sensors

MDPI

Article

Vision Based Modeling of Plants Phenotyping in Vertical Farming under Artificial Lighting

Benjamin Franchetti [1,*], Valsamis Ntouskos [2], Pierluigi Giuliani [1], Tiara Herman [1], Luke Barnes [1] and Fiora Pirri [2]

[1] Agricola Moderna, Viale Col di Lana 8, 20136 Milan, Italy; pierluigi@agricolamoderna.com (P.G.); tiara@agricolamoderna.com (T.H.); luke@agricolamoderna.com (L.B.)

[2] Alcor Lab, DIAG, Sapienza University of Rome, Via Ariosto 25, 00185 Rome, Italy; ntouskos@diag.uniroma1.it (V.N.); pirri@diag.uniroma1.it (F.P.)

* Correspondence: benjamin@agricolamoderna.com

Received: 2 September 2019; Accepted: 8 October 2019; Published: 10 October 2019

Abstract: In this paper, we present a novel method for vision based plants phenotyping in indoor vertical farming under artificial lighting. The method combines 3D plants modeling and deep segmentation of the higher leaves, during a period of 25–30 days, related to their growth. The novelty of our approach is in providing 3D reconstruction, leaf segmentation, geometric surface modeling, and deep network estimation for weight prediction to effectively measure plant growth, under three relevant phenotype features: height, weight and leaf area. Together with the vision based measurements, to verify the soundness of our proposed method, we also harvested the plants at specific time periods to take manual measurements, collecting a great amount of data. In particular, we manually collected 2592 data points related to the plant phenotype and 1728 images of the plants. This allowed us to show with a good number of experiments that the vision based methods ensure a quite accurate prediction of the considered features, providing a way to predict plant behavior, under specific conditions, without any need to resort to human measurements.

Keywords: vision based phenotyping; plants growth prediction; vertical farming; LED

1. Introduction

Eating habits have created a demand for food that is available 365 days a year, regardless of season and location of production. Food travels on average 1500 km, and wastes valuable energy to reach the end consumer. Thirty percent of produce is wasted before it arrives to the end consumer, contributing to 8% of global CO_2 emissions. The food that does survive the long journey is not fresh and lacks vital nutrients. Increasingly, consumers are demanding local food, rich in nutritional values and without chemical pesticides and that has caused minimal damage to the environment. According to FAO (https://science.sciencemag.org/content/292/5519/1071), 30% of global energy consumption and 70% of freshwater is used for agriculture [1]. In addition, the increase of the global population suggest that agricultural land availability per capita will decrease in the coming years [2].

The increasing global population, coupled with environmental concerns is leading to the development of new forms of agriculture that consume fewer resources and are not dependent on arable land. Vertical farms with artificial lighting (VFAL) grow crops in vertically stacked layers under carefully controlled conditions, using light-emitting diodes (LEDs) that mimic sunlight and out of soil techniques (e.g., hydroponics and aeroponics) where water is recycled. The ability to precisely tailor the growing environmental also leads to increased yields and crop quality in comparison with field and greenhouse production [3].

VFAL and AI Based Methods

As VFAL does not rely on external environmental conditions and is protected from changes in weather or soil fertility, it enables the production of crops year round with minimal demand for water, nutrients and land [4]. Each farming unit is its own individual ecosystem, creating the exact environment for plants to flourish. By developing the optimal light spectrum, temperature, humidity, pH, and nutrients, researchers can ensure the best possible flavor, color and nutritional quality for each plant.

Nonetheless, the large energy associated with especially lighting but also cooling and dehumidification is limiting the diffusion of VFALs [5]. Consequently in VFAL optimizing resource efficiency, while maximizing yield and quality is critical.

Under these premises, plant growth optimization and quality assessment while minimizing the required resources asks for AI methods to ensure a highly flexible technology for a good rate of development. By capturing many data and images relating to the plant physiology, growth and environmental conditions, it is possible to estimate information of the plant growth behavior.

Accuracy of data collected and predicted by methods based on vision and advanced sensory tools, with respect to physiological measurement, makes it possible to create predictive models of plant productivity based solely on automated data collection. Additionally, discovering relationships between image sensor data and more in-depth parameters of crops such as flavor, nutrient deficiency and disease presence, high-throughput AI based input testing could be developed. Accelerating the screening of inputs to the VFAL system, as well as predictive tools for optimizing environmental inputs are important for enabling VFAL to significantly contribute to reductions in the environmental impact of future food production.

2. Related Works

Since commercially operative VFALs are a relatively recent phenomenon, predictive tools for crop quality and yield are not currently available. Conventional field and outdoor agriculture by comparison has benefited from long-standing multidisciplinary and legislative efforts to provide methods for improving resource use efficiency and productivity [6].

Precision agriculture has arisen as a field that uses data science, computer vision (CV) and machine learning (ML) to increase agricultural productivity while minimizing its environmental impact [7]. More recently, precision agriculture techniques have provided tools for improved crop management through detailed monitoring of environmental conditions, maps of spatial variability in soil conditions and both remote and proximal sensing of reflected and radiated energy from crop surfaces [8,9]. Liakos et al. [7] provided an excellent review paper on the ML techniques employed in field agriculture. Specifically, Liakos et al. [7] reviewed studies which used ML methods to model the yield prediction, disease detection, weed detection and crop quality in field agriculture. Yield prediction is critical in precision agriculture as it enables to increase crop management and productivity [7]. For example, Ramos et al. [10] developed non-destructive method that automatically counted coffee fruits and estimated their weight on a branch.

Vision-based analysis of crops are an essential component of remote sensing in agriculture. For example, the advent of hyperspectral cameras, which allow images to be divided into many spectral bands, have led to the development of vegetation indices which can be used to estimate growth parameters including biomass, chlorophyll content and leaf area index [8]. Recently, hyperspectral cameras have been developed which can be attached to unmanned aerial vehicles and used to capture spatial resolutions of up to 1 cm. Consequently, much attention has been given to the development of methods for extracting important information on crop development using hyperspectral imaging. For example, Sylvain et al. [11] showed that vegetation indices extraction from ultra-high resolution images can be used to estimate green fraction, green area index, leaf/canopy chlorophyll content and nitrogen contents of sugar beet crops.

Monitoring of greenhouse crops using remote sensing-type applications also show potential for improving knowledge of crop requirements. In contrast to field-based farming, greenhouses allow greater control of the growing environment. Nevertheless, the practical application of remote sensing for yield prediction in many crops is limited.

With the advent of new high-quality cameras and the recent advances in CV due to deep learning methods, RGB and RGBD images have played a relevant role too. Ballesteros et al. [12] demonstrated that onion yield predictions can be made using RGB images which are processed to determine plant height and volume using a modified version of Volume and Leaf Area Index Calculation (VOL-LAIC) software.

However, monitoring often focuses entirely on the macro-environmental parameters, which is only partially related to the microclimate surrounding the leaves. Story and Kacira [13] developed a crop monitoring system which captures and processes RGB, NIR and thermal image data using a machine vision system to determine overall plant growth and health, as well as parameters for plant phenotyping applications.

Nevertheless, CV applications in greenhouses are made difficult by several sources of variation, e.g., variations in solar intensity, an issue that is likely to be compounded in facilities using variable-intensity supplemental lighting. Since VFALs provide a greater degree of environmental stability than either field or greenhouse cultivation, the development of intelligent crop monitoring systems will enable very high-resolution crop management.

3. Contribution and Overview

3.1. Contribution

The aim of this study was to show how, using AI and specifically vision based methods, it is possible to obtain information on the plant phenotype in a closed VFAL environment. A major difficulty in plant phenotyping is to document day by day the plants response to several factors even in controlled environments such as vertical farms.

Consequently, there is still a lack of experimental studies evaluating the automatic measurements made possible by novel techniques with ground truth taken with manual measurements. Manual measurements require a punctual and meticulous and continuous effort to process a great amount of data along the growing cycle of a plant.

This work provides a joint effort between the agronomy and computer science fields to evaluate the effectiveness of off the shelf research methods, suitably adapted to the plants conditions, for some relevant aspects of phenotyping. Relevant data were collected digitally, visually and manually to both facilitate the development of the training set and provide outputs from the model, which are crucial for fast development of VFAL technologies.

The authors performed various experiments on the growth of Basil, where a large dataset was collected both manually and visually for the first time. Specifically, through three cycles of experiments, 2592 data points were manually collected on: (1) plant height; (2) leaf weight, for the whole set of leaves of each plug; (3) number of leaves per plant; and (4) weight of each plug. In parallel, 1728 images were collected. To the best of the authors knowledge, this is the first time many data were gathered to compare predictions and observations, and the models validated.

The main outcome of the research is the identification of relevant phenotype features with images and precise computation of three important growth measures, namely plants heights, leaf area and leaf weight, at different stages of growth (see Figure 1). Precise here means that the accuracy of the prediction could be assessed by comparing the results with manual measurements. These results are reported in Section 5.

