agriculture

Article

Deep Learning Tools for the Automatic Measurement of Coverage Area of Water-Based Pesticide Surfactant Formulation on Plant Leaves

Fabio Grazioso *, Anzhelika Aleksandrovna Atsapina, Gardoon Lukman Obaeed Obaeed and Natalia Anatolievna Ivanova

Photonics and Microfluidics Laboratory, Tyumen State University, Volodarskogo 6, Tyumen 625003, Russia; a.a.acapina@utmn.ru (A.A.A.); g.obaid@utmn.ru (G.L.O.O.); n.a.ivanova@utmn.ru (N.A.I.)

* Correspondence: f.grazioso@utmn.ru

Abstract: A method to efficiently and quantitatively study the delivery of a pesticide-surfactant formulation in a water solution to plant leaves is presented. The methodology of measurement of the surface of the leaf wet area is used instead of the more problematic measurement of the contact angle. A method based on a Deep Learning model was used to automatically measure the wet area of cucumber leaves by processing the frames of video footage. We have individuated an existing Deep Learning model, called HED-UNet, reported in the literature for other applications, and we have applied it to this different task with a minor modification. The model was selected because it combines edge detection with image segmentation, which is what is needed for the task at hand. This novel application of the HED-UNet model proves effective, and opens a wide range of new applications, the one presented here being just a first example. We present the measurement technique, some details of the Deep Learning model, its training procedure and its image segmentation performance. We report the results of the wet area surface measurement as a function of the concentration of a surfactant in the pesticide solution, which helps to plan the surfactant concentration. It can be concluded that the most effective concentration is the highest in the range tested, which is 11.25 times the CMC concentration. Moreover, a validation error on the Deep Learning model, as low as 0.012 is obtained, which leads to the conclusion that the chosen Deep Learning model can be effectively used to automatically measure the wet area on leaves.

Keywords: pesticides; surfactants; wet area; deep learning; segmentation; precision agriculture

check for updates

Citation: Grazioso, F.; Atsapina, A.A.; Obaeed, G.L.O.; Ivanova, N.A. Deep Learning Tools for the Automatic Measurement of Coverage Area of Water-Based Pesticide Surfactant Formulation on Plant Leaves. *Agriculture* **2023**, *13*, 2182. https://doi.org/10.3390/agriculture13122182

Academic Editors: Panagiotis Tziachris, Christos Karydas and Miltiadis Iatrou

Received: 18 September 2023
Revised: 18 November 2023
Accepted: 20 November 2023
Published: 22 November 2023

1. Introduction

1.1. Research Backgrounds

The present work is part of the scientific effort to improve agriculture, using the latest technologies to contribute to what is commonly known as *precision agriculture* [1–4]. One of the goals in this field of research is the study and improvement of the *delivery of pesticides*, which are usually deployed as part of water-based solutions [5]. The use of pesticides improves the efficiency and productivity of agriculture and forestry through crop protection and growth stimulation. Surfactants (also referred to as adjuvants) are usually added to enhance the biological activity of pesticide formulations, which allows for overcoming the leaf protection barriers, which plants produce in the form of hydrophobic cuticular wax or trichomes and increase the effective coverage area of pesticide formulations. Surfactants reduce the surface tension of water-based pesticide formulations allowing the adhesion of the spray to the leaf surfaces, significantly increasing the wetting of the foliar surface and enhancing pesticide uptake into leaves [6–18]. Indeed, by definition, a surfactant is a chemical compound that changes (usually decreases) the surface tension between the liquid to which it is added, and the surface of a solid (or more in general any other substance, be it solid, liquid or gas) with which it comes into contact. The common structure of a surfactant molecule is elongated, with a hydrophilic chemical group on one end (which tends to connect with water molecules) and a hydrophobic chemical group on the other end, which tends to repel water molecules. When introduced into water, the molecules of surfactants start to aggregate, and if their number, i.e., their concentration, is high enough, they form closed, globular structures, called *micelles*, where all the hydrophobic ends are arranged in the center, and all the hydrophilic ends form the surface of the micelle, which is in contact with the surrounding water. The structure of the micelles is the key mechanism that enables the surfactant to change the surface tension, which in turn is the parameter that mostly influences the ability of the pesticide formulation to wet and spread to a more extended surface of the plant leaves. Because of this mechanism, an important parameter used to characterize a surfactant is the *critical micelle concentration* (CMC), i.e., the concentration at which the amount of surfactant molecules is enough for complete closed micelles to start to form. At any concentration above the CMC, the additional surfactant molecules will aggregate and form more micelles. Since CMC is so important, in the present work, a direct measurement of the CMC of the surfactant used was conducted, which is described in Section 2.2. The existing formulations and dosages for the agricultural applications of surfactants do not always give the expected results for the various types of crop leaves. The latter fact stimulates excessive use of both surfactants and pesticides, which, on the one hand, does not result in a uniform residual coverage of plant leaves after spreading and evaporation of the spray droplets [19,20], and on the other hand, leads to an increase in negative impact on the ecosystem in the form of contamination of soil and water reservoirs and deaths of pollinator insects and honeybees [21,22].

