and their corresponding ground truth (bottom)

142 *2.2.2. Additional data from Komatsuna dataset*

143 Similar to the LSC dataset, Komatsuna dataset consists in RGB images of
144 plants taken in a top view. This dataset contains a large number of plant growth

145 stages, it was designed to solve the problem of 3D phenotyping (Uchiyama et al.,
 146 2017). It is used here primarily as a test set due to its similarity to the CVPPP
 147 dataset (training set), as shown in Figure 4. This dataset includes 900 images
 148 of size $480 \times 480 \times 3$.

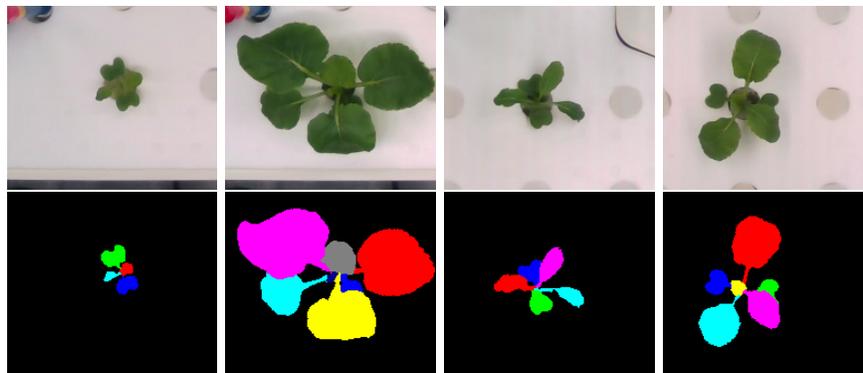


Figure 4: Example of RGB images from the Komatsuna dataset (top) and their corresponding ground truth (bottom)

149 3. Methodology

150 We consider the leaf segmentation as a binary segmentation problem of
 151 boundaries as proposed by Morris (2018). The main idea is to detect the sharp
 152 edges of leaves or to follow the projected shadow of a leaf on the one below it.
 153 This methodology section is split into three sub-tasks (3.1) the proposed CNN
 154 architecture to detect and separate leaves, (3.2) the specific loss function defined
 155 to limit under and over-segmentation, and (3.3) a simple watershed algorithm
 156 which takes the CNN output and a vegetation mask to refine the segmentation.

157 3.1. Proposed CNN architecture

158 Unlike recent CNN architectures, the proposed approach is slightly more de-
 159 composed like standard biomedical and agricultural computer vision pipelines
 160 (Perez-Sanz et al., 2017; Lottes and Stachniss, 2017). Thus, the architecture

161 (Figure 5) is composed of three **upstream modules** (IIT, IBF, UFA) that im-
 162 proves the input data and eliminates unnecessary information. This step com-
 163 posed of 3 **upstream** modules, was proposed in a previous work to construct a
 164 vegetation index (Vayssade et al. (2021)). It is used to identify relevant spectral
 165 features on the input data to exploit the inter-channel relationships. After this
 166 stage, a **core network** is used to consider spatial information at different scales
 167 on the image, the core network returns four down-scaled feature maps. Finally,
 168 at the end of the network, three **downstream modules** (CoordConv, UFA,
 169 Classification) are proposed to fuse spectral and spatial information. Sigmoid
 170 activation function is used at the end of all convolution layers to learn more
 171 complex structures and allows non-linearity of the reconstructed function.

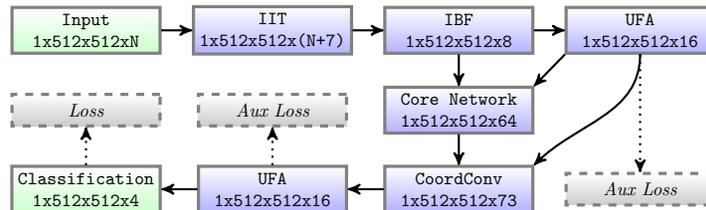


Figure 5: Diagram of the CNN network architecture and losses (dotted). Multiple arrow show concatenation as input layer.

172 The network takes an input image of size $512 \times 512 \times 6$, thus the learning
 173 and computation are done by a sliding window within the registered images of
 174 the Airphen camera of size $1200 \times 800 \times 6$. The output layer is defined as a
 175 semantic segmentation of four classes. One class is dedicated to the inner of
 176 individuals, while three classes are dedicated to the detection of edges to keep
 177 aware of leaves instance. As mentioned by Bell and Dee (2019), one class is
 178 dedicated to outer of leaf (touching soil texture), while the second is dedicated
 179 to inner edges (touching/overlapping another leaf). Within our dataset, a third

180 class appears with thin leaves which can be considered as a kind of edge. The
 181 ground truth is a set of polygons, drawn with the respect of these classes. Edge
 182 classes were empirically drawn with three pixel thickness. The following figure
 183 shows the input ground truth.

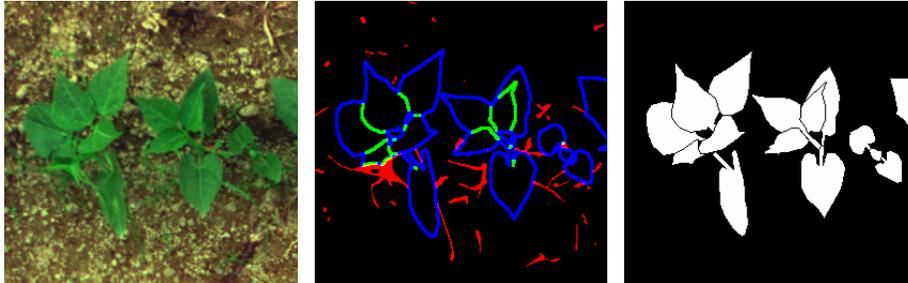


Figure 6: Example of input data: from left to right, the first three bands (RGB) of the Airphen multispectral camera, three edge classes and the inner of individual class

184 3.1.1. Upstream of the network

185 It is composed of three **upstream** modules: Initial Image Transforms (IIT),
 186 Input Band Filter (IBF) and Universal Function Approximator (UFA).

187 *Initial Image Transforms (IIT)*. In order to enrich the pool of information,
 188 spectral band transformations are added to take into account specific spatial
 189 gradients in the image and spectral mixing. Seven important transformations
 190 are considered. The standard deviation between spectral bands, noted ρ_{std} can
 191 help to detect the spectral mixture. For example, between two different surfaces
 192 like ground and leaf (which have opposite spectral radiance), the spectral mixing
 193 makes a pixel with linear combination, thus the standard deviation tends to zero
 194 (Louargant et al., 2017). Three Gaussian derivatives on different orientations
 195 are computed. Gxx, Gxy and Gyy filters on ρ_{std} give an important spatial
 196 information about the breaks of gradient, and therefore about the outer limits
 197 of surfaces. The Laplacian, the minimum and maximum eigenvalues (of the
 198 Hessian matrix) also called ridge of the ρ_{std} seems to easily detect fine elements

199 (Lin et al., 2014), such as monocotyledons for vegetation images. All these
200 transformations are concatenated to the channel-wise normalized spectral band
201 input and build the final input image. In the end we have six spectral images
202 plus seven transformations for a final image composed of 13 channels.

203 *Input Band Filter (IBF)*. To remove unneeded parts of the signal, low-pass,
204 high-pass and band-pass filters are added. To apply the low-pass filter we use
205 the equation $z = \max(x - a, 0)/(1 - a)$ and thus it allows to suppress low values.
206 For the high-pass filter we apply the equation $w = \max(b - x, 0)/b$ to suppress
207 the high values. The band-pass filter is the product of low-pass and high-pass
208 filters defined by $y = z * w$. The output layer is the concatenation in the channel-
209 wise of the input images, the low-pass, the high-pass and the band-pass filter
210 which produce $4 * 13 = 52$ channels. Finally, to reduce the output data for the
211 rest of the network, a bottleneck is inserted using a 3×3 convolution layer, and
212 it generates a new tensor with 16 channels.

213 *Universal Function Approximator (UFA)*. To separate efficiently leaves from
214 the background and to learn spectral features, a Universal Function Approximator
215 is defined on the upstream of the network (Figure 7). The UFA is based
216 on Taylor expansion theorem, an approach to learn this form of development
217 in deep-learning is called DenseNet and then corresponds to the sum of the
218 concatenation of the signal with these spatio-spectral derivatives. This was
219 successfully used for vegetation segmentation Vayssade et al. (2021). Three pa-
220 rameters, such as the *depth* (number of convolutions), the *width* (number of
221 filters denoted W) and k (kernel size) configure the network and were empiri-
222 cally fixed to $depth = 3$, $width = 16$ and $k = 1$. An auxiliary output is used
223 here to maximize the class similarity on the upstream of the network and to
224 extract pure spectral information.