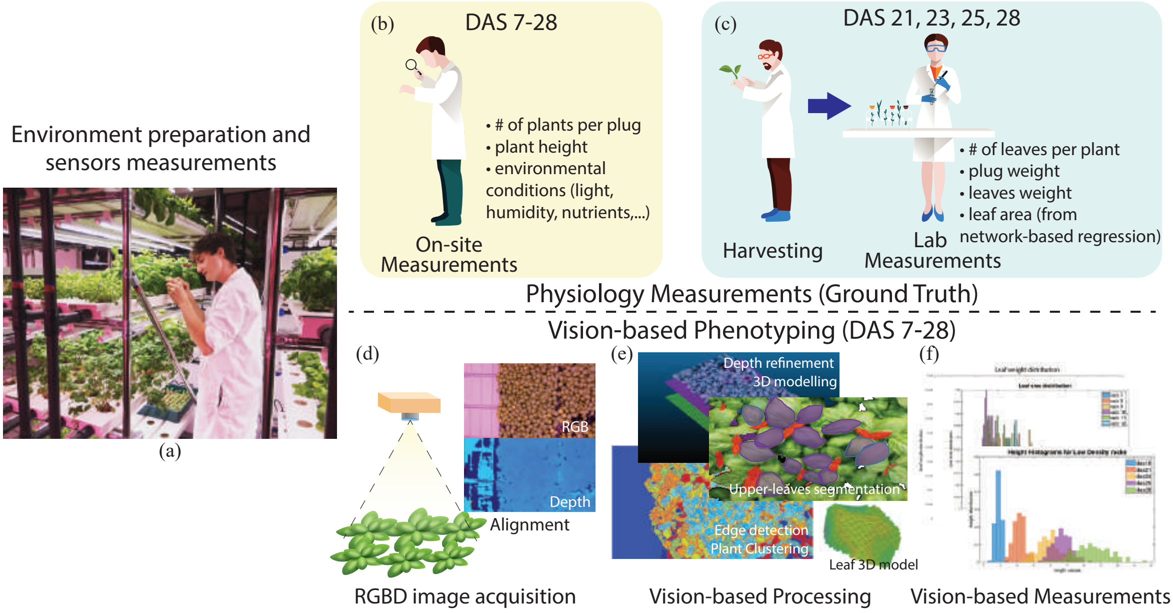


Figure 1. The figure shows the schema of the two parallel computational processes leading to automatic measurements: the experiment setting (**a**); the manual data collection via on-site (**b**) and lab measurements (**c**); the images acquisition and alignment (**d**); and the ML and CV methods (**e**) to predict phenotype features, such as plants height, leaf area, and leaf weight (**f**).

3.2. Paper Outline

In the following sections, we describe each of the above contributions providing results against collected ground truth data in Section 5.

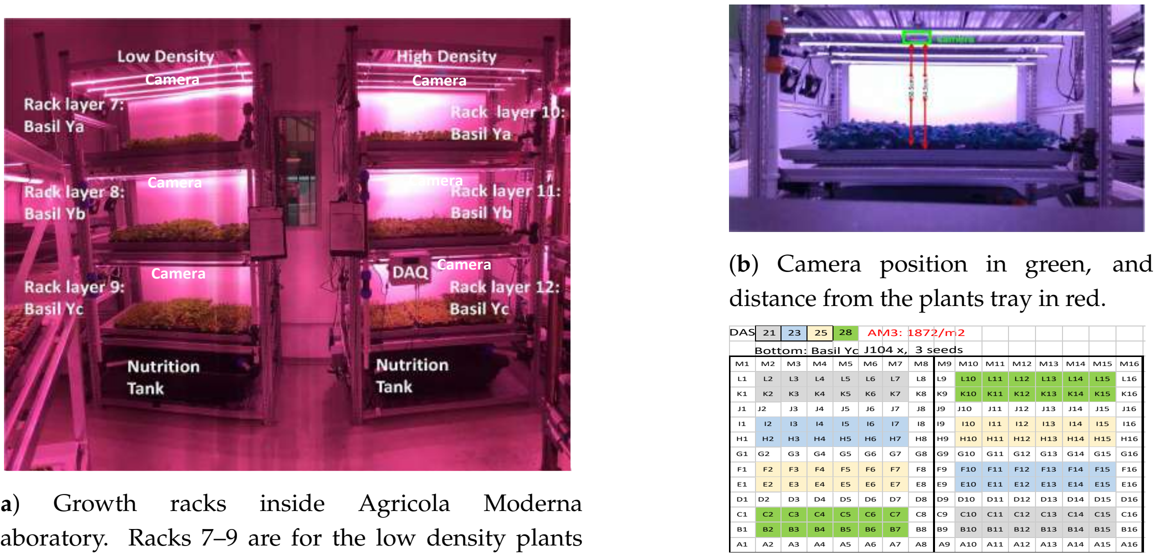
In Section 4, we introduce the vision and learning methods leading to the above results. First, in Section 4.1, we introduce the outcome of image acquisition, the noise removal and 3D modeling. Further, in Section 4.2, we explain the approach taken to measure plants height at specific grow days, Days After Seedling (DAS), namely 7, 18, 21, 24, 28, based on three-dimensional depth maps and possible alternative methods based on point clouds. After introducing the computation of height, in Section 4.3, we introduce the processing required to measure the leaf area, restricted only to the surface (i.e., visible) leaves, without losing generality since it turns out to be exactly the same process approach used when taking manual measurements. In addition, for leaf area, we provide accuracy and derive the leaf weights according to the density computed by the experimental studies and described in Section 5.

Finally, in Section 5, we illustrate the results and evaluate accuracy against the ground truth observed measurements taken by the agronomists.

4. Vision Process Flow

As described in more detail in Section 5, each experiment was carried out for 28 DAS, for the six layers shown in Figure 2.

Our task was to predict the height of the plant from DAS 7 to DAS 28 and to predict both the leaf size of the visible layer of leaves for the last DAS (DAS 21 to 28) and leaf weight.



(**a**) Growth racks inside Agricola Moderna laboratory. Racks 7–9 are for the low density plants distribution, while Racks 10–12 are for high density plants distribution.

(**b**) Camera position in green, and distance from the plants tray in red.

DAS	21	23	25	28	AM3: 1872/m2										
	Bottom: Basil Yc				J104 x,	3 seeds									
M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15	M16
L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14	L15	L16
K1	K2	K3	K4	K5	K6	K7	K8	K9	K10	K11	K12	K13	K14	K15	K16
J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	J15	J16
I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11	I12	I13	I14	I15	I16
H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16
G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16
E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14	E15	E16
D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16
C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16
B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15	B16
A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16

(**c**) An example of the manual scheme for ground truth data collection.

Figure 2. Experiments setting within the Agricola Moderna laboratory. (**c**) Each cell indicates a basil plant, for a high density tray collecting plugs of the Yc specie.

4.1. Acquisition and Noise Removal

To collect images for the entire DAS cycle, we positioned commercial close range RGBD sensors [14], one for each rack layer, at about 54.3 cm from the rack plane and at 50.5 cm from tray plane, as illustrated in Figure 2. The cameras were fixed at 50.5 cm from the plant substrate and 54.5 cm from the bottom of the trays, because this distance represents the typical distance between the LEDs and plant base in a VFAL. Each high density tray contains 144 plugs and each low density tray contains 104 plugs. Since each frame images two trays, each image collects either 144 × 2 plugs for the high density layers or 104 × 2 plugs for the low density layer.

An RGBD image realizes two representations: a color image with three color channels (R, G and B), returning a tensor of size $H \times W \times 3$ and a depth map with only the depth values, namely each pixel at the x and y coordinates of the image specifies the depth value d, given in mm, indicating how far away the imaged object is.

The RGBD sensor has intrinsic parameters denoted by a matrix K, specifying the sensor focal length f, the scale values m_x and m_y specifying the number of pixels per mm., a skew parameter s and the center of projection (p_x, p_y), hence:

$$K = \begin{pmatrix} fm_x & s & p_x m_x \\ 0 & fm_y & p_y m_y \\ 0 & 0 & 1 \end{pmatrix} \tag{1}$$

The intrinsic matrix allows projecting a point in space measured in mm into a point on the image plane, measured in pixels, and to deproject a pixel on the image plane to a point in space. Together with the intrinsic K distortion parameters are made available for radial corrections $(\kappa_1, \kappa_2, \ldots)$ such that the corrected coordinates $(\hat{x}, \hat{y})$ of a pixel are obtained as: $\hat{x} = x_c + L(r)(x - x_c)$ and $\hat{y} = y_c + L(r)(y - y_c)$, where (x_c, y_c) is the distortion center, (x, y) are the distorted coordinates, and $L(r) = 1 + \kappa_1 r + \kappa_2 r + \ldots$, with $r^2 = (x - x_c)^2 + (y - y_c)^2$ (see [15]).

Typically, RGBD sensors require a procedure to align the RGB image to the depth image, which is also useful to obtain a point cloud from the depth map with points colored as in the RGB image.

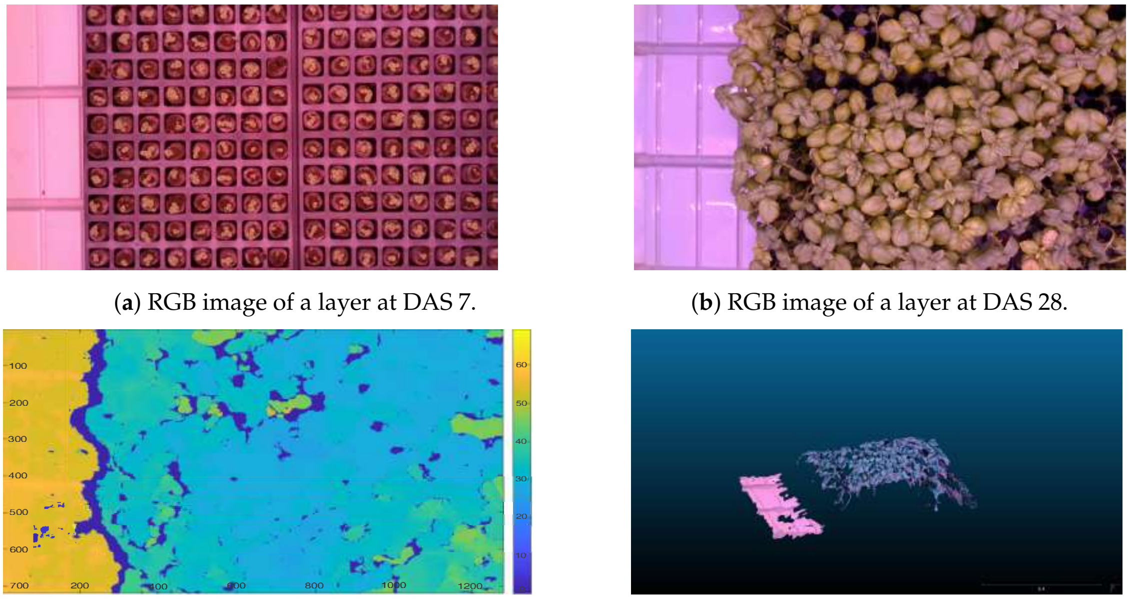
To map a depth image to a point cloud the usual formula, with d the depth, $\mathbf{R}$ a rotation matrix and $\mathbf{t}$ the translation vector, is:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = dK^{-1}\mathbf{R} \begin{pmatrix} \hat{x} \\ \hat{y} \\ 1 \end{pmatrix} + \mathbf{t} \tag{2}$$

X, Y and Z are the coordinates of the scene with the camera as the global reference point, while $\hat{x}$, $\hat{y}$ and d are the corrected coordinates of the depth image.