Relevance of the Present Study

Taking into account the context described above, the development of effective surfactant formulations for crop protection requires laboratory and field measurements of the *wetting area* in order to determine the *optimal ratio* between the coverage area and the surfactant (adjuvants) concentration. One usual approach found in the literature is the use of goniometry to measure the evolution of the contact angle and the diameter of the droplet contact with the leaf surface to obtain quantitative data about surfactant performance [13]. However, this method does not provide reliable information due to the non-uniform front of the droplet spreading on the surface of leaves with complex morphology. For example, on leaves with longitudinal veining common to cereal crops, the droplet will spread predominantly along the veins, while in the case of the reticulate vein pattern (cucurbit crops), the liquid will drain in the vein valleys. Moreover, the presence of trichomes does not allow the droplet contact angle to be clearly defined. Finally, the maximum coverage area of the leaves depends on which part of the leaf surface the droplet is deposited [15].

In light of these considerations, to determine the criterion of surfactant efficiency, another approach seems to be preferable: the study of the *spreading area*. For this study, the process of droplet spreading over the leaf surface is recorded from above using a camera, and then some images (frames) from the video sequence are processed to measure the coverage area. Precise measurement of the coverage area of plant leaves as a function of wetting and evaporation time after spray deposition and as a function of the concentration of chemicals and leaf morphology is strongly required to develop precise rates for the usage of plant protection products and adjuvants. Along this line of research on the phenomenon of spreading of aqueous solutions on different types of surfaces, the following literature from the same authors of the present work can be mentioned [23–27].

In order to process images for the measurement of the wet area, it is necessary to perform the *segmentation*, which consists in the process of selection of the parts of the images (pixels) that represent the wet area, differentiating them from all the other parts of the image. Once this segmentation is carried out, the wet area's surface will be measured counting the total number of pixels representing the wet area. One possible approach to perform the segmentation is the manual process, in which a person manually selects the wet area of each image. This method is probably the most accurate, but it is also the slowest. To put the time required into context, it should be considered that it is common to record thousands of images, extracting the frames from videos, which can last several minutes, expecially if we want to achieve a high time resolution. Manual processing is impractical for the processing of such a high number of frames. In order to perform the surface measurement in a high number of images (in the range of thousands), an automatic segmentation method is necessary.

1.2. Related Work

1.2.1. Comparison with Numerical Algorithms

In comparing the results reported in the literature, with respect to the automatic image segmentation, it is possible to divide the automatic methods for image segmentation into two groups. The first group consists of *numerical comparative algorithms*, which extract the numerical values of intensity, and/or saturation, luminosity, hue, etc., of each pixel and compare them; the second group consists of methods based on Deep Learning (DL), a specific implementation of *Artificial Intelligence* based on the technology of *neural networks*. In this section the numerical comparative algorithms and their *drawbacks* are discussed and compared with the approach used in the present work, which is based on Artificial Intelligence, and in particular on DL tools. Indeed, in a previous publication, other attempts from our group have already been reported with some numerical algorithms for the automatic image segmentation [27], and in that publication, a similar discussion can be found.