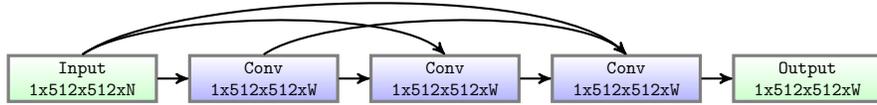


Figure 7: Universal function approximator based on DenseNet. Multiple arrows shows concatenation as input layer

225 *3.1.2. Core network model*

226 The proposed core CNN architecture is based on an advanced U-Net archi-
 227 tecture named MFP-Unet (multi-feature pyramid U-net) proposed by [Moradi](#)
 228 [et al. \(2019\)](#). It is a neural network composed by several 2D convolution layers
 229 between different sub-scales. At each sub-scale, the spatial size is divided by
 230 two and the number of filters is multiplied by two. Sub-scales are obtained by
 231 Max-Pooling layers. Then, to retrieve the original size a 2D UpSampling layer
 232 is used. In this study the depth of the U-Net model is fixed to three down-scale
 233 (size 512, 256, 128, 64). The specificity of MFP-Unet is that all sub-scale feature
 234 maps are directly up-sampled to the initial size, concatenated to the channel-
 235 wise and then used for the classification (Figure 8). In addition according to
 236 [Morris \(2018\)](#) and [Xie et al. \(2020\)](#) an auxiliary loss function is put at each
 237 sub-scale feature layer and it enforces the learning of edges at different scale,
 238 making the network more robust to spatial resolution. Losses at each predic-
 239 tion also shorten the back propagation path leading to faster convergence. All
 240 convolution layers use a kernel size of 3×3 and are followed by a Batch Nor-
 241 malization and a sigmoid activation function ([Moradi et al. \(2019\)](#); [Nwankpa](#)
 242 [et al. \(2018\)](#)).

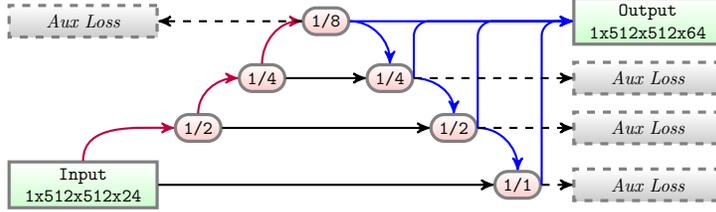


Figure 8: Synthesis of the core network based on MFP-Unet. Red arrows shows MaxPooling. Blue arrows shows Conv+UpSampling. Black arrows shows Conv. Sub-scale ratio are labelled on corresponding layer. The final output is the concatenation of features UpSampling. Multiple arrows shows concatenation as input layer

243 3.1.3. Downstream of the network

244 It is composed of three **downstream** modules: CoordConv and Universal
 245 Function Approximator (UFA) and Classification.

246 *CoordConv.* The core network model produces a concatenation of 4 layers of 16
 247 features ($4 \times 512 \times 512 \times 16$) which results of a layer of size $1 \times 512 \times 512 \times 64$.
 248 A coordinate layer (Liu et al., 2018) is also concatenated and allows to consider
 249 the mapping between the coordinates in (x,y) Cartesian space to one-hot pixel
 250 space. Three variables are appended, the normalized x and y coordinate and the
 251 radial coordinate $\sqrt{(x - 0.5)^2 + (y - 0.5)^2}$. This module improves the results
 252 removing noise, ground moisture and it fixes few small holes in the segmentation
 253 mask.

254 *Universal Function Approximator (UFA).* This UFA – also presented on the
 255 upstream – is used to accurately mix features coming from various scales as
 256 well as the Cartesian coords. This module reconstructs the mapping function
 257 from the Cartesian space to a spatio-spectral feature of size $1 \times 512 \times 512 \times 16$.
 258 However, in contrast with the upstream UFA, the kernel size k was fixed to 3
 259 to take into account neighboring.

260 *Classification and Auxiliary output.* The classification is done through a small
 261 network composed of two 1×1 convolution layers. Followed by a “Pyramid
 262 Pooling Module” to consider different scales before the outputs and smooth the
 263 boundary prediction. Zhao et al. (2017) showed that fusing the low to high-level
 264 features improved the segmentation task. It consists in the sum of different 2D
 265 convolutions whose kernel sizes have been set to 3, 5, 7 and 9. The number of
 266 filters is the same as the number of classes : 4 (defined in 3.1). The result of
 267 each convolution is concatenated and the final image output is given by a 2D
 268 convolution. In addition all convolutions are followed by a Batch Normalization
 269 and a sigmoid activation function. The figure 9 shows that sub-network.

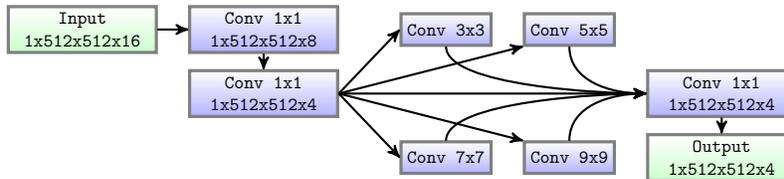


Figure 9: Auxiliary output and classification module. Multiple arrows shows concatenation as input layer

270 3.2. Loss function

A wide variety of loss functions have been developed during the emergence of deep-learning. Recently, Rahman and Wang (2016) proposed a solution to optimize an approximation of the mean Intersection over Union (mIoU) which seems to be optimal for binary segmentation (Zhou et al., 2019). The loss function using ground truth (p) and the prediction (\hat{p}) is defined by:

$$\text{mIoU}(p, \hat{p}) = 1 - \frac{p\hat{p}}{p + \hat{p} - p\hat{p}} \quad (1)$$

This loss function was used on each auxiliary. In addition the loss is computed separately on each class, weighted (denoted W_C) and summed. The result

of this function is:

$$\text{Aux}(p, \hat{p}) = \sum_{C=0}^4 W_C \times \text{mIoU}(p_C, \hat{p}_C) \quad (2)$$

271 In the above equation, the weight W were empirically set to $[0.175, 0.526, 0.211, 0.09]$.
272 Meaning that the second class (inner edges) is prioritized (to separate inner in-
273 stance). Then it is outer edges that allow to separate the “big” leaves from
274 small or thin leaves. Finally the thin leaves (mostly spectral mixing) and inner
275 big leaves (essentially vegetation minus boundaries) have the smallest weight
276 values because these classes should be easier to learn.

277 In recent CNN architectures for instance segmentation, the loss function
278 does not take into account the number of detected instances or the shape of
279 the segmentation. This aspect is only evaluated after learning, e.g., using a
280 symmetric best dice metric. This implies that we can not guaranty that the
281 network is well learned on crowded scene, where instance is generally merged.
282 One problem is that until recently, instances could not be retrieved directly dur-
283 ing the learning phase, this is due to the “non-maximal suppression” algorithm
284 required after the CNN or the time required for the association between the
285 detected instances and the ground truth. In this paper, we introduce a new loss
286 function considered at the downstream of the network. The main idea is to take
287 into account an approximation of the segmentation quality of each leaf.

288 To estimate the segmentation quality, the undetected, under/over segmented
289 and fused objects can be evaluated, trough a sorted histogram of the number of
290 pixels associated to each connected component for both prediction and ground
291 truth, as showed in the next figure 10 for both prediction (orange) and ground
292 truth (blue).

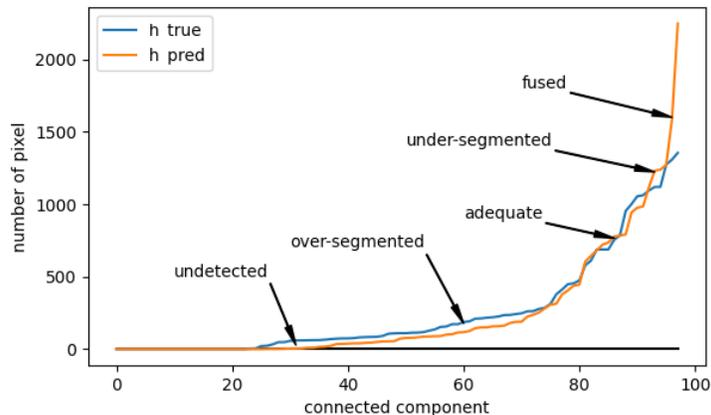


Figure 10: Sorted histogram of the number of pixels associated to each component, the blue (resp. red) line represents the true (resp. predicted) number of pixels for each component.