Because the camera is fixed, a single image is not quite dense, and there are points of reflection where the sensor does not return data. This can be observed in Figure 3, where RGBD results without specific processing are shown. In particular, we can note in Figure 3c the holes in the depth map, and in Figure 3d that the produced point cloud is noisy. To address these problems and obtain an almost dense representation, we merged multiple consecutive depth maps and smoothed the resulting image with a Gaussian kernel. This merging makes sense, even though the camera is fixed, since natural flickering of the plants illumination and the ventilation, moving the leafs and their illumination, changes the image of the leaves, thus have the effect of small camera motions (see Figure 4a).

In fact, because the point cloud results from the depth images, all taken from the same vantage point, as the camera is fixed (see Figure 2), noise elimination is not trivial and may result in removal of useful points. To reduce the noise, we first fitted a plane to the point cloud, parallel to the tray surface, thus eliminating all points beyond the plane (see Figure 4b,c). Further, we used the well known Iterative Closest point (ICP) algorithm [16,17], exploiting two point clouds from merged depth maps to clean points with loose correspondences to the common fitted plane.



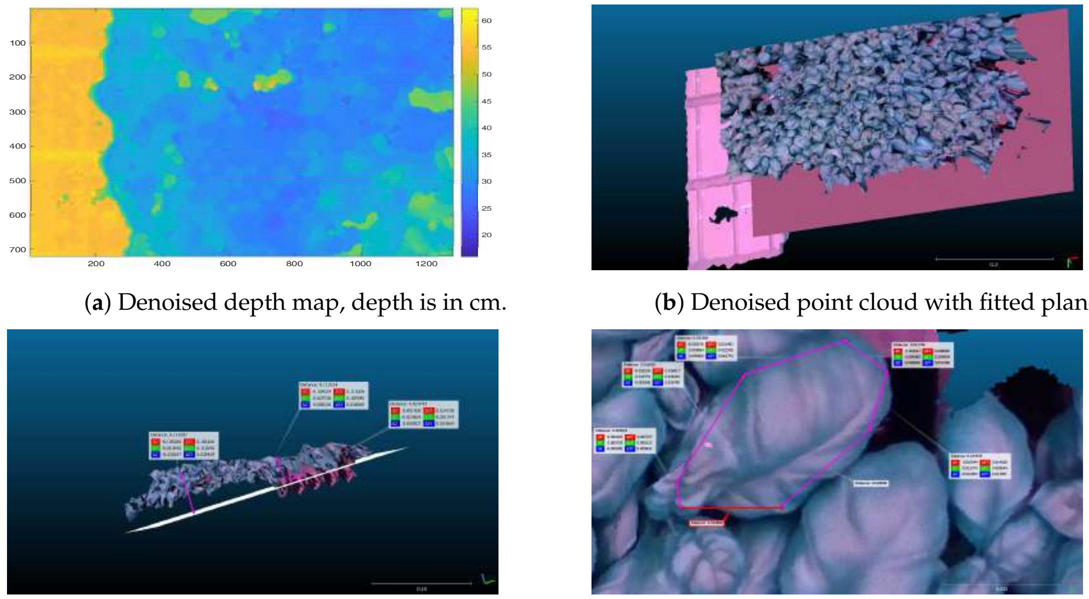
(**a**) RGB image of a layer at DAS 7.

(**b**) RGB image of a layer at DAS 28.

(**c**) Raw depth map from the RGBD sensor.

(**d**) Single point cloud, from raw depth map.

Figure 3. The data collected include 1728 color images and 1728 depth maps of the plants (better seen in color).



(**a**) Denoised depth map, depth is in cm.

(**b**) Denoised point cloud with fitted plane.

(**c**) Rescaled point cloud.

(**d**) Leaf contour measures taken on rescaled point cloud.

Figure 4. The figure shows relevant preprocessing steps for both depth map and point cloud.

Finally, the point cloud was rescaled using a known metric of the imaged objects, namely the dimension of the cell where plugs are put in, and the tray dimension. This step is required for two reasons: first because the camera, despite being fixed, is not well aligned with the tray plane, and secondly because the intrinsic parameters mapping pixels to mm are not sufficiently accurate. To rescale, we considered four points, which are the four vertices of a visible cell on the tray, namely the cell where plugs are located, which we know are separated by 30 mm, and computed the transformation $\mathcal{T}$ mapping the current points to the expected ones. The transformation $\mathcal{T}$ is an affine transformation including rotation, translation and scaling, such that: $\mathbf{V}_E = \mathcal{T}\mathbf{V}_M$, where $\mathbf{V}_E$ is the expected matrix formed by the four points in real world dimension, and $\mathbf{V}_M$ is the matrix of points as measured on the point cloud. According to the transformation, effective measures can be taken on the point cloud also at close distance, as illustrated in Figure 4d.

We actually used the depth map for the height prediction (next section) and the point cloud for the leaf area and weight prediction.

4.2. Height Prediction

Given the basic setting described above, we next elaborate about automatic plant height computation at each defined DAS step.

A relevant insight given by the agronomists in height computation is the difference amid plants at growth stages, namely DAS, according to plant density (see Section 5.1). This information can be hardly assessed by the average growth for each tray and the heights standard deviation. In fact, it is interesting to note, for example, that the standard deviation significantly increases at later growth steps and according to density, namely the most dense trays have lower standard deviation because both plants competitiveness and lack of space stabilize plants growth at similar heights. Note that, because there are more than one plants in each plug, the agronomists measure the highest plant. However, it seems necessary to predict the height surface according to the region where each plant grows, namely near the cell on the tray where its plug is located.

To obtain height measures for each tray and each DAS, we considered the depth maps obtained day by day for each rack involved in the experiments. However, the plant surface provided both by the point cloud and the depth map hardly distinguishes individual plants, especially at later stages, at which only a carpet of leaves is visible.

To obtain quite precise measurements for each cell, where plugs are located, we need to recover the cells, which at later stage are clearly not visible.

To this end, we fitted lines to the tray images taken at the early DAS. Lines were computed first computing a Canny edge detector of images of early stages, where the tray cells are still visible. The Canny edge detection is shown in Figure 5, where the image is obviously rectified and undistorted. Then, from the binary image of the edges, the lines were obtained with the Hough transform, and a new binary matrix L, with zeroes at the line locations, was obtained. Then, letting M be the depth map and L be the image of the line fitted, the new depth map with the line fitted is $M_l = M \circ L$, with $\circ$ the pointwise Hadamard product.

The new depth map M_l returns the distance between the camera and the plants, except for the lines bounding the cells, where plugs are located (see Figure 5), where as mentioned above the value is zero.

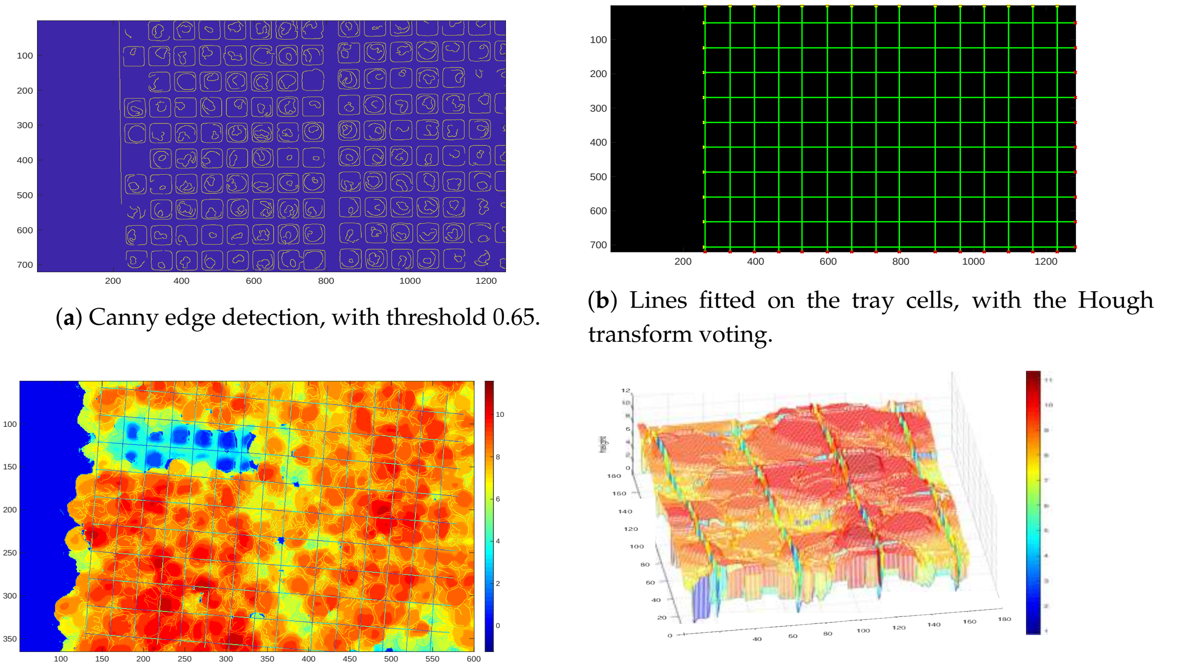
As noted above, the x and y coordinates are given in pixel. In other words, the depth map is a surface covering an area of 720×1280 pixels, while the tray area measures 110 cm × 63 cm. However, since the depth d is measured in millimeters, computing the height rescaling is not required.