When the *numerical algorithms* are considered, their approach relies on the analysis of the values of each pixel and their comparison. Therefore, if they are used for the segmentation of leaf images, the first step to accomplish is the manual analysis of a certain number of images, in order to find some common numerical rule that can effectively differentiate the pixels from the wet and the dry areas of the leaves. Then, the numerical algorithms can be further grouped into two general classes: those that rely on some threshold value and those that rely on some differential computation. The algorithms in the first group compare each pixel to an intensity threshold value in order to determine to which class the pixel belongs (in our case, wet or dry surface). Regarding this category of algorithms, it can be observed that this approach is not ideal when the images come from a *general purpose camera*. Indeed, most of such cameras have a mechanism that dynamically changes the gain of their CCD (charge-coupled device, the array of miniaturized light sensors that capture optical images and turn them into digital information), depending on the amount of dark and bright areas present in the subject. The result of this feature is that the brightness and intensities of the pixels in the different frames of a video may change continuously, depending on the bigger or smaller amount of dark and clear areas present in the field of view. Moreover, even on a single image, the threshold intensity of the pixels may not be uniform due to some non-uniformity in the illumination and/or irregularity in the surface. The overall result is that the optimal threshold can change substantially from image to image, and from region to region of the same image, making the algorithms based on thresholds not very accurate for this particular application. Regarding the second class of algorithms, which relies on differences between neighbour pixels, they are much more accurate because they are able to compensate for the difference in brightness, from image to image and from region to region of the same image. However, due to the complexity of the computations required, especially if they consider not only the first neighbours of each pixel but also some longer range, the algorithms from this category are computationally intensive, need powerful computers and usually have lower efficiency. Moreover, the algorithms based on differential computation usually need some external information about the direction along which to compute the differentiation; this can be sometimes problematic, and it puts an extra burden on the side of the user and makes this category of algorithms less practical. Moreover, if we analyze the images used in this work in particular, we can notice that the wet area frequently has some bright spots that reflect the light of the illumination source, and these spots have a very bright and clear color, which is usually wrongly considered as part of the dry area. This situation can be seen, e.g., in Figure 4. More generally, we have noticed several mistakes due to the uneven surface of the leaves: the light source can never illuminate evenly the surface, creating brighter and darker spots, which confuses the numerical algorithms. A possible workaround for this type of mistake would be to consider all the pixels topologically internal to the wet area to belong to the wet area. In this way, the bright spots due to the reflection would be correctly included in the wet area, although their brightness and color values would assign them to the dry area. However, this approach is not viable, because in several cases, the spreading of the water is such that it leaves dry spots completely surrounded by the wet area. Therefore, with this "topological" rule, the numerical algorithm would wrongly consider the dry "islands" as wet. Finally, the presence of a dark background with an intensity similar to the wet area can easily confuse the numerical algorithm. In conclusion, these limits in accuracy and efficiency have motivated the authors to apply DL models to image segmentation [28,29].