293 This figure shows that when the prediction curve is lower (orange) than the
 294 ground truth curve (blue), it means that there is over-segmentation. This arise
 295 when a bigger element is split in two, when the borders of the shape are trimmed
 296 or when zero pixels of the shape are detected (undetected). On the other hand,
 297 when the prediction curve is higher than the ground truth, it means that the
 298 contours of the shape are roughly detected or if it greatly exceeds the ground
 299 truth, then we are in the presence of fused shapes. Based on these curves, a loss
 300 function can be construct the deal with over and under segmentation on shape
 301 criterion, thus a custom absolute percentage error was defined:

$$\text{Leaf}(p, \hat{p}) = \frac{\sum_{i=0}^N |\text{leaky_relu}(h_i(p) - h_i(\hat{p}))|}{h_{max}(p) + 1} \quad (3)$$

$$\text{leaky_relu}(x) = \begin{cases} x & \text{if } x \geq 0 \\ x * 0.2 & \text{if } x < 0 \end{cases} \quad (4)$$

302 On the above equation N is the number of components, $h_i(p)$, and $h_i(\hat{p})$ is
 303 respectively the number of pixels of a connected component in the ground truth

304 and on the predicted segmentation within the sorted histogram. While $h_{max}(p)$
305 is the maximum number of pixels of a component in the ground truth.

306 The leaky_relu, is used to explicitly prioritise the learning on under-segmentation
307 rather than over-segmentation which allows to prioritize the merged objects.
308 This was defined because conventional losses did not give good results in dense
309 vegetation cover, causing a large segmented area that is detected as a single en-
310 tity. Note also that over-segmentation is less problematic, since it occurs mainly
311 around the borders of the leaves, which are recovered later, through a watershed
312 algorithm (section 3.3). It is the first study to suggest this type of loss.

313 The downstream loss is defined by $\text{DownAux}(p, \hat{p}) = \text{Aux}(p, \hat{p}) + \text{Leaf}(p, \hat{p})$.
314 Finally the global loss considers the upstream auxiliary loss, each of the 4 feature
315 scale auxiliary loss and the downstream loss. Thus the global loss is defined as
316 the weighted sum of all auxiliaries losses (in the same order) where the weights
317 W were empirically defined with $W = [0.01, 1.0, 0.1, 0.1, 0.01, 4.0]$.

318 3.3. Refinement with vegetation mask and watershed

319 The used U-Net architecture is good at detecting “big” elements on the scene
320 but lacks precision on very small and thin elements. A method to produce an
321 optimized vegetation mask was proposed in a previous work [Vayssade et al.](#)
322 (2021). Using this mask provides better performance than adding a specific
323 class to the network. Thus our previous work is used here to get the best
324 foreground/background segmentation mask as input of the watershed. It is also
325 learned on a specific dataset with more illumination conditions and should be
326 more robust, especially for thin elements.

327 The proposed network generates 4 classes. The first two are associated to
328 “big” leaf boundaries (denoted $Edges = Outer + Inner$). The third one is small
329 and thin leaves (denoted $Thin$), and the fourth is the inner of big leaves denoted
330 Big . The watershed seed is defined with the following equation 5 to generate

331 the seed mask and can be seen in the figure 11 :

$$Seed = Thin + Big - Edges \quad (5)$$

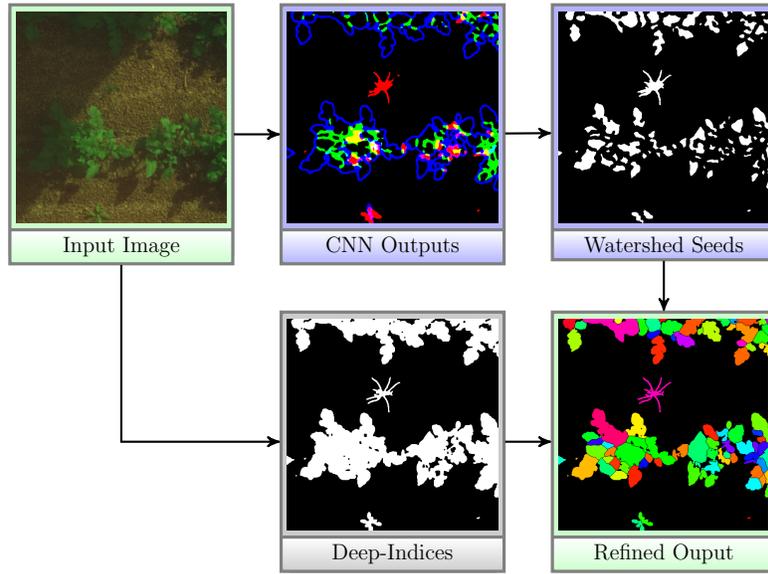


Figure 11: The refinement of the CNN output through watershed and Deep-Indices. The Seed of the equation 5 can be seen on the “watershed seed” image.

332 3.4. Training setup

333 The image dataset is randomly split into a training set (80%) and a validation
 334 set (20%). However the initial seed is kept for reproducibility. In addition a
 335 random data augmentation is used during the training to increase the dataset
 336 variability. A random vertical and horizontal flip is considered as well as a
 337 perlin simplex noise [Bae et al. \(2018\)](#) (of size 512×512), set with 2 modal in
 338 range $[0.7 - 1.3]$ which multiplies the number of input images. Low values add
 339 shadows, while high values add specular effects. The training is done through
 340 Keras module within Tensorflow 2.6.0 framework. All computations are done

341 on an NVidia RTX 3060 which have 12GiB of memory and due to the size of
342 the network only one image is computed at once.

343 3.5. Evaluation metrics

344 There is a large number of possible evaluation metrics for instance segmen-
345 tation, which have been reviewed in [Scharr et al. \(2016\)](#). However, we de-
346 cided to keep the evaluation metrics used in the Leaf Segmentation Challenge
347 (LSC) as a reference for comparison ([Scharr et al., 2017](#); [Kulikov et al., 2018](#);
348 [Bell and Dee, 2019](#); [Ward and Moghadam, 2020](#)). Therefore, we use the fore-
349 ground/background DICE metric to evaluate the separation of large leaves from
350 the ground (denoted *FgBgDice*). We estimate the average accuracy of leaf seg-
351 mentation by the symmetric best DICE score among all objects (leaves). The
352 best DICE score among all objects (leaves) to estimate the average accuracy of
353 leaf segmentation is denoted *BestDice*. The *AbsDiffFG* estimates how good
354 the algorithm is at segmenting the leaves. And finally, *DiffFG* estimates the
355 efficiency of the algorithm for counting leaves. *SBD* for Symmetric best DICE
356 is extract to estimate the average leaf segmentation accuracy. All these metrics
357 are common and presented by [Scharr et al. \(2016\)](#). In addition, to compare
358 datasets results, a new metric is introduced named *NAbsDiffFG* defined by
359 the division of *AbsDiffFG* and the mean of leaves count in the dataset.

360 4. Results

361 As previously defined, all datasets were split into training and validation
362 sets with a ratio of 80 – 20%. Once training is done on the defined setup and
363 using the loss function, the evaluation metrics are extracted : the *FGBGDice*
364 metric is used to evaluate the soil versus vegetation segmentation, while the
365 *BestDice* metric is used to evaluate the instance segmentation. Each connected
366 component is associated to its best corresponding ground truth based on a dice

367 score. Then the metric is defined by the mean dice score of the best match.
 368 Other metrics show relevant information about over and under segmentation.
 369 The following subsections are dedicated to each dataset, from the simplest to
 370 the most difficult to segment.

371 4.1. *Komatsuna dataset*

372 The first one is the Komatsuna dataset. It is composed of RGB data of
 373 growing Japanese Mustard. This dataset is interesting for its controlled illumi-
 374 nation conditions. In addition, the number of leaves is quite the same for all the
 375 growing stages, ranging from 3 to 6 leaves. This is important regarding our loss
 376 function which takes into account the quantity and quality of each segmented
 377 leaf. However, this dataset does not contain small and thin leaves, thus the
 378 third class was replaced by a foreground class. The following Table 1 shows the
 379 evaluation on this dataset.

metric	training	validation
FgBgDice	0.9732	0.9715
BestDice	0.8796	0.8565
SBD	0.8713	0.8517
DiffFG	-0.2639	-0.4444
AbsDiffFG	0.3944	0.5222
Number of images	720	180
Mean leaf count	4.9194	4.9722
NAbsDiffFG	0.0802	0.1050

Table 1: Evaluation metrics for the Komatsuna dataset

380 The result on this dataset shows a relatively perfect score of the soil and
 381 vegetation segmentation with a *FGBGDice* of 0.97. Few errors remain as shown
 382 in figure 12, a green bottle is visible on the left side of some images, which
 383 seems to indicate that the green component has a major impact on vegetation
 384 detection. Since RGB data are used, this error should be visible in most studies.
 385 This interesting element also demonstrates that the used *CoordConv* layer does
 386 not play its role in the spatial management of this element.