Finally, the plants height surface was obtained by subtracting from the depth map the distance to the plane parallel to the tray surface, which is known to be 505 mm, as noted above.

Let $D(\pi, \tau)$ be the distance between the camera image plane π and the surface plane τ of the tray, measured along the ray normal to the camera image plane, and let M_l be the depth map, with lines bounding the cells. The depth map was smoothed by convolving it with a Gaussian kernel of size 11×11 with $\sigma = 1.3$. Then, the plant height surface H_p was computed as:

$$H_p = D(\pi, \tau) - M_l + \epsilon \tag{3}$$

with ϵ Gaussian noise, possibly not reduced by smoothing.



(**a**) Canny edge detection, with threshold 0.65.

(**b**) Lines fitted on the tray cells, with the Hough transform voting.

(**c**) Height surface at DAS 24, in mm.

(**d**) 3D reconstructed leaves with the lines separating the tray cells visible, with height in mm.

Figure 5. The figure shows the main steps to obtain the surface height of a tray partitioned according to the tray cells. (**c**) A section that has been cut out for measurements is visible.

To compute the height of plants relative to a plug, we only need to consider all the points within the cell bound by the appropriate vertices. Indeed, we can just consider the projection on the plane of the surface points, as far as the z coordinate (note that we name it z because it is the plant height, not the depth d) is not zero. First, note that V is a vertex on the tray lines if all its four-connected points are equal to 0. Therefore, let $\mathcal{V} = \{V_j | V_j$ be a vertex of the approximate rectangle bounding a tray cell, $j = 1, \ldots, n\}$. Let $\hat{p} = (x, y)$ be the projection on the plane of a point $p = (x, y, z)^\top$ on the surface height, $z > 0$. Then, $\hat{p}$ is within a cell q bounded by the vertices $\langle V_{i,q}, V_{j,q}, V_{k,q} \rangle \in \mathcal{V}$ iff the vertices are connected and:

$$\begin{aligned} \mathcal{L}(p,q) \quad = \quad & \Big(((V_{i,q} - V_{j,q})^\top (V_{i,q} - V_{j,q})) > (V_{i,q} - V_{j,q})^\top (V_{i,q} - \hat{p}) > 0 \wedge ((V_{j,q} - V_{k,q})^\top (V_{j,q} - V_{k,q})) > \\ & ((V_{j,q} - V_{k,q})^\top (V_{j,q} - \hat{p})) > 0 \Big) \equiv \top \end{aligned} \tag{4}$$

Then, the set of points on the height surface of a cell q is $H_q = \{p = (x, y, z)^\top | \mathcal{L}(\hat{p}, q) \equiv \top\}$. This defined, it is trivial to determine the highest and average height for each cell q, and the height variance within a cell:

$$\begin{aligned} h_{max}(q) &= \max\nolimits_z \{(x, y, z) \in H_q\}, \\ h_\mu(q) &= \tfrac{1}{N} \textstyle\sum_{(x,y,z)_j \in H_q}^{N} z_j \\ \sigma_q &= \tfrac{1}{N-1} \textstyle\sum_{(x,y,z)_j}^{N} (z_j - h_\mu)^2 \end{aligned} \tag{5}$$

Here, $N = |H_q|$, with $|\cdot|$ the set cardinality.

The results and comparison of the predicted height with the measured ones, according to the modeled surface, are given in Section 5.

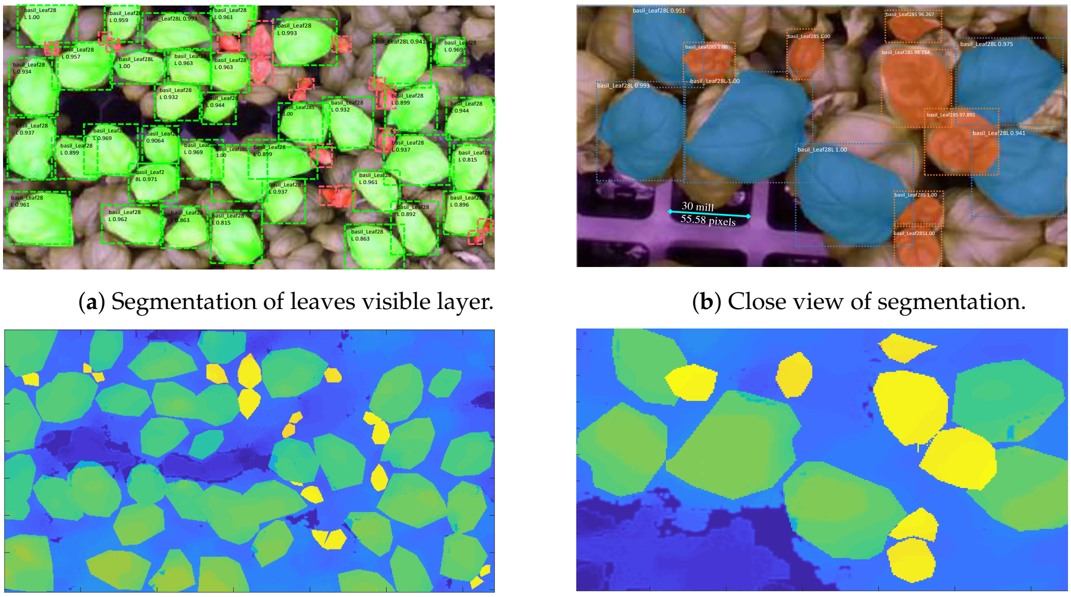
4.3. Leaf Area and Weight

In this section, we discuss the computation of both leaf area and leaf weight. We recall that leaf area was not measured by the agronomists, however they collected the weight of the leaves of the plants in a plug, and the number of leaves. Furthermore, they made available, from previous studies, a sample of leaves for the considered species with area and weight. From this sample, they measured $\rho_{BL} = \dfrac{W_{BL}}{Area_{BL}}$, where W_{BL} is the weight of a basil leaf, $Area_{BL}$ is the area of the leaf and ρ_{BL} is the density. From this sample, the weight, and the knowledge of the number of leaves, we derived a reference ground truth for leaf area to be directly comparable with our predicted leaf area.

From the predicted leaf area, using the density, we computed the leaf weight. However, we could not predict thus far the number of leaves in a plant or the number of plants in a plug.

For predicting the leaf area, we proceeded as following. We computed instance segmentation of superficial leaves at DAS 18, 21, 24 and 28. Using the colored mask and Equation (2), we projected the mask to the point cloud to obtaini a surface well approximating in dimension and shape the considered leaf. Because each point of the projected mask is colored, the mask can also be segmented from the point cloud, simply selecting all points with the specific color. Note that the point cloud can be saved as a matrix of dimension $N \times 6$ where N is the number of points, the first three column are the point locations and the last three the associated color.

Then, we fitted a mesh surface to the chosen leaf, eliminating all the redundant meshes. The process is shown in Figures 6 and 7.



(**a**) Segmentation of leaves visible layer.

(**b**) Close view of segmentation.

(**c**) Projection on the depth map.

(**d**) Close view of projection.

Figure 6. Segmentation of large and new leaves and projection of the segmentation on the depth map.

As noted in Section 3, we can measure leaf area only for the surface leaves not occluded by more than 60% by other leaves. To obtain this partial yet significant result, we resorted to leaf segmentation with Mask R-CNN [18], a well known deep network for instance segmentation. We used our implementation in Tensorflow, which is freely available on GitHub (See https://github.com/alcor-lab).

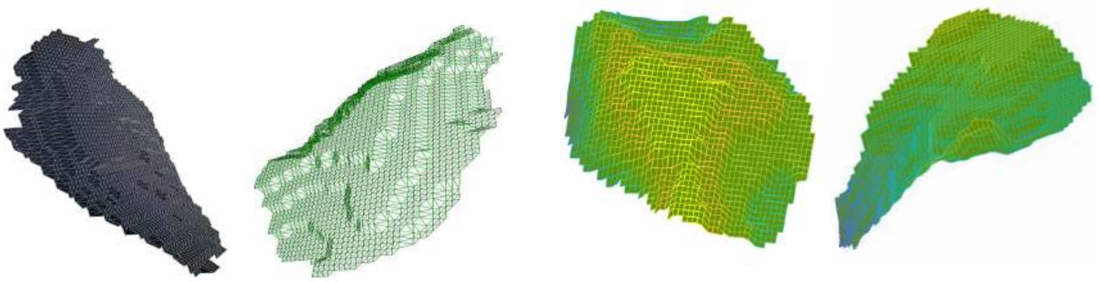
As is well known, CNNs typically require a huge collection of data for training, and in our case well known datasets such as COCO (https://github.com/cocodataset/cocoapi) could not be used. In addition, we noted that leaves completely change aspect after the early DAS, namely around DAS 17, when they gain a characteristic grain.

Therefore, we decided to collect both small and large leaves of almost mature plants, starting from DAS 17.

We collected only 270 labeled large leaves and 180 small leaves, with LabelMe (See http://labelme2.csail.mit.edu/Release3.0/browserTools/php/matlab_toolbox.php) and following Barriuso and Torralba [19], selecting images from Rack 7, the first experiment and DAS 17, 18, 19, 21, 23, 24, and 28. We considered only Rack 7, to make testing on the remaining racks clearer. However, in the end, we also used Rack 7 for testing, although we delivered separate tests to not corrupt results.

We tested the network on two resolutions, as shown in Figure 6, obtaining very good results for both training and testing. The accuracy of each large leaf recognition is given within each bounding box; note that small leaf accuracy is not shown.

It is apparent that, because the number of samples is so small, leaves occluded by more than 60% of the area are not recognized. However, we noticed that the samples we could collect from the rack surface are good representatives of the whole leaves set. Furthermore, as explained in the sequel, this does not affect significantly the measured area.



(**a**) Leaf surface by triangulation.

(**b**) Leaf surfaces obtained by fitting a quadrangular mesh with the spring method.