The option of measuring the wetting area of pesticides and adjuvants formulations to study the efficacy of surfactants is well present in the literature [14,15,17,21,30]. This approach is reported either as an alternative or as a complementary technique with respect to the measurement of the contact angle. Regarding the algorithmic image processing, several examples can be found in the literature. A simple implementation of a numerical algorithm is demonstrated in [30], where the authors evaluate the effectiveness of pesticide formulations by spraying them on a water-sensitive paper and then scanning it. The images obtained are converted into black-and-white images using the ImageJ software and the number of pixels with an intensity corresponding to the wet area is counted. It should be noted that testing formulations on paper does not give adequate information about the behavior of formulations with surfactants interacting with the surface of plant leaves. In a number of studies, the image analysis is carried out using the polygonal hand-trace feature found in commercial software [14–18]. In the work reported in [31], the algorithm for measuring the maximum leaf coverage area is based on a multi-stage complex image processing procedure and implemented in MatLab. In the wetting experiments, a blue pigment is added to the water-based pesticide formulations. The image processing consists of the extraction from the RGB image of the blue component (related to the blue pigment used), followed by the filtration of the image, the generation of the binary image using a segmentation method, and counting the number of pixels by applying a disk mask together with the procedure for the noise elimination. In general, it is possible to find several works in the literature on the application of Deep Learning in agriculture. Useful reviews are found, e.g., in [32,33].

1.2.2. Summary of Existing Studies

In summary, after the analysis of the existing studies and reporting existing approaches for image segmentation, we can observe that numerical methods have a limitation in accuracy, in particular when the illumination and imaging conditions are not ideal, and this represents a clear shortcoming of the existing studies. On the other hand, a thorough search of the relevant literature yielded no article with DL detection of wet surfaces of leaves. This, therefore, can be considered as the entry point of the present study, which intends to fill this gap.

1.3. Original Contribution

The present research aims at filling the gap found in agriculture research, where the use of Deep Learning tools is still rather limited. Here, the question if the approach with DL can be effectively applied to the research of precision agriculture is addressed, and in the conclusion, a positive answer to this question will be given. In particular, we have identified the problem of studying the optimal concentration of surfactants. This can be considered an example of the type of research that can be carried out with this new tool.

In finding this optimal concentration, two parameters can be taken into account: the spreading area, and the lifetime of wet formulation on the leaf surface. On the one hand, for the optimal application of pesticides to the plant leaf, it is desirable to have a wetting area as big as possible. But on the other hand, a big spreading area leads to a faster drying of the formulation. Although the formulation will stay on the leaf after it dries, the ability to penetrate the leaf tissue and be absorbed by the plant is mostly limited to the time when it is still in the liquid phase. Therefore, the study of the optimal concentration is not trivial, and can not be limited to the measurement of the maximal spreading area. Indeed, from this consideration, it becomes clear that this study needs to take into account the dynamic of the spreading process, i.e., its evolution over time, following the expansion, plateau and decrease in the area. In turn, this motivates the collection and analysis of as many images as possible of the drop of the formulation as it spreads and evaporates on the leaf so as to have a good enough plot of the phenomenon with enough datapoints and time resolution. Only an automatic area measurement can provide enough datapoints in a reasonable time, and this further motivates the use of Deep Learning.

2. Measurements

2.1. Materials

For the measurements, cucumber leaves were used. The fluid used for the wetting experiments was a solution of distilled water, with 5% concentration by volume of a colloidal suspension of silver (the suspension itself has a concentration of 3 g/L). The colloidal silver suspension is a commercial product acting as fungicide and bactericide, purchased from the company AgroKimProm (Saint Petersburg, Russia), and commercialized under the name ‘Zeroxxe’. Then, an organic, silicon-based super-wetting agent, in particular the commercial product ‘Majestik’, also purchased from AgroKimProm (Russian Federation), was added to this solution in different concentrations. The choice of the surfactant Majestik was due to its wide use in agriculture in the class of organo-silicone adjuvants. As reported in the literature [6], this class of surfactants has strong power in lowering the surface tension, and therefore studying them is of high interest. The choice of Zeroxxe was due to its properties as a pesticide and fungicide. It was important to conduct the present research with a formulation as similar as possible to the actual formulations used in agriculture; therefore, it was decided to avoid using just a solution of water and surfactant, which may miss components and characteristics that may be important for the results of interest. Zeroxxe is a colloidal suspension with a complex formulation of rather big particles that influence the spreading process, interacting in particular with the spreading front. The silver particles precipitate on the leaf surface, and this is visible to the naked eye, especially in the last part of the process, soon before and after the complete evaporation of the formulation. The choice of cucumber was motivated by several factors. Firstly, the morphology of the cucumber leaves is particularly convenient for the present research. The macroscopic geometry of cucumber leaves is rather corrugated, which helps to study the spreading of water and water-based formulations. Another characteristic is the presence of trichomes on the surface of the leaves, which play a similar role in giving complexity to the surface and to the spreading process. Moreover, the average size of the cucumber leaves is also big enough to let the drops spread to their full extent without reaching the edges of the leaf, so as to avoid ’boundary effects’ that may interfere with the measurement. Finally, the cucumber is a very widespread plant variety, used in household garden cultivation as well as for commercial production, and this creates interest in studying it. The seeds are relatively fast to germinate, and at the time the present research was conducted, it was appropriate to grow this plant, which was conveniently available in the laboratory.