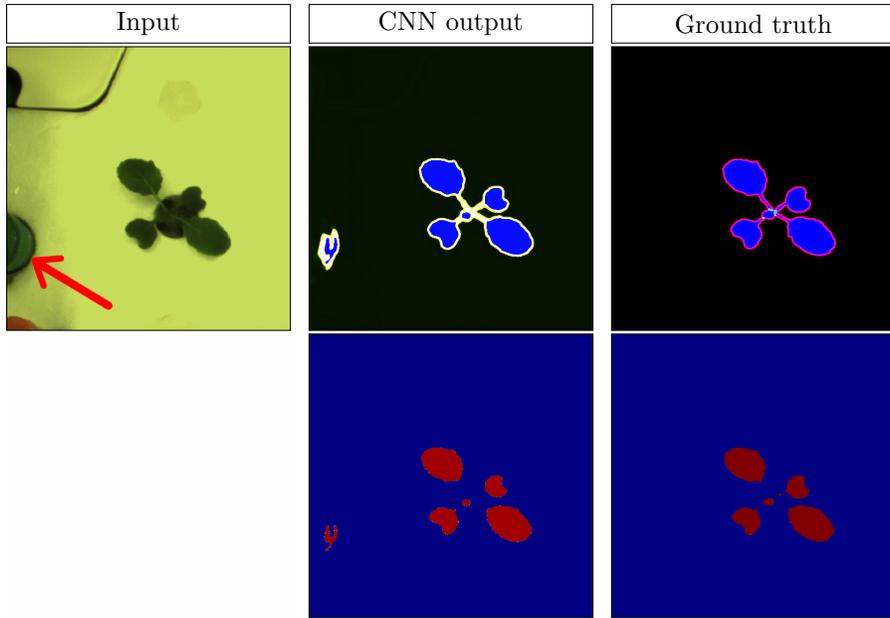


Figure 12: Visual comparison of Komatsuna dataset after the training, the red arrow in the input image shows a green bottle

387 As shown by the *DiffFG* score, our method mostly detects the right number
388 of leaves. This score is small and negative and shows that few leaves are split,
389 which occurs when a big leaf mostly overlaps a smaller one. A visible bottle
390 also has an impact on the other metrics, shown in figure 12. However, most
391 of the errors for the *BestDice* comes from under and over-segmentation. The
392 under-segmentation occurs on the leaves stems. The stem that connects the
393 leaf to the plant is usually not well detected, and this can play a significant
394 role in lowering the *BestDice* and *SymmetricBestDice* scores. The stem can
395 be undetected (under-segmentation) or may be identified as another leaf (over-
396 segmentation). It can also cover some small leaves and divide them in two, which
397 is expected in pixelwise segmentation. To benchmark our study, the following
398 table 2 reports the results of few previous studies using the same metrics.

Study	SBD	AbsDiffFG
Our method’s results (2021)	0.8565	0.5222
Ward and Moghadam (2020) ward 2020	0.7776	-
Gomes and Zheng (2020) gomes 2020	0.7700	0.8800

Table 2: Comparison of the ResNet 101 based on Detectron II framework of facebook and the UPGen based on Mask-RCNN. Our solution is the most effective in both metrics

399 As comparison our results tackle one of the states of the art ResNet 101
400 based on Detectron II framework of facebook ([Gomes and Zheng, 2020](#)). And
401 the proposed UPGen methods proposed by [Ward and Moghadam \(2020\)](#).

402 4.2. Leaf Segmentation Challenge dataset

403 The second dataset is interesting for its controlled illumination conditions.
404 Unlike the Komatsuna dataset, this one includes a greater diversity of leaf quan-
405 tity, between 2 and 16 leaves. This is important to show the robustness of our
406 loss function. In the same regard as the Komatsuna section, this dataset does
407 not contain small and thin leaves. Thus third class was replaced by a foreground
408 class, which is also used by the competition evaluation. In addition, the testing
409 dataset is not publicly available, thus a separated evaluation was performed on
410 the online competition website ¹. The training and validation are defined by
411 splitting the publicly available dataset. This step is used to reduce the over-
412 fitting and retrieve stable parameters. The next table summarizes the number
413 of images for each training, validation and test datasets.

dataset	A1	A2	A3	A4	total
training	102	25	22	499	648
validation	25	6	5	125	161
test	33	9	56	168	266

Table 3: Number of images for each sub LSC dataset

414 The following Table 4 shows the overall evaluation on this dataset (merged

¹competitions.codalab.org/competitions/18405

415 A1, A2, A3, A4). The test results can be found in the online leader-board. The
 416 results are fairly the same between training, validation and test datasets in term
 417 of *FGBGDice* and *BestDice*. However a notable divergence with the testing
 418 dataset for the others metrics is visible. The *DiffFG*, and *AbsDiffFG* for
 419 training and validation indicate an over-segmentation, while the test dataset
 420 highlights an under-segmentation. This over-segmentation probably enhances
 421 the *SBD* score while the *BestDice* remains stable.

metric	training	validation	test
FgBgDice	0.9489	0.9522	0.9489
BestDice	0.7659	0.7707	0.7608
SBD	0.7608	0.7655	0.8047
DiffFG	-1.9634	-2.1223	3.5628
AbsDiffFG	2.1707	2.3050	6.1257
Number of images	648	161	266
Mean leaf count	13.8390	13.1463	-
NAbsDiffFG	0.1568	0.1753	-

Table 4: Evaluation metrics of the overall LSC dataset

422 As discussed in the materials and data section, this database is composed of
 423 four sub-databases with different cameras and plants. The learning was done
 424 independently on each of them, but they were merged for the presentation in the
 425 previous table. The test on the online evaluation platform returns the results for
 426 each of them, summarized in the table 5. First of all, the sub-datasets A2 and
 427 A3 contain a very small amount of images, respectively 25 and 22 for the training
 428 sets. Moreover the A3 dataset is composed of images of size 2048×2048
 429 rescaled to 512×512 . This imply a huge loss of information, especially on the leaf
 430 boundaries. In addition, this A3 dataset contains hard shadows. These three
 431 factors explain most of the errors for this dataset. The quality, acquisition
 432 conditions, and plants of the other three data sets (A1, A2, A4) are similar.
 433 Only the background and the amount of images mainly changes, which is also
 434 reflected in the evaluation score. The figure 16 also shows unlabeled leaves in

435 the center of the plant, moreover they are even defined as background.

metric	A1	A2	A3	A4
FGBGDice	0.9641	0.9294	0.9692	0.9416
BestDice	0.6905	0.6944	0.6783	0.7964
SBD	0.8047	0.7895	0.7317	0.8294
DiffFG	2.0909	1.7778	21.2857	0.0892
AbsDiffFG	3.0000	3.1111	14.9285	1.6250

Table 5: Evaluation metrics of the independent sub LSC dataset

436 For all sub-dataset the *FGBGDice* score is slightly less than the Komatsuna
 437 dataset. Most pictures of the A1 present wide area of green moisture on the
 438 ground, visible on the Input inside the figure 13. A2 contains few very small
 439 weeds and few small spots of moisture, but the performances of *FGBGDice* is
 440 mainly due to the quantity of images for the learning, as showed by the figure
 441 14 boundaries are miss-classified. For the same reason, it is also visible for the
 442 A3 dataset shown by the figure 15. It’s also important to notice that leaves
 443 from an outside plant are visible in few images and detected by the algorithm,
 444 however these leaves are unlabelled, resulting an over-segmentation.