Figure 7. Examples of leaves surface geometry as obtained from the estimation process described in the paragraph.

Given a leaf mask, as obtained from the RGB images, first, it was projected on the depth map, as shown in the lower panels of Figure 6 and, further, using Equation (2), the points of the point cloud were obtained [20,21]. Note that the projection of a depth map point (x, y, d) into space coordinates (X, Y, Z) allows recovering the colors as well. To maximally exploit the segmentation masks for each image, we assigned a different color to each mask, so that when two masks were superimposed it was easy to distinguish to which mask leaf points belong. More precisely, the color masks were projected into space together with the points on the plants surface height (the depth map) and, because each mask was assigned a different color, we could use them as an indicator function:

$$P = (X, Y, Z) \in Leaf_j \equiv \begin{cases} 1 \text{ if } (R, G, B)_j = (0, 1, 0) \\ 0 \text{ otherwise} \end{cases} \tag{6}$$

Note that, to exemplify, we assumed that the mask for leaf j is green.

Since we used more than a depth image, thereby generating more than one point cloud, for the leaves occluded on the RGB image there are two possibilities. Either the occlusion remains the same in space or it can be resolved; however, the mask does not extend on the parts which have been disoccluded in space. In this last case, to recover the occluded part of the leaf not distinguishable by the lack of the segmentation mask, we resorted to the computation of normals using k-nearest neighbor and fitting a plane to close points. As is well known, for three points on the plane, the normal is simply obtained as the cross product of the vectors linking them, further normalized to a unit vector. Given points $(P_1, \dots, P_n)^\top$ on the boundary b of a mask, the boundary is moved according to the normals to close by points P'_j. Normals are assigned a cost according to the distance to the points on the boundary and according to the cosine angle between the normal on the boundary and normal beyond it:

$$\begin{aligned} w_{\mathbf{n}}(P, P') &= (\kappa D(\mathbf{n}_P, \mathbf{n}_{P'}) + (1 - \kappa)(\mathbf{n}_P \cdot \mathbf{n}_{P'})) \\ w^\star_{P'} &= \arg\min_{w_{\mathbf{n}}} \{ w_{\mathbf{n}}(P, P') \mid P \in b, P' \notin b \} \end{aligned} \tag{7}$$

The above essentially assigns a cost $w^\star_{P'}$ to each point P' beyond the boundary of a leaf mask, by choosing only those points whose cost of the normal is less than a threshold obtained empirically, hence it allows to extend, when this is possible, the regions of occluded leaves, without significant risks.

Having obtained the segmented leaves in space, and recalling that the space is suitably scaled so that distances correspond to real world distances, from the transformation of pixels into mm, we could measure the leaf area, even though in many cases the leaf was not full, because of occlusions, as noted above.

To obtain the leaf area, we built both a triangulation and a quadrangular mesh. The triangulation, despite being more precise, is more costly and the triangles do not have the same shape and size, therefore area computation requires looking at each triangle. The triangulation can be appreciated Figure 7a.

To fit the quadrangular mesh to the leaf points, we used the spring method (see [22]) solving the fitting as a non-linear least square problem, with relaxation.

The results of the fitting for the quadrangular meshes are given in Figure 7b.

Finally, we checked if a mesh is occupied by one or more points, and if it is not, then the specific mesh was removed, but only if it is on the boundary. We can see in Figure 7 that the result is sufficiently accurate, for measuring area, clearly not for graphical representations, and it is quite fast. The only difficulty is the choice of the size of the quadrangles, which determines the smoothness of the shape and, obviously, the goodnesses of fit, yet augmenting the computation cost.

Hence, the leaf area is simply: $\sum_{i=1}^{N} A_i$, where each A_i is determined by the choice of the size of the quadrangular meshes.

The results on the accuracy of the computation are given in Section 5. To obtain the leaf weight, we simply used the above density formula, and obtained the weight multiplying the recovered area with the density ρ_{BL}. For the weight, we show the accuracy in Section 5.

5. Results

In this section, we present the results of our experiments. First, we recall how the experiments were conducted at the farm, and then we consider the specific features predicted for automatic phenotyping, namely, plant height, plant leaf area and leaf weight. Finally, we shortly address feature discriminant analysis for the high and low density of plants.

5.1. Experiments Method

The experiments analyzed the growth behavior of three species of basil, namely Ya, Yb and Yc, and their vital cycle confined within 28 Days After Sowing (DAS).

Each experiment was performed on two racks, as shown in Figure 2, one for high density and the other for low density plants distribution. High density implies 2574 plants per m^2 (144 × 2 plugs per layer) and low density implies 1872 plants per m^2 (104 × 2 plugs per layer). Each rack consisted of three layers, one for each variety of Basil, were the racks are numbered from 7 to 12. Racks 7–9 are for low plants density and Racks 10–12 are for high plants density.

Each layer was equipped with a camera, and set of sensors including: water and air temperature, humidity, CO_2, PH and EC. The data from the sensors were collected with a Data Acquisition System (DAQ), which controls if there are any significant variations with the environmental conditions.

The LED lights used were 28 W Valoya A673L, with the following spectrum characteristics: 14% blue, 16% green, 53% red and 17% far red. The LED lights were positioned such to provide the necessary light intensity varying between 200 and 240 μ mol/m^2.

The overall conditions throughout the experiments are shown in Table 1. The nutritions in the tank were controlled using a PH and EC meter and adjusted manually.

Each experiment considered three phenotype features: plant height, leaf area and leaf weight. The objective was to provide accurate measures of these features, given variability of only a single condition, in particular here the single variable condition is plant density. All other factors affecting the features, such as light, water, nutrients, substrate, humidity and ventilation are assumed to be the same for all plants, as shown in Table 1.

An experiment consisted of two parallel monitoring techniques: manual monitoring, with measures obtained by systematically harvesting group of plants from each tray, and vision based monitoring, with measures automatically derived for all DAS.

Table 1. Environmental Conditions

Racks	**Low Density (Rack Layers: 7,8,9)**	**High Density (Rack Layers: 10,11,12)**
Density (plugs/m ^{2})	624	849
Plugs per layer	104 × 2	144 × 2
Plants/Plug	3	3
Light cycle (hours/day)	16 and 20	16 and 20
Humidity (%)	65–70	65–70
Temperature (°C)	16 and 20	16 and 20
Light intensity (μ mol/m ^{2})	200–240	200–240
PH	5.5–6.0	5.5–6.0

The measures taken manually were: (1) plant height; (2) leaf weight, for the whole set of leaves of each plug; (3) number of leaves per plant; and (4) weight of each plug. The height was measured as is (i.e., the plant was not manually elongated to measure the full height). For the remaining measurements at DAS 21 (gray), DAS 23 (blue), DAS 25 (yellow) and DAS 28 (green) (see Figure 2 right), destructive testing was performed by cutting different sections of the layer.

In total, 2592 manual measurements for the mentioned phenotype features were recorded for three experimental cycles.

Visual monitoring started at DAS 7, when plants were moved on to the racks, and continued up to DAS 28, and was based on four images taken each day for each rack with a commercial RGBD camera, as described in Section 4.1. In total, 1728 images were collected for the three experimental cycles.

5.2. Height Prediction

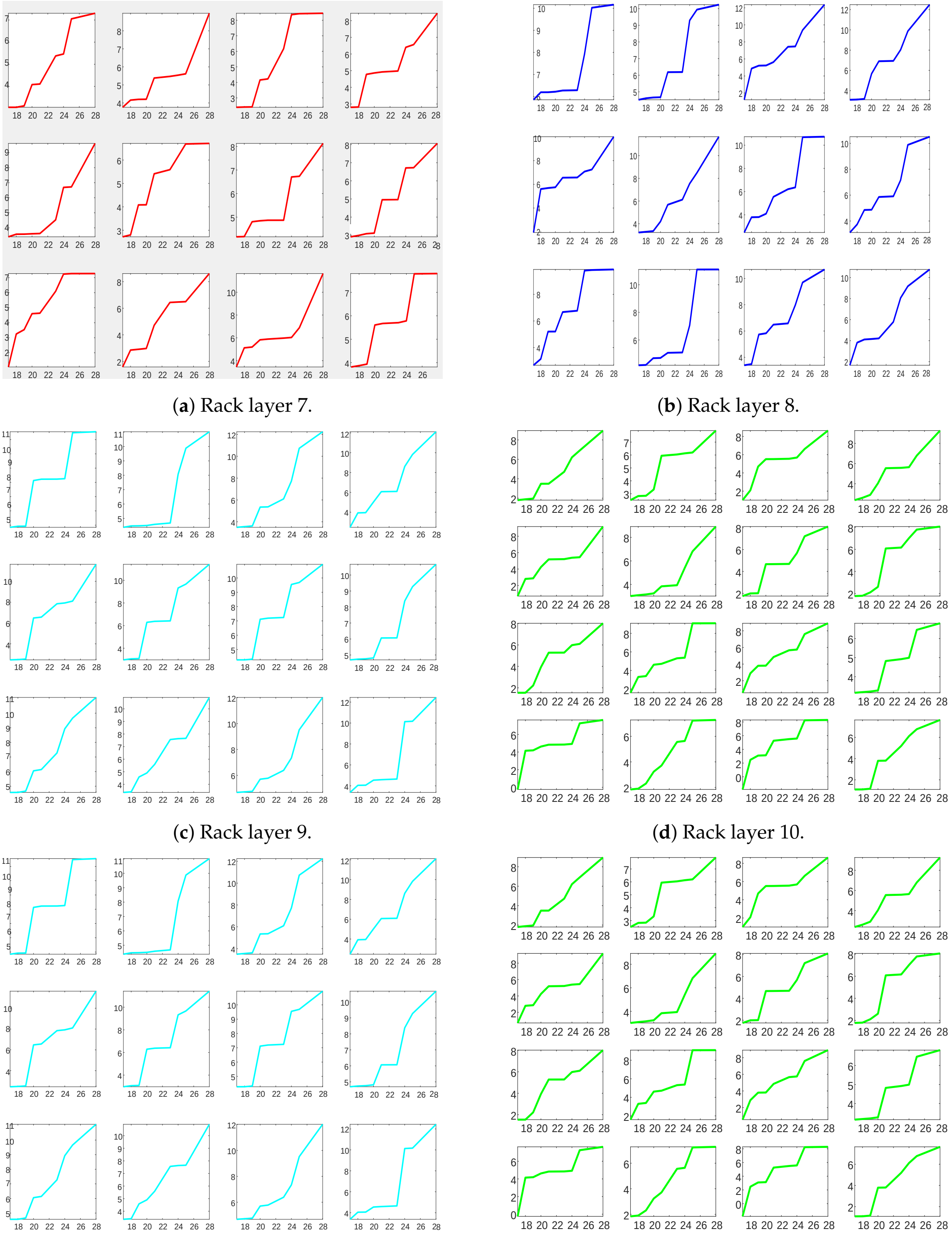
Height was measured starting at DAS 7 and height vision monitoring started at the same time. As described in Section 4.2, height was measured at each point of the depth map, with points within a cell collected together to have a more precise monitoring of plants growth in a specific region.