2.2. CMC Measurement

In order to have precise information on the surfactant used in the formulation, a direct measurement of its CMC was performed, using the tensiometer model *DCAT 15* from the company DataPhysics Instruments (Filderstadt, Germany), based on the Wilhelmy plate method. A tensiometer is a device to measure the surface tension of a liquid. In particular, those that use the Wilhelmy plate method have a sensitive dynamometer (an instrument to measure the force), to which we connected a metallic plate. The plate was immersed in the liquid under measurement, contained in a cup below the dynamometer. A motorized system lowered the plate into the liquid, and then lifted it slowly until the plate was completely extracted from the liquid. This procedure was repeated and at each repetition, some amount of an additive was added to the liquid using a motorized automatic system. In this way, at each repetition, the surface tension was measured for different (increasing) values of the concentration of the additive to the liquid. The instrument with all the automatic systems was driven by a computer, and the dynamometer readings were collected in a file together with the concentration values. In Figure 1, we report a plot of such datapoints in a semi-logarithmic plot, where on the vertical axis of the plot, the values for the surface tension (SFT) are reported, and on the horizontal axis, the values for the volume concentration are reported. This plot, together with the linear best fits, leads to the estimate of the CMC. According to the theory of CMC, the surface tension is supposed to follow an exponential decay as the concentration of surfactant grows until it reaches the critical micelle concentration (CMC), and then it is supposed to continue linearly. So, the data were plotted in a semi-logarithmic plane in order for the exponential part to appear linear. In the figure, the datapoints are represented in two different colors to highlight those belonging to the two different regimes (exponential in red, and linear in green). Then, two linear best fits are performed for the initial and the final regime (see the blue lines), and the CMC is estimated as the concentration at the interception of the two lines (see the black circle in the plot). The estimate for the CMC is 80 ± 5 µL/L. The concentrations in the plot, and this final result, are all expressed in µL/L, with the volume of the surfactant, as it is found in the vendor-sealed container, being measured per liter of water. We do not know the concentration of the surfactant, expressed in mass per volume, as it is in the vendor's container; however, in our measurement of the CMC, we have expressed the concentrations in the same way (volume of vendor's concentration per volume of water). This allows us to express the concentrations as a fraction of the CMC, and this will also be used later in Figure 10.