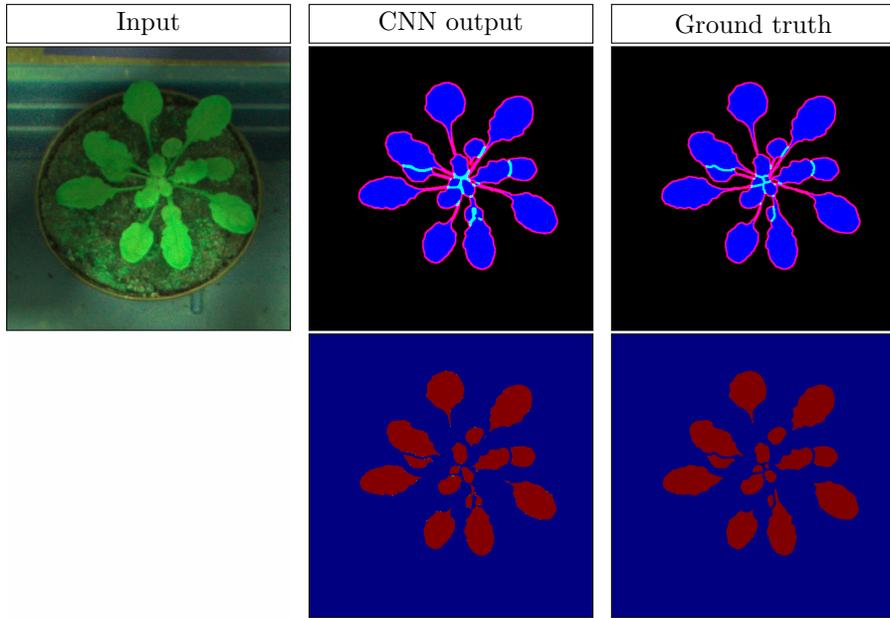


Figure 13: Visual example of LSC results for A1

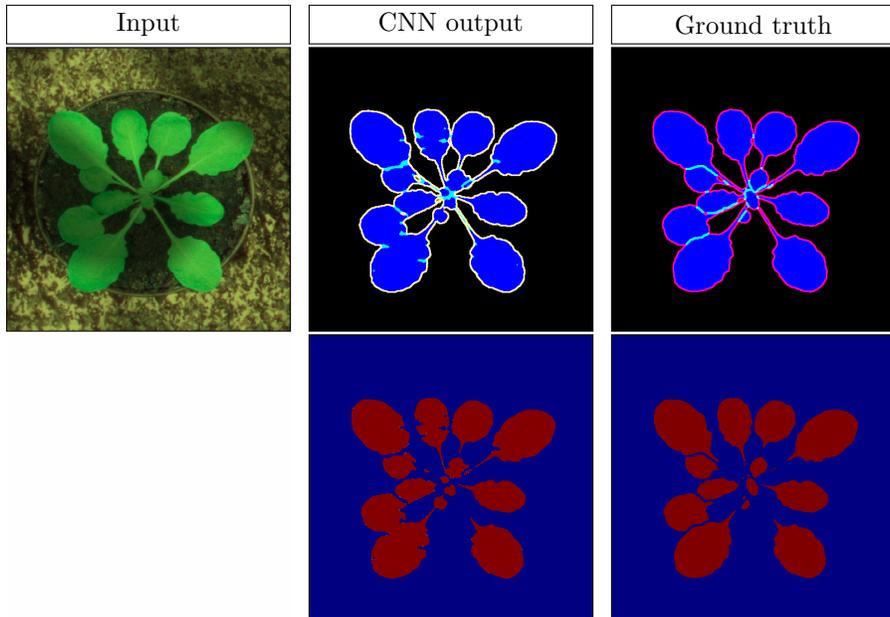


Figure 14: Visual example of LSC results for A2

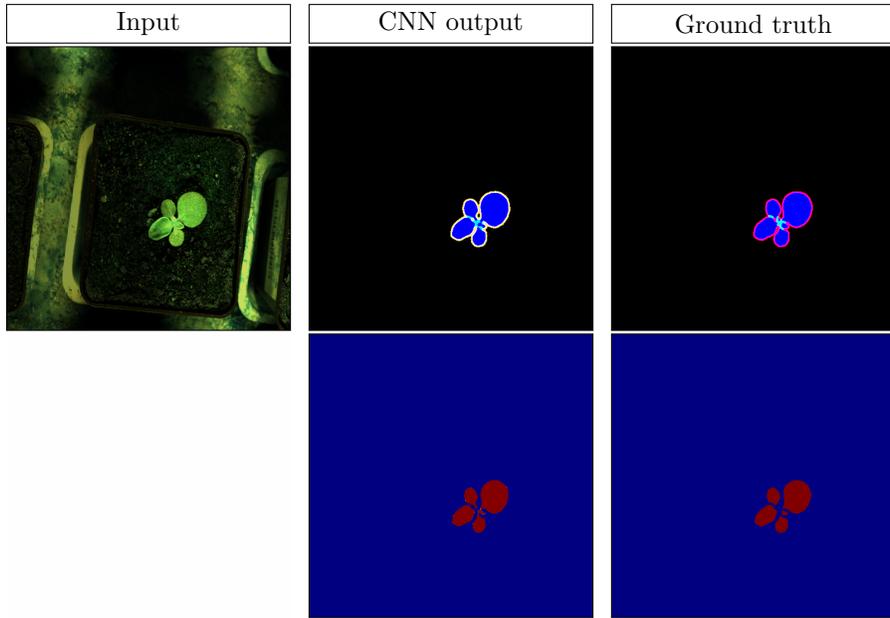


Figure 15: Visual example of LSC results for A3

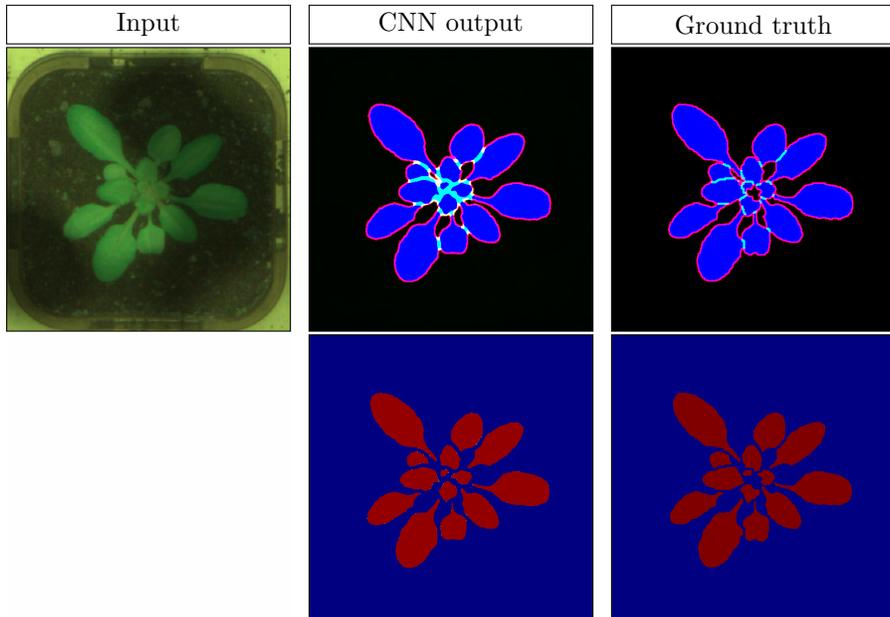


Figure 16: Visual example of LSC results for A4

445 The following table 6 reports method’s results of different previous studies,
 446 to benchmark our method’s results on this LSC dataset.

Study	SBD	AbsDiffFG
Ward and Moghadam (2020) ward 2020	0.8800	-
Kulikov et al. (2018) kulikov 2018	0.8040	2.00
Gomes and Zheng (2020) gomes 2020	0.7700	0.88
Ours method’s results (2021)	0.7608	6.12
Pape and Klukas (2015) pape 2015	0.7440	2.60
Scharr et al. (2016) scharr 2016	0.6830	3.80

Table 6: Comparison of our solution with state-of-the-art challengers in this dataset.

447 These results show that the studied method is less efficient than Ward and
 448 Moghadam (2020); Kulikov et al. (2018) and Gomes and Zheng (2020). Due to
 449 the small size of the training sample (102, 25, 22, 499 images for data subsets A1,
 450 A2, A3, A4, respectively), it can be assumed that this is due to the increase in
 451 data they use in order to expand their data set. It seems the data augmentation
 452 based on Perlin noise is insufficient. Nevertheless, our method is better than the
 453 following two: (Pape and Klukas, 2015; Scharr et al., 2016), since they don’t
 454 use any data augmentation. However it can be noted that the *AbsDiffFG*
 455 value for the studied method is much higher than for the others. This implies
 456 an important over-segmentation, resulting in particular from the quality of the
 457 A3 dataset as it can be seen in the table 5.