Figure 8 shows the height measured as described in Section 4.2. For visibility purposes, cells are grouped into a neighborhood of four cells, and we excluded the first row for the low density and the two extreme columns for the high density racks.

Note that for these precise vision based measurements of the height, for all the 21 DAS (7–28), we do not have the same manual measurements. Manual measurement of the whole rack would have required to harvesting all the plants.

Comparison with manual measurements are given in Figure 9. Here, to assess the error between the predicted and measured plants height for each rack, we considered the mean and variance of the retrieved values, and computed the Bhattacharyya distance. Note, in fact, that, while we have the highest value of about 40 plugs manually measured for DAS 18, 21, 24, 25, and 28 and rack, we have about (i.e., excluding the zeroes on the fitted cells separation lines) 720 × 1280 measured points from the depth maps, therefore a distance between probabilities is more convenient and correct. Given σ_p and μ_p are the standard deviation and mean of the predicted values p, for each DAS and rack, and similarly given that σ_m and μ_m are the standard deviation and mean of the manually measured values m, for each rack and DAS, the distance between the two height measures, at each DAS and rack, is defined as:

$$D_B(p,m) = \frac{1}{4}\log\left(\frac{1}{4}\left[\frac{\sigma_p^2}{\sigma_m^2} + \frac{\sigma_m^2}{\sigma_p^2} + 2\right]\right) + \frac{1}{4}\left(\frac{(\mu_p - \mu_m)^2}{(\sigma_p^2 + \sigma_m^2)}\right) \tag{8}$$



(**a**) Rack layer 7.

(**b**) Rack layer 8.

(**c**) Rack layer 9.

(**d**) Rack layer 10.

(**e**) Rack layer 11.

(**f**) Rack layer 12.

Figure 8. Plants height trends from DAS 17 to DAS 28 for groups of four cells in each rack layer, where cells were computed as described in Section 4.

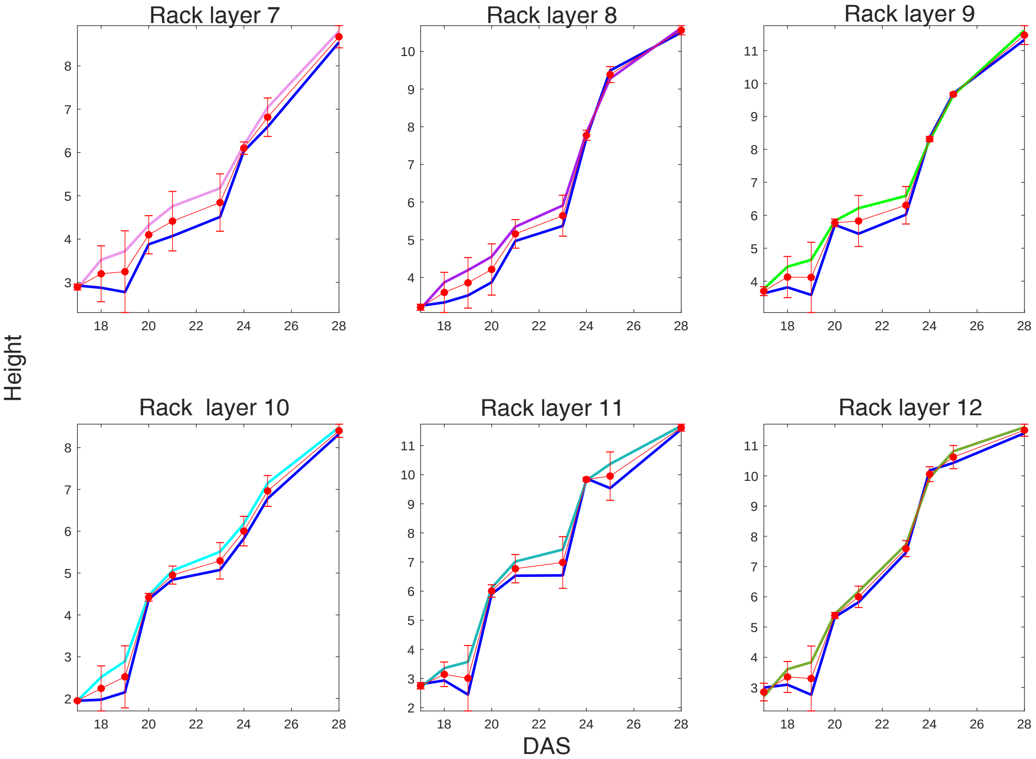


Figure 9. Plant height trends compared with ground truth, from DAS 17 to DAS 28 for Racks 7–12. Recall that Racks 7–9 have low density, while Racks 10–12 have high density plants distribution. For each graph, blue represents the ground truth height measurement taken manually, and the other colors the height predicted according to the vision based methods. Finally, the graph line in red, with red markers, and orange errors segments, shows the distance (see Equation (8)) between the ground truth and the predicted height.

Clearly, D_B is an error distance between the predicted and the measured values.

There are interesting observations to make. First, note that plants height reaches higher values where plants are more dense, and the standard deviation is slightly inferior, because of competitiveness amid plants. This is somewhat expected as increased density means that plants will compete for the photons from the LEDs. In Figure 10, assuming height is normally distributed, we provide a comparison between the probability density functions (pdf) generated by the two measures.

From the above results, as expected, it turns out that height measured from the depth maps is extremely accurate. The slight difference is due to the greater amount of data on the whole surface, collected from the vision based method, as opposed to the manual method.

5.3. Leaf Area and Weight Prediction

As already highlighted in Section 4.3, leaf area was not measured by the agronomists and only the weight was measured. However, several samples of basil leaves from the three species were taken and both weight and area for these samples were measured. Furthermore, note that weight was given for the set of all leaves for a plug, and the number of collected leaves was also given.

To properly confront the predicted leaf area with something derived by the true measurements, we proceeded as follows. First, instead of considering the density $\rho_{BL} = \frac{W_{BL}}{Area_{BL}}$ (see Section 4.3), we computed a nonlinear regression using the few samples of basil leaves about which we had both the weight and the leaf area.

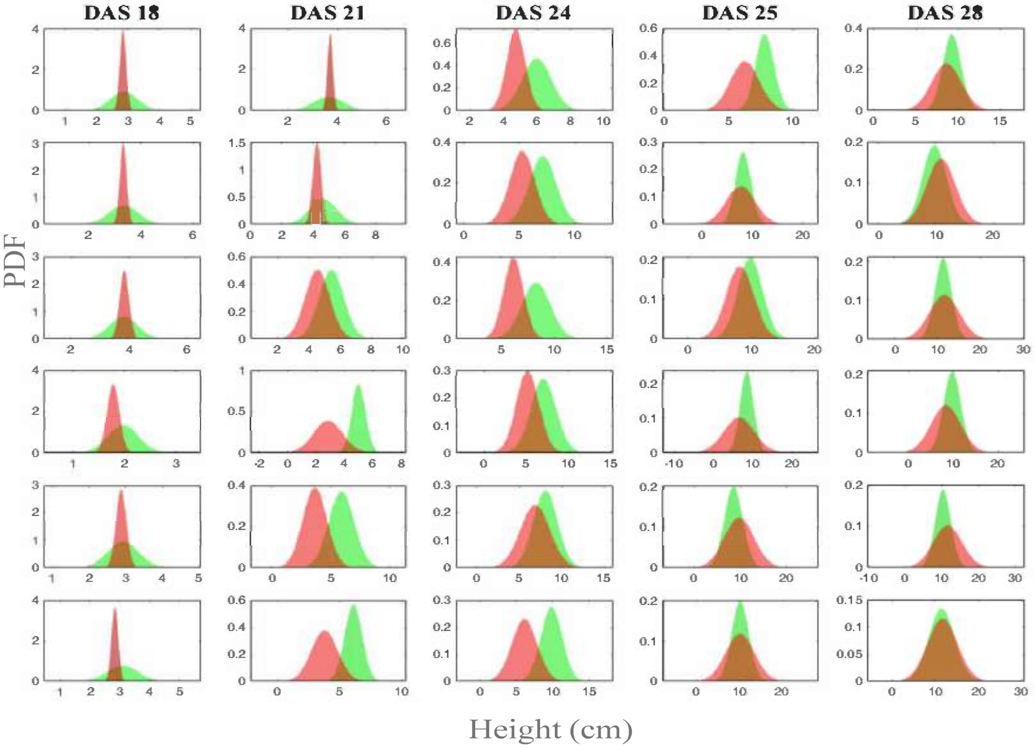


Figure 10. Plants height distribution for DAS 18, 21, 24, 25 and 28, and for Racks 7–12. Recall that Racks 7–9 are low density, while Racks 10–12 are high density. In green is the probability density function (pdf) generated by the mean and variance of the manually collected height measures, and in red is the pdf generated by the predicted mean and variance. Note that the pdf does not need to be up to 1. We preferred to give the pdf because it better highlights the increase of the variance as DAS increases, and it shows that it is also wider for low plant density racks.