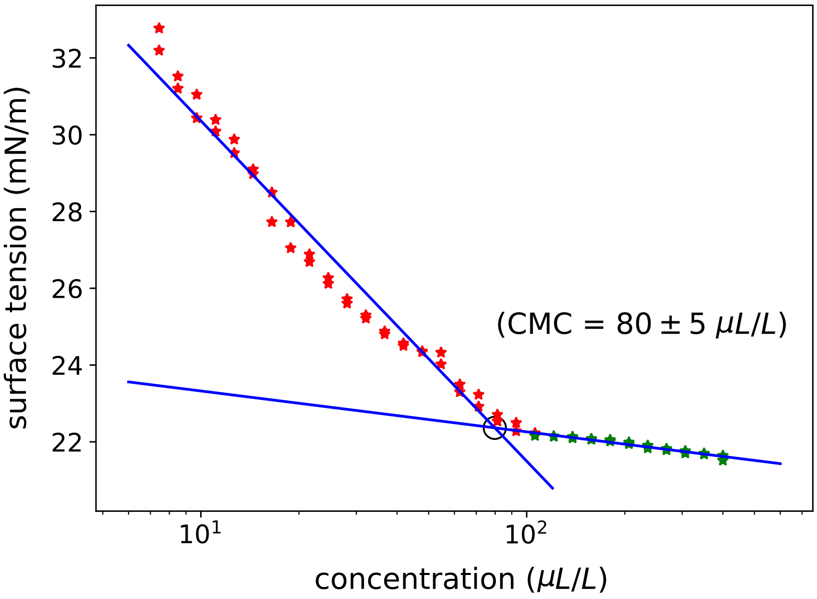

Figure 1. Plot of the data for CMC measurement. On vertical axis the surface tension (SFT), on horizontal axis the concentration. The experimental datapoints are represented as stars. Two linear best fits follow the two parts of the dataset. The intersection of the two lines measures the CMC ("critical micelle concentration").

3. Methods

3.1. Image Processing

As described in the introduction, the wet area measurement was performed using imaging methods: a drop of the water-based solution was deposited on a leaf using a manual adjustable pipette, with the volume set to 10 µL; its spreading process was video-recorded and analyzed. Several samples of a solution of colloidal silver in a fixed concentration, and with different concentrations of silicon-based surfactant, were prepared. Then, a drop of each surfactant concentration was deployed on a clean, dry cucumber leaf in horizontal position, and the spread was recorded. The images were obtained with a Zeiss AXIO Zoom.V16 microscope model (Jena, Germany) equipped with a PlanApo Z 0.5x/0.125 FWD 114 objective operated at 3.5 (i.e., minimal) optical magnification, and the digital video was acquired with a Zeiss Axiocam 506 color camera integrated into the optical system, with a frame rate of 30 frames per second. Custom-made software tools were developed and written in the Python programming language (version 3.9) [34] using the OpenCV library (version 4.6) [35] to perform the processing of the video recordings. The first step was the extraction of the frames from the video files; then, the Deep Learning (DL) model adapted from the HED-UNet model [36] was implemented using the PyTorch DL framework (version 1.11) [37], again using the Python language, to process the single frames (still images) from the video, in particular for the segmentation, assigning each pixel to one out of two categories: wet and not wet. After the segmentation, the number of pixels of the wet area of each image was computed, and the wet area was computed using the appropriate conversion coefficient, taking into account the optical magnification. The data of the timestamp of each frame and the area estimate were recorded in a data file for each value of the surfactant concentration. The data were further elaborated to study the effect of the surfactant on the spreading of the water-based pesticide solution.

3.2. HED-UNet Model

The present work is based on the HED-UNet DL model [36].

To apply DL to the segmentation needed for the present work, it was decided to apply to this task a DL model already existing, but which was designed for automatic processing (segmentation) of satellite imagery for geographic and meteorological applications. So, the contribution of the present work in this respect is the idea to apply an existing model to a task completely different from the one for which it was originally developed. In this section, some of the main features and characteristics of the model are reported. The details of the Deep Learning model described in this section are not essential for the comprehension of the work reported in this article, and can be skipped without missing any crucial information. On the other hand, the description contained here is only general, and the reader interested in the full description of the model is referred to the original publication cited above. The strength of this DL model, which is also the reason why it was chosen, is that it combines the capabilities and the advantages of two different, previous models—the holistically nested edge detection (HED) model [38] and the UNet model [39]. Those two older models are designed for two different tasks, namely *edge detection* (HED) and *semantic segmentation* (UNet). Although those are two different and separate tasks, on a second look, they have a lot in common. Indeed, in essence, segmenting an image means separating different areas of the image, i.e., grouping the pixels of the different areas into different classes. It is not difficult to show that this is tightly linked to the task of finding the pixels on the boundaries between said areas. To highlight the strong link between the two tasks, it is worth noting that both have the same type of output: a “mask” of the same size as the input image, where each element specifies the class of the corresponding pixel in the image. In the HED-UNet model, the classification was designed using only two classes: the class of pixels representing the water and the class of pixels representing “all the rest”, i.e., the dry areas of leaves and possibly the background, if present. The main components of the HED-UNet model architecture are summarized in Figure 2.