458 4.3. Airphen dataset

459 The last one presented is our multispectral dataset, which contains variable
 460 acquisition conditions (sunny, cloudy, rainy, etc.), a variable number of leaves
 461 (from a few to hundreds), and contains very small leaves that touch or overlap
 462 the others. The next table 7 show the results of our method applied our dataset.

metric	training	validation
FgBgDice	0.9791	0.9785
BestDice	0.6744	0.6678
SBD	0.6670	0.6638
DiffFG	-41.3583	-49.4667
AbsDiffFG	46.7500	53.2333
Number of images	240	60
Mean leaf count	265.83	295.80
NAbsDiffFG	0.1758	0.1799

Table 7: Evaluation metrics for the Airphen dataset

463 It can be seen that the *FGBGDice* score shows adequate soil/vegetation
 464 segmentation, as demonstrated in our previous study [Vayssade et al. \(2021\)](#).
 465 The scores *DiffFG* and *AbsDiffFG* show the presence of over-segmentation.
 466 This is due to two issues: the presence of small leaves and the advanced stage of
 467 mustard development. Indeed, the method sometimes confuses small leaves and
 468 large leaves, which implies in some cases under-segmentation for monocotyle-
 469 dons or over-segmentation for dycotyledons, probably generated by an imprecise
 470 annotation of classes. In the second case, this is due to the proposed loss func-
 471 tion that forces over-segmentation, resulting in the detection of leaflets instead
 472 of leaves in the case of advanced mustard development, as showed in the figure
 473 [17](#). These errors imply that the *BestDice* and *SBD* scores are not as good as
 474 on the Komatsuna dataset and LSC Challenge, studied previously.

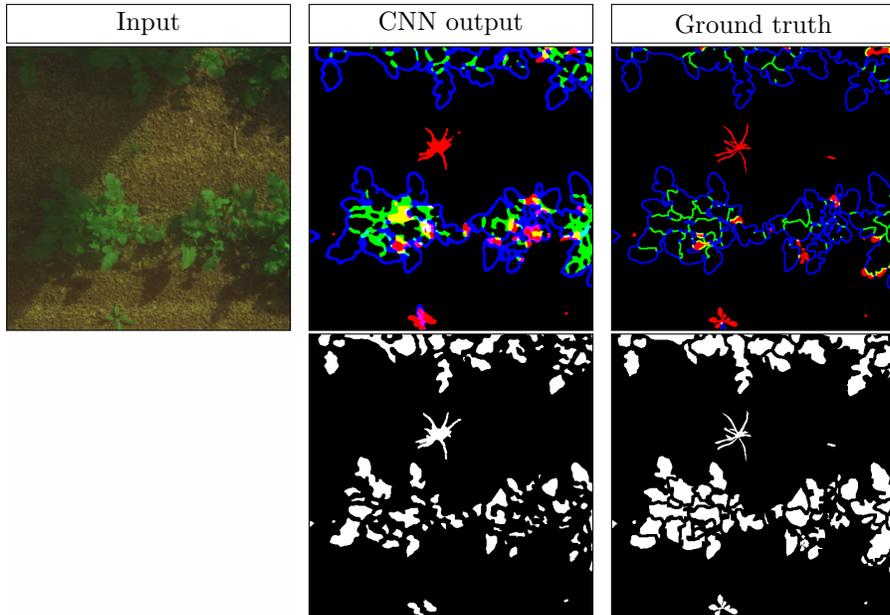


Figure 17: Example of the visual results for the Airphen dataset

475 To show the difference and complexity of this dataset, the method developed
 476 by [Ward and Moghadam \(2020\)](#) was also learned. This method is based on
 477 the Mask-RCNN. However, the method uses data augmentation based on a
 478 library that only supports 8-bit unsigned integers. Thus, data augmentation
 479 was disabled because the dataset uses a 32-bit float format. The following table
 480 8 shows these results.

metric	training	validation
FgBgDice	0.6266	0.6111
BestDice	0.2271	0.2157
SBD	0.2266	0.2149
DiffFG	-186.0292	-217.0667
AbsDiffFG	186.5542	217.1333
NAbsDiffFG	0.7018	0.7341

Table 8: Evaluation metrics for the Airphen dataset using Mask-RCNN

481 These results show that the method proposed by [Ward and Moghadam](#)
 482 (2020) does not correctly detect leaves in a dense foliage environment. The

483 $FgBgDice$ score shows that the soil/vegetation discrimination is weak. As well
 484 as the $BestDice$ and SBD scores which shows poor detection of individuals seg-
 485 mentation mask. The $DiffFG$ and $AbsDiffFG$ scores confirm these results
 486 and show a large amount of undetected elements. As announced in the intro-
 487 duction, the methods based on object detection uses bounding box regressions
 488 and non-max suppression which strongly limit the detection in dense environ-
 489 ment. Moreover the part of the network allowing to obtain the segmentation
 490 mask uses a fixed low resolution resulting in coarse segmentation masks. As
 491 shows in the figure 18.

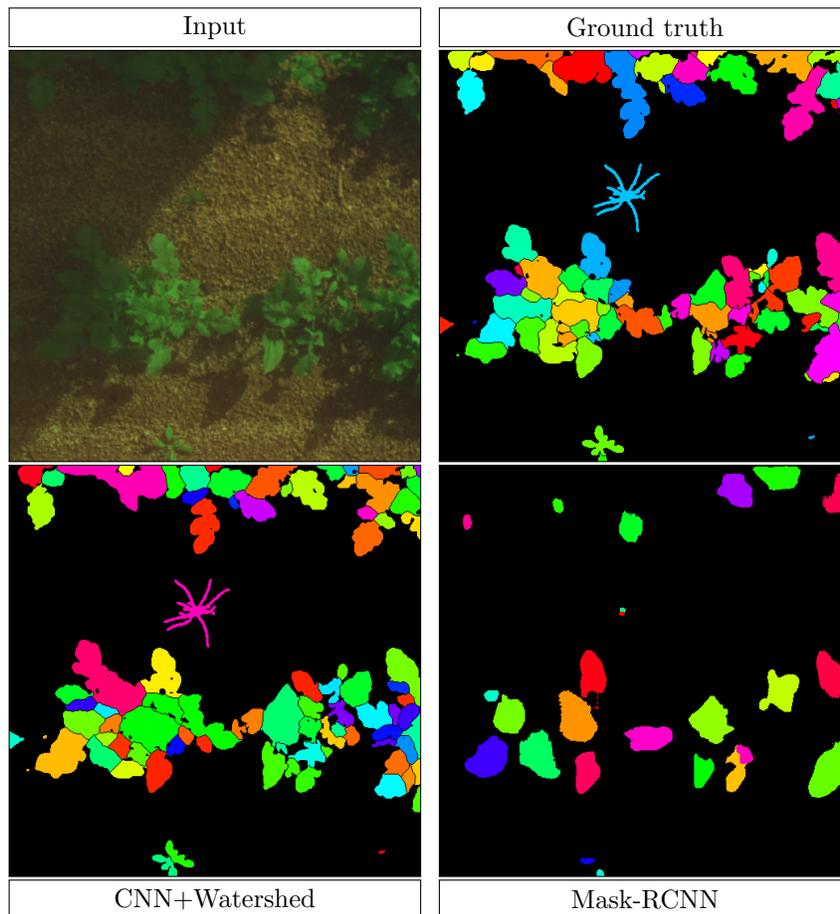


Figure 18: Example of the Watershed output versus Mask-RCNN for Airphen dataset

492 5. Discussion

493 In this study, we used different datasets. The Komatsuna dataset is com-
494 posed only of mustard leaves at different stages of development acquired indoors
495 under controlled acquisition conditions with a single camera. There are between
496 3 and 6 leaves per image for an average of 4.9 leaves. The LSC dataset is de-
497 composed into three sub-datasets of Arabidopsis-Thaliana and one of Rosette
498 plants all at different stages of development. They are acquired indoors under
499 controlled acquisition conditions with their own camera. There are between 2
500 and 16 leaves per image for an average of 13.5 leaves. Our dataset is composed of
501 bean plants and weeds. Beans are at a single stage of development while weeds
502 are at different stages of development. The acquisitions were made outdoors
503 under uncontrolled conditions. There are between 4 and 777 leaves per image
504 with an average of 271.83 leaves. This dataset is now available online² and
505 may help other studies working on leaves segmentation in natural and complex
506 situations.

507 We are interested in the metric $NAbsDiffFG$ to compare the performances
508 of our method on the different datasets. We see that for our dataset it is
509 0.1758, 0.1799 for the LSC dataset and 0.1050 for the Komatsuna dataset. We
510 thus obtain good results for the Komatsuna dataset which is the simplest, we
511 obtain an average result for our dataset which is the most complex and finally
512 weaker results on the LSC dataset. This allows us to deduce that the lighting
513 conditions as well as the number of leaves do not influence the performances.
514 On the other hand, the use of several cameras, as on the LSC dataset, could
515 be at the origin of the decrease in performance due to a variation in spatial
516 resolution from one camera to another. We deduce that our method, designed
517 to be used on complex data such as our dataset, gives results comparable to

²<https://data.inrae.fr/dataset.xhtml?persistentId=doi:10.15454/JMKP9S>

518 classical methods on other types of datasets.