We resorted to a fully connected network with two layers and obtained the parameters, so that for batches of values of the same size as the tiny training set we obtained a prediction of a leaf area, given its weight. Note that the accuracy of the net, to predict leaf area given leaf weight, is 100% on the training set, while for example with a linear regression we obtained 20% accuracy on the training set.

Once the relation between single leaf weight and single leaf area was assessed, via the fully connected network, we used the total leaf weight computed for each of the harvested plugs, at specific DAS, and the number of leaves for that plug, to obtain a distribution of leaf area per rack at DAS 21, 24, 25 and 28. These values were used for comparison with the vision based methods.

More precisely, let *net* be the fully connected network function predicting the area given a weight w of a single leaf, and let N_ℓ be the number of leaves for the specific plug, then the leaf area predicted by the weights is (see Figure 11):

$$A_W = net(\frac{1}{N_{leaves}} W_{leaves}) \tag{9}$$

where W_{leaves} is the weight of all leaves collected for a plug and N_{leaves} is the number of leaves collected for that plug. In particular, as repeated several times, the manually collected data are available only after harvesting the plants.

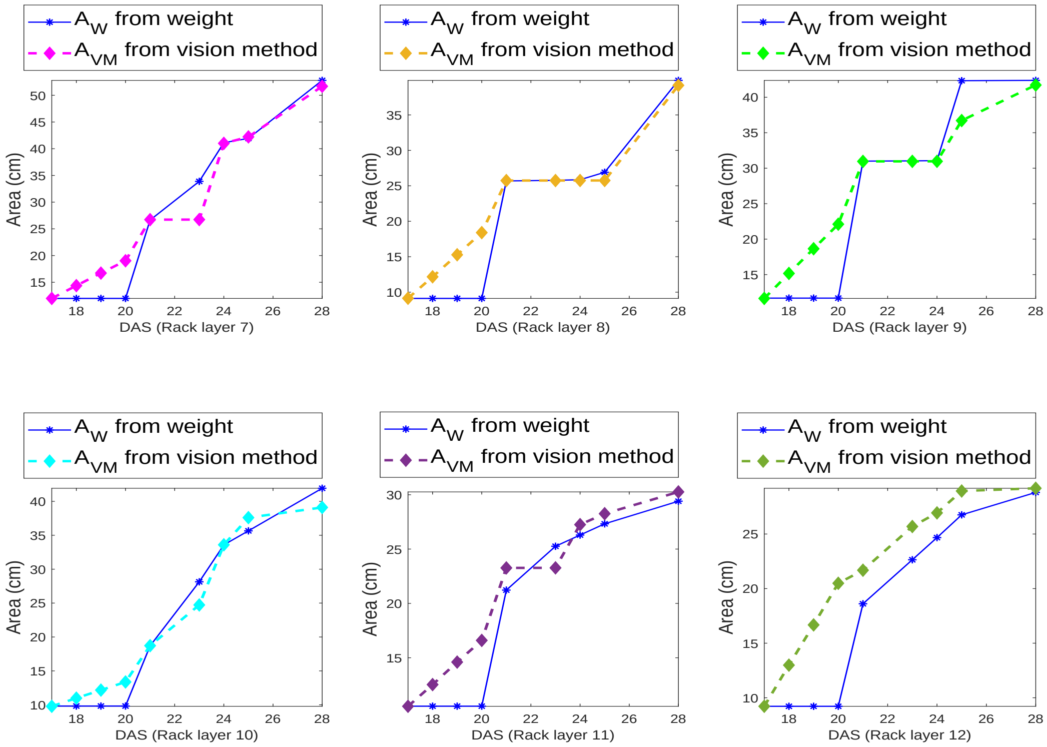


Figure 11. Leaf area trends for both the area computed by training a small set of samples, and by vision method, as described in Section 4.3. Data from vision are given for DAS 18–28, while the leaf area computed by the network, using the manually measured weights is only for DAS 21, 24, 25 and 28, for Racks 7–12.

The leaf area obtained by the vision method is defined as A_{VM}, and the method is described in Section 4.3, and is available as an average measure (based on the segmented leaves for image), for all DAS.

A comparison between the two computed leaf areas is given in Figure 11. We can note from the figure that, where there are values for the weights (at DAS 21, 24, 25, and 28), the difference between predicted area from vision and predicted area from weights is minimal or even not existent, while there is a large difference where there are no reported values for the manually collected weights, and weights are simply fitted.

This implies that either both the net and the vision based area inference provide the same wrong measurements or that both are quite accurate.

To verify the accuracy of leaf area prediction, we computed the leaf weights from the predicted leaf area from vision, by inverting the network , namely we used the network output to compute back the weights. Of course, this is subject to noise and prone to errors. The results on the recomputed weights are given in Figure 12. We can see that again the error is quite limited where the predicted weight (from vision based predicted area) are compared with manually collected leaf weights, at DAS points where the values were measured, while it is relatively high where they are simply fitted in the lack of provided values.

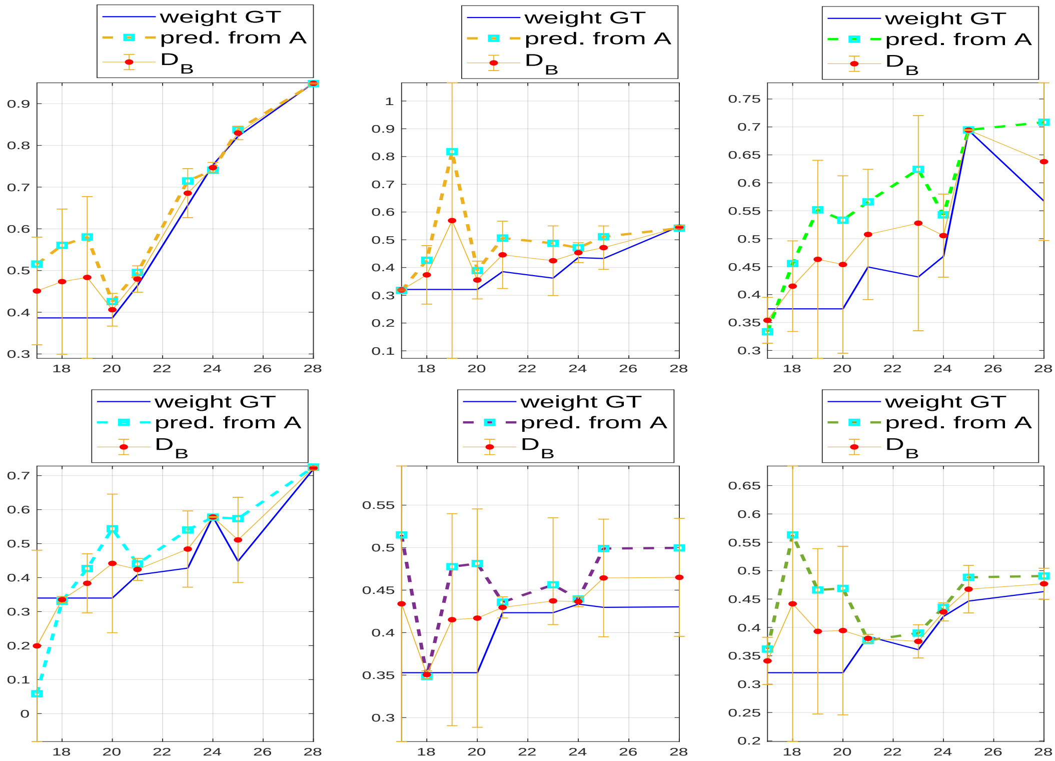


Figure 12. Leaf weight trends both computed from the predicted leaf area and as given by the manual measurements. As for plants height the error is computed basing on the Bhattacharyya distance between probability functions (see Equation (8)), for the same reasons, namely we have comparable probabilities, although not comparable samples. In fact, data from vision are given for DAS 18–28, while the area computed by the network, using the weights the leaf area, are limited to DAS 21, 24, 25 and 28, for Racks 7–12. We recall that Racks 7–9 (**top**) are low density plant distribution, while Racks 10–12 (**bottom**) are high density plants distribution.

The fitting of unknown values was based on polynomial fitting.

Note that for the weights we also computed the error using the Bhattacharyya distance as for plants height, for the same reason given for the height, namely there are no comparable samples, although there are comparable probabilities from these samples.

5.4. Discriminant Feature Analysis for High and Low Plants Densities

A final observation on the reported results concerns the features chosen. Plant height, leaf area and leaf weight are typical features for plant phenotyping. A typical question is: Do these features discriminate between high and low plant densities?

We showed that the plant height distribution, namely the mean and variance, actually highlight the difference, because the variance is slightly inferior for high density. Here, we show in Figure 13 that also the area combined with the weight discriminates between high and low plant density, while for weight this is less evident using combined features. It is clear, however, that the low density leaves have higher weight, from the histogram shown in the last panel of Figure 13.

5.5. Implementation

All algorithms were implemented in Python 3.6, Matlab 2019a, and Tensorflow. We used a i9CPU with 2.9 GHz, 128 GB of RAM and 4 NVIDIA Titan V. For the point clouds, we used PCL, CloudeCompare and MeshLab.

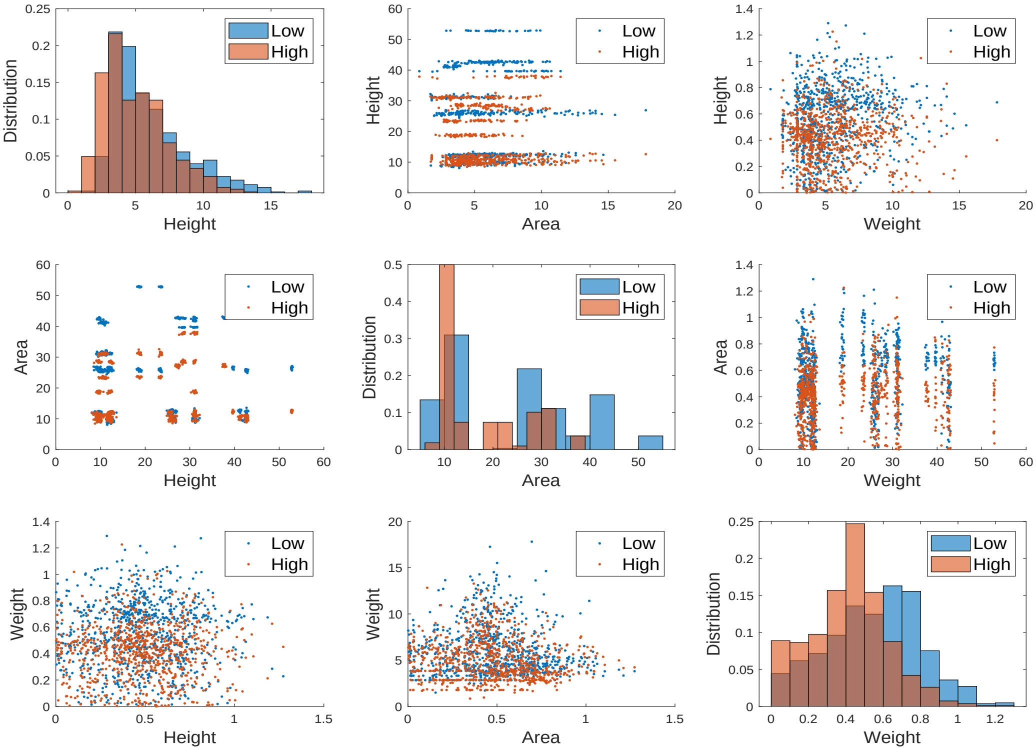


Figure 13. Features *plants height*, *leaf area* and *leaf weight* data as a pairwise scatter plot. The diagonal plots the marginal histograms of the three features. The off diagonals contain scatterplots of all possible pairs of features. As indicated in the legend, brown points are related to feature of plants gathered in low density racks, while the azure points are related to features of plants gathered in high density plants.

6. Discussion and Conclusions

This work proposes an automatic method for extracting phenotype features, based on CV, 3D modeling and deep learning. From the extracted features, height, weight and leaf area were predicted and validated with ground truths obtained manually.

This study is particularly significant because it exploited many data collected with extreme precision and due diligence. Manually collected data made it possible to evaluate the accuracy of the vision methods.

This, thus far, seems to be a unique kind of experiment in phenotyping, being based only on RGBD images. The results show that the plant height, leaf area and weight obtained using inexpensive RGBD cameras matched closely with the detailed measurements.

These results are especially relevant in indoor VFAL conditions, where plants are stacked vertically and in difficult to reach conditions.

The ability to obtain detailed information on the plant weight and therefore yield, without employing destructive techniques, facilitates the process of automation of the growth of vegetables in indoor VFAL conditions and consequently can substantially diminish the costs of production. In addition, by gathering a large amount of image based information on the plant growth and phenotype, it is possible to train the validated algorithm to provide information which aids the optimization of plant growth and therefore the reduction of resource use.

More importantly, this study shows how visual based monitoring methods, which are gaining increased attention in outdoor, "traditional" agriculture can be adapted and successfully used for indoor VFAL. In fact, the authors believe that the ability to closely control the VFAL growing environment and input conditions makes visual based methods especially useful for monitoring and optimizing the growth of fruits and vegetables in indoor VFAL.

This study opens the way for future work on the ability to obtain quantitative information on the plant phenotype and eventually determine qualitative information such as nutrition values and taste using vision based techniques in a closed and closely monitored VFAL environment. Specifically, it is possible to understand the effect of LED light (e.g., intensity, spectrum, and cycle), environmental conditions (e.g., CO_2, humidity, temperature, and PH) and grow conditions (e.g., plant density, seed type, nutrients, etc.) on crop growth behavior. The two main objectives were: (1) to maximize growth and plant quality while minimizing the resource use; and (2) determine nutritional other qualitative values (e.g., taste) and the presence of external agents on the crops using inexpensive and visual based methods rather than expensive and time-consuming non-destructive techniques.

Author Contributions: B.F. contributed to the whole view of the experiment, including organization and main writing. V.N. contributed to the 3D setting and modeling. P.G. contributed to the management. T.H. and L.B. contributed to the agronomy experiments. F.P. is the usual professor who does most of the work but stays behind just because she does not need recognition.

Funding: This research was partially funded by @Sunspring, Agricola Moderna.

Acknowledgments: We acknowledge Niccolo Caccioti for curating the data acquisition from the cameras; Duccio Piovani for being available in discussion and clarification on the data collected; and the researchers of ALCOR Lab, in particular Edoardo Alati and Lorenzo Mauro for their work on the implementation of Mask R-CNN in Tensorflow, which is publicly available, and training and testing. Figure 1 contains modified graphical elements originally designed by macrovector/Freepik.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

VFAL	Vertical farms with artificial lighting
DAS	Days at Sowing
Mask R-CNN	Mask Region based Convolutional Neural Network
ICP	Iterative Closest Point algorithm
ML	Machine Learning
CV	Computer Vision

References

1. Chena, B.; Han, M.; Peng, K.; Zhoud, S.; Shaoe, L.; Wu, X.; Wei, W.; Liuh, S.; Lia, Z.; Lid, J.; et al. Global land-water nexus: Agricultural land and freshwater use embodied in worldwide supply chains. *Sci. Total Environ.* **2018**, *613-614*, 931–943. [CrossRef] [PubMed]
2. Orsini, F.; Kahane, R.; Nono-Womdim, R.; Gianquinto, G. Urban agriculture in the developing world: A review. *Agron. Sustain. Dev.* **2013**, *33*, 695–720. [CrossRef]
3. Niu, G.; Masabni, J. Plant production in controlled environments. *Horticulturae* **2018**, *4*, 28. [CrossRef]
4. Kalantari, F.; Tahir, O.M.; Joni, R.A.; Fatemi, E. Opportunities and challenges in sustainability of vertical farming: A review. *J. Landsc. Ecol.* **2018**, *11*, 35–60. [CrossRef]
5. Graamans, L.; Baeza, E.; Van Den Dobbelsteen, A.; Tsafaras, I.; Stanghellini, C. Plant factories versus greenhouses: Comparison of resource use efficiency. *Agric. Syst.* **2018**, *160*, 31–43. [CrossRef]
6. De Wit, M.; Londo, M.; Faaij, A. Productivity developments in European agriculture: Relations to and opportunities for biomass production. *Renew. Sustain. Energy Rev.* **2011**, *15*, 2397–2412. [CrossRef]
7. Liakos, K.G.; Busato, P.; Moshou, D.; Pearson, S.; Bochtis, D. Machine learning in agriculture: A review. *Sensors* **2018**, *18*, 2674. [CrossRef] [PubMed]
8. Zhou, X.; Zheng, H.; Xu, X.; He, J.; Ge, X.; Yao, X.; Cheng, T.; Zhu, Y.; Cao, W.; Tian, Y. Predicting grain yield in rice using multi-temporal vegetation indices from UAV-based multispectral and digital imagery. *ISPRS J. Photogramm. Remote. Sens.* **2017**, *130*, 246–255. [CrossRef]
9. Monzon, J.; Calviño, P.; Sadras, V.; Zubiaurre, J.; Andrade, F. Precision agriculture based on crop physiological principles improves whole-farm yield and profit: A case study. *Eur. J. Agron.* **2018**, *99*, 62–71. [CrossRef]

10. Ramos, P.; Prieto, F.; Montoya, E.; Oliveros, C. Automatic fruit count on coffee branches using computer vision. *Comput. Electron. Agric.* **2017**, *137*, 9–22. [CrossRef]
11. Jay, S.; Baret, F.; Dutartre, D.; Malatesta, G.; Héno, S.; Comar, A.; Weiss, M.; Maupas, F. Exploiting the centimeter resolution of UAV multispectral imagery to improve remote-sensing estimates of canopy structure and biochemistry in sugar beet crops. *Remote. Sens. Environ.* **2019**, *231*, 110898. [CrossRef]
12. Ballesteros, R.; Ortega, J.; Hernandez, D. Onion biomass monitoring using UAV-based RGB imaging. *Precis. Agric.* **2018**, *19*, 840–857. [CrossRef]
13. Story, D.; Kacira, M. Design and implementation of a computer vision-guided greenhouse crop diagnostics system. *Mach. Vis. Appl.* **2015**, *26*, 495–506. [CrossRef]
14. Vit, A.; Shani, G. Comparing RGB-D Sensors for Close Range Outdoor Agricultural Phenotyping. *Sensors* **2018**, *18*, 4413. [CrossRef] [PubMed]
15. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*; Cambridge University Press: Cambridge, UK, 2003 .
16. Besl, P.J.; McKay, N.D. Method for registration of 3-D shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*; SPIE: Bellingham, WA, USA, 1992; Volume 1611, pp. 586–606.
17. Chen, Y.; Medioni, G. Object modelling by registration of multiple range images. *Image Vis. Comput.* **1992**, *10*, 145–155. [CrossRef]
18. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
19. Barriuso, A.; Torralba, A. Notes on image annotation. *arXiv* **2012**, arXiv:1210.3448.
20. Ntouskos, V.; Sanzari, M.; Cafaro, B.; Nardi, F.; Natola, F.; Pirri, F.; Ruiz, M. Component-wise modeling of articulated objects. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 13–16 December 2015; pp. 2327–2335.
21. Natola, F.; Ntouskos, V.; Pirri, F.; Sanzari, M. Single image object modeling based on BRDF and r-surfaces learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 4414–4423.
22. Krishnamurthy, V.; Levoy, M. Fitting smooth surfaces to dense polygon meshes. In Proceedings of the SIGGRAPH, New Orleans, LA, USA, 4–9 August 1996; Volume 96, pp. 313–324.