519 The CNN used in our method deliberately gives a coarse segmentation in
520 order to maximize the detection and separation of large leaves. The smaller
521 ones are therefore poorly detected (over-segmentation or undetected). These
522 results are reflected in the fact that there is little over-segmentation and under-
523 segmentation of large leaves due to the loss function introduced which takes
524 into account the number of elements detected and their surface. However two
525 problems are raised with the use of this loss function. First of all, it is subject
526 to error jumps in order to avoid merging objects, which makes its optimization
527 difficult and thus requires several learning sessions for an optimal result. On
528 the other hand, avoiding merging causes over-segmentation in some cases, such
529 as the mustard in our dataset that detects leaflets. At this time we cannot say
530 whether this is problematic. It is possible that the leaflet scale is more relevant
531 in terms of segmentation than the leaf for advanced stages of plant development.

532 With the watershed used next, we improve the coarse detection provided by
533 the CNN by extending the detected areas to the ideal soil/vegetation segmenta-
534 tion mask defined by the method developed in a previous study [Vayssade et al.](#)
535 (2021). This step allows the identification of the smallest leaves. Although the
536 smallest elements, with a diameter of 0 – 2 pixels, are still difficult to detect due
537 to spectral mixing ([Louargant et al., 2017](#)). This is an important result as it
538 allows to configure an acquisition system for a specific minimum size of leaves
539 to detect. In this work, the remaining small leaves are already segmented as
540 vegetation and can be considered as stable due to their sizes. Some features
541 are still relevant to be extracted (such as the size of the leaf and its distance
542 from the crop row) and should help in discriminating crop from weeds, other
543 may suffer from the spectral mixing and become irrelevant (spectral signature,
544 texture...).

545 There is still an important type of error. Indeed, a single pixel can be at the
546 origin of errors leading to a bad fusion between two areas. It would therefore
547 be interesting to study structural analysis methods in order to overcome these
548 deficiencies. A possible method would be to vectorize the contours and look
549 for an algorithm to reconnect or split the contours for instance according to
550 convexity singularities.

551 The main interest of the proposed method is its efficiency on mixes of plant
552 species, acquired with natural light. It will be integrated in a processing chain
553 dedicated to the discrimination of crops and weeds in agronomic scenes. Indeed,
554 the detected leaves can be classified according to a large set of criteria (spectral
555 signature, morphological characteristics, texture, distance from the crop row).
556 The underlying hypothesis is that these criteria are more stable at the scale
557 of the leaf than at the scale of the plant. However, this approach has certain
558 limitations when leaves overlap others, as the detected shapes would be hetero-
559 geneous. Multifoliate leaves could also be difficult to characterize. In that case,
560 detecting leaflets instead of leaves may be more relevant.

561 **6. Conclusion**

562 The presented work shows that the CNN network enhances the quality of
563 the segmentation based on multispectral images. Indeed, the background is
564 well removed due to the upstream network with IBF and UFA modules with an
565 accuracy of 95 – 98% of mIoU. Our method is effective in the majority of cases,
566 such as the segmentation of unifoliate leaves like *Arabidopsis-Thaliana* and early
567 developmental stages of plants. However, our method is not effective in the
568 advanced stages of plant development, especially on mustard which has highly
569 segmented leaves. In this case our method detects leaflets instead of leaves.
570 Their identification is nevertheless relevant for the phenotyping or classification

571 of weeds.

572 We have seen that the developed method presents better SBD performances
573 $+ [1.66 - 7.89]\%$ compared to studies that do not use data augmentation. How-
574 ever, on small datasets, it presents lower SBD performances $- [4.32 - 11.92]\%$
575 compared to recent studies that use it. To make the method more consistent
576 we will therefore focus on data augmentation. To improve the detection of
577 classes it would be interesting to improve the upstream modules with atten-
578 tion mechanisms. This method would allow to correct the illumination of the
579 images. As well as the use of a *MIRNET* (Zamir et al., 2020) would allow to
580 eliminate the noise. Finally, to improve the downstream modules, the use of
581 *Selective Kernel Convolution* (Zamir et al., 2020) would allow a better fusion
582 of the multi-scale information instead of *Universal Function Approximator*. We
583 will then try to minimize the under-segmentation detected in our method, by
584 using the *Deep Watershed Transform for Instance Segmentation* method (Bai
585 and Urtasun, 2017).

586 A small performance loss for the developed method is seen on our dataset
587 compared to Komatsuna and LSC dataset. It would be interesting to evaluate
588 the reason(s) leading to this loss as it may come from the uncontrolled acqui-
589 sition conditions, the multispectral nature of the images, the size differences
590 between leaves or the important number of leaves in each images.

591 In all cases, this approach should lead to an enhancement of features ex-
592 traction which may improve crop/weed classification. These increased perfor-
593 mances would lead to a better tracking of the weed flora. These algorithms
594 show promising and robust results in natural acquisition conditions. The seg-
595 mentation results are obtained fast enough to be used in a real-time crop/weed
596 discrimination setting and could be embedded on a Unmanned Ground Vehicle
597 (UGV) for quick and localised intervention. Applied on images acquired from

598 an Unmanned Aerial Vehicle (UAV) this could be used for tool assisted plot
599 management to help farmer in their decision making.

600 **7. Further research**

601 We defined the CNN architecture from the state of the art, adding compo-
602 nents in an increasing development cycle. Thus, we have not established the
603 contribution of each block on this paper. The first 3 modules (iit, ibf, ufa)
604 and the last sprb modules have been widely developed in our previous study
605 about vegetation segmentation, each module have showed a contribution. But
606 further research is undergoing to show the impacts of different modules on the
607 upstream and downstream of the network, and the proposed loss function, and
608 loss weights. This task must be automated trough an hyper-parameter opti-
609 mization process which we did'nt explored wet. And the watershed algorithm
610 used to refine the output must be added to the neural network, and part of
611 this optimization process. Finally more advanced data augmentation could be
612 explored.

613 **Data availability**

614 The Airphen dataset used in this study have been released and it's publicly
615 available at [https://data.inrae.fr/dataset.xhtml?persistentId=doi:10.](https://data.inrae.fr/dataset.xhtml?persistentId=doi:10.15454/JMKP9S)
616 [15454/JMKP9S](https://data.inrae.fr/dataset.xhtml?persistentId=doi:10.15454/JMKP9S), using Creative Common CC0 1.0 Public Domain Dedication li-
617 cence

618 **Funds & Conflicts of Interest**

619 This project has received funding from the European Union's Horizon 2020
620 research and innovation program [grant agreement ID: 727321] (project acronym:
621 IWMPRAISE) and from the French National Research Agency Challenge RoSE

622 [grant agreement ID: ANR-17-ROSE-0002] (project acronym: ROSEAU). The
623 authors declare no conflict of interest.

624 **References**

625 M. Omari, J. Lee, M. . A. Faqeerzada, E. Park, B.-K. Cho, Digital image-based
626 plant phenotyping: a review (2020). doi:[10.7744/kjoas.20200004](https://doi.org/10.7744/kjoas.20200004).

627 M. Louargant, S. Villette, G. Jones, N. Vigneau, J.-N. Paoli, C. Gée, Weed
628 detection by UAV: simulation of the impact of spectral mixing in multi-
629 spectral images, *Precision Agriculture* 18 (2017) 932–951. doi:[10.1007/
630 s11119-017-9528-3](https://doi.org/10.1007/s11119-017-9528-3).

631 C. Gée, E. Denimal, J. Merienne, A. Larmure, Evaluation of weed impact
632 on wheat biomass by combining visible imagery with a plant growth model:
633 towards new non-destructive indicators for weed competition, *Precision Agri-
634 culture* 22 (2021) 550–568. doi:[10.1007/s11119-020-09776-6](https://doi.org/10.1007/s11119-020-09776-6).

635 A. Wang, W. Zhang, X. Wei, A review on weed detection using ground-based
636 machine vision and image processing techniques, *Computers and Electronics
637 in Agriculture* 158 (2019) 226–240. doi:[10.1016/j.compag.2019.02.005](https://doi.org/10.1016/j.compag.2019.02.005).

638 A. M. Hafiz, G. M. Bhat, A survey on instance segmentation: state of the art,
639 *International Journal of Multimedia Information Retrieval* 9 (2020) 171–189.
640 doi:[10.1007/s13735-020-00195-x](https://doi.org/10.1007/s13735-020-00195-x).

641 M. Bonneau, J.-A. Vayssade, W. Troupe, R. Arquet, Outdoor animal tracking
642 combining neural network and time-lapse cameras, *Computers and Electron-
643 ics in Agriculture* 168 (2020) 105150. doi:[10.1016/j.compag.2019.105150](https://doi.org/10.1016/j.compag.2019.105150).

644 H. Scharr, M. Minervini, A. P. French, C. Klukas, D. M. Kramer, X. Liu,
645 I. Luengo, J.-M. Pape, G. Polder, D. Vukadinovic, et al., Leaf segmentation

646 in plant phenotyping: a collation study, *Machine vision and applications* 27
647 (2016) 585–606. doi:[10.1007/s00138-015-0737-3](https://doi.org/10.1007/s00138-015-0737-3).

648 H. Chen, X. Qi, L. Yu, P.-A. Heng, Dcan: Deep contour-aware networks for
649 accurate gland segmentation, 2016. [arXiv:1604.02677](https://arxiv.org/abs/1604.02677).

650 D. D. Morris, A pyramid cnn for dense-leaves segmentation, 2018.
651 [arXiv:1804.01646](https://arxiv.org/abs/1804.01646).

652 Y. Cui, G. Zhang, Z. Liu, Z. Xiong, J. Hu, A deep learning algorithm for one-
653 step contour aware nuclei segmentation of histopathology images, *Medical and*
654 *Biological Engineering and Computing* 57 (2019) 2027–2043. doi:[10.1007/
655 s11517-019-02008-8](https://doi.org/10.1007/s11517-019-02008-8).

656 J. Bell, H. M. Dee, Leaf segmentation through the classification of edges, 2019.
657 [arXiv:1904.03124](https://arxiv.org/abs/1904.03124).

658 X. Xie, J. Chen, Y. Li, L. Shen, K. Ma, Y. Zheng, Instance-aware self-supervised
659 learning for nuclei segmentation, 2020. [arXiv:2007.11186](https://arxiv.org/abs/2007.11186).

660 J.-A. Vayssade, J.-N. Paoli, C. Gée, G. JONES, DeepIndices: Remote
661 Sensing Indices Based on Approximation of Functions through Deep-
662 Learning, Application to Uncalibrated Vegetation Images, *Remote Sensing* 13
663 (2021) 1–21. URL: [https://hal-agrosup-dijon.archives-ouvertes.fr/
664 hal-03263161](https://hal-agrosup-dijon.archives-ouvertes.fr/hal-03263161). doi:[10.3390/rs13122261](https://doi.org/10.3390/rs13122261).

665 G. Avrin, D. Boffety, S. Lardy-Fontan, R. Régnier, R. Rescoussié, V. Barbosa,
666 Design and validation of testing facilities for weeding robots as part of rose
667 challenge, in: 1st International Workshop on Evaluating Progress in Artificial
668 Intelligence (EPAI), 2020. URL: [https://hal.archives-ouvertes.fr/
669 hal-03010299](https://hal.archives-ouvertes.fr/hal-03010299).

- 670 J.-A. Vayssade, G. JONES, J.-N. Paoli, C. Gée, Two-step multi-spectral reg-
671 istration via key-point detector and gradient similarity. Application to agro-
672 nomic scenes for proxy-sensing, in: Proceedings of the 15th International Joint
673 Conference on Computer Vision, Imaging and Computer Graphics Theory and
674 Applications, La Valette, Malta, 2020. URL: [https://hal-agrosup-dijon.
675 archives-ouvertes.fr/hal-02499730](https://hal-agrosup-dijon.archives-ouvertes.fr/hal-02499730). doi:10.5220/0009169301030110.
- 676 A. Dutta, A. Zisserman, The VIA annotation software for images, audio and
677 video, in: Proceedings of the 27th ACM International Conference on Mul-
678 timedia, MM '19, ACM, New York, NY, USA, 2019. doi:10.1145/3343031.
679 3350535.
- 680 H. Scharr, T. Pridmore, S. Tsafaris, Computer vision problems in plant pheno-
681 typing, cvppp 2017: Introduction to the cvppp 2017 workshop papers, 2017,
682 pp. 2020–2021. doi:10.1109/ICCVW.2017.236.
- 683 H. Uchiyama, S. Sakurai, M. Mishima, D. Arita, T. Okayasu, A. Shimada, R.-i.
684 Taniguchi, An easy-to-setup 3d phenotyping platform for komatsuna dataset,
685 2017, pp. 2038–2045. doi:10.1109/ICCVW.2017.239.
- 686 F. Perez-Sanz, P. J. Navarro, M. Egea-Cortines, Plant phenomics: an overview
687 of image acquisition technologies and image data analysis algorithms, Giga-
688 Science 6 (2017). doi:10.1093/gigascience/gix092.
- 689 P. Lottes, C. Stachniss, Semi-supervised online visual crop and weed classifica-
690 tion in precision farming exploiting plant arrangement, in: 2017 IEEE/RSJ
691 International Conference on Intelligent Robots and Systems (IROS), 2017,
692 pp. 5155–5161. doi:10.1109/IROS.2017.8206403.
- 693 B. Lin, Y. Sun, J. Sanchez, Efficient vessel feature detection for endoscopic
694 image analysis, IEEE transactions on bio-medical engineering 62 (2014).
695 doi:10.1109/TBME.2014.2373273.

696 S. Moradi, M. Ghelich-Oghli, A. Alizadehasl, I. Shiri, N. Oveisi, M. Oveisi,
697 M. Maleki, J. Dhooge, A novel deep learning based approach for left ventricle
698 segmentation in echocardiography: Mfp-unet, 2019. [arXiv:1906.10486](https://arxiv.org/abs/1906.10486).

699 C. Nwankpa, W. Ijomah, A. Gachagan, S. Marshall, Activation functions:
700 Comparison of trends in practice and research for deep learning, 2018.
701 [arXiv:1811.03378](https://arxiv.org/abs/1811.03378).

702 R. Liu, J. Lehman, P. Molino, F. P. Such, E. Frank, A. Sergeev, J. Yosinski, An
703 intriguing failing of convolutional neural networks and the coordconv solution,
704 CoRR abs/1807.03247 (2018). [arXiv:1807.03247](https://arxiv.org/abs/1807.03247).

705 H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network, 2017.
706 [arXiv:1612.01105](https://arxiv.org/abs/1612.01105).

707 M. Rahman, Y. Wang, Optimizing intersection-over-union in deep neu-
708 ral networks for image segmentation, volume 10072, 2016, pp. 234–244.
709 doi:[10.1007/978-3-319-50835-1_22](https://doi.org/10.1007/978-3-319-50835-1_22).

710 D. Zhou, J. Fang, X. Song, C. Guan, J. Yin, Y. Dai, R. Yang, Iou loss for 2d/3d
711 object detection, 2019. [arXiv:1908.03851](https://arxiv.org/abs/1908.03851).

712 H.-J. Bae, C.-W. Kim, N. Kim, B. Park, N. Kim, J. B. Seo, S. M. Lee,
713 A perlin noise-based augmentation strategy for deep learning with small
714 data samples of hrct images, Scientific Reports 8 (2018). doi:[10.1038/
715 s41598-018-36047-2](https://doi.org/10.1038/s41598-018-36047-2).

716 V. Kulikov, V. Yurchenko, V. Lempitsky, Instance segmentation by deep color-
717 ing, 2018. [arXiv:1807.10007](https://arxiv.org/abs/1807.10007).

718 D. Ward, P. Moghadam, Scalable learning for bridging the species gap in image-
719 based plant phenotyping, Computer Vision and Image Understanding 197-198
720 (2020) 103009. doi:[10.1016/j.cviu.2020.103009](https://doi.org/10.1016/j.cviu.2020.103009).

- 721 D. P. S. Gomes, L. Zheng, Leaf segmentation and counting with deep
722 learning: on model certainty, test-time augmentation, trade-offs, 2020.
723 [arXiv:2012.11486](https://arxiv.org/abs/2012.11486).
- 724 J.-M. Pape, C. Klukas, 3-d histogram-based segmentation and leaf detection for
725 rosette plants, in: L. Agapito, M. M. Bronstein, C. Rother (Eds.), Computer
726 Vision - ECCV 2014 Workshops, Springer International Publishing, Cham,
727 2015, pp. 61–74. doi:[10.1007/978-3-319-16220-1_5](https://doi.org/10.1007/978-3-319-16220-1_5).
- 728 S. W. Zamir, A. Arora, S. H. Khan, M. Hayat, F. S. Khan, M. Yang, L. Shao,
729 Learning enriched features for real image restoration and enhancement, vol-
730 ume abs/2003.06792, 2020. [arXiv:2003.06792](https://arxiv.org/abs/2003.06792).
- 731 M. Bai, R. Urtasun, Deep watershed transform for instance segmentation, in:
732 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),
733 2017, pp. 2858–2866. doi:[10.1109/CVPR.2017.305](https://doi.org/10.1109/CVPR.2017.305).