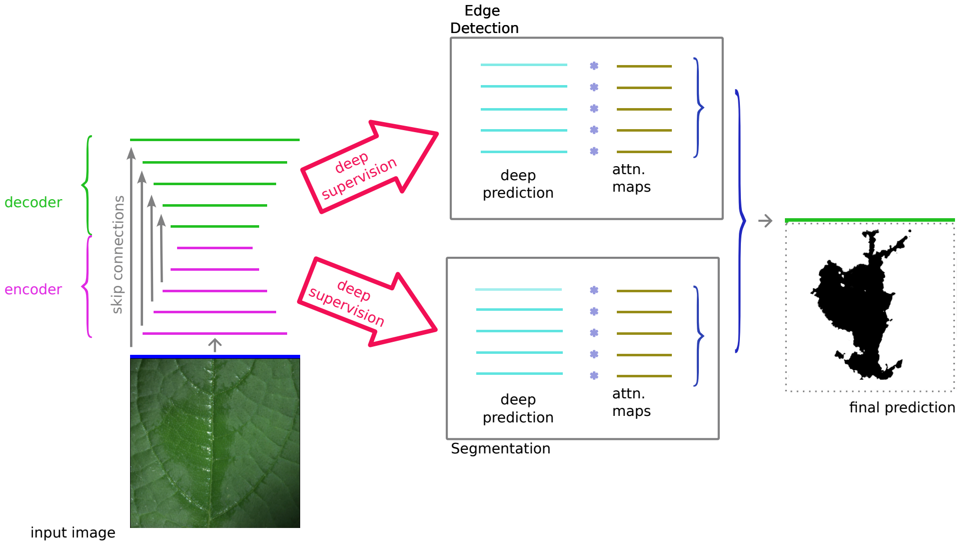

Figure 2. Schematic of the structure of the HED-UNet model (based on the schemes present in [36]. In the lower left corner, a depiction of a possible input image is shown. Only this image and the final prediction mask are shown in this schematic in their extension: all the other images, feature masks, and intermediate predictions are represented just as lines, as if the images were seen from an edge. On the right, the final prediction of the wet area (black pixels) is presented at the output of the model.

The authors of [36] have applied this intuition to a practical image-processing problem: finding the shoreline in satellite pictures of the Antarctica landmass, obtaining good results: the "combined model" has the two HED and UNet parts in it, which both generate a prediction map which are then combined for a final result. Moreover, the HED-UNet model (as the HED and the U-net models separately) has an "encoder–decoder" structure, where a series of downsampling steps (encoder) generate maps at lower and lower resolutions, allowing for the aggregation of contextual information, and then a series of upsampling steps generate a series of maps which redistribute this information at a higher resolution, until the original resolution of the input image is restored. The encoder–decoder structure makes it possible to exploit both the localized information gathered in the high-resolution layers and the global, nonlocal information obtained from the low-resolution layers, as is extensively discussed in [40]. The HED-UNet model was used in the present work almost unchanged. The contribution of the authors was to modify the model, which was created to elaborate synthetic aperture radar (SAR) images, which are grey-scale images with only one image layer, and adapt it so as to make it able to process optical color images with three image layers for the three red, green and blue (RGB) color components.

3.3. Loss Function

As for the previous section, this section discusses some technical detail of the implementation of Deep Learning. In particular, this section describes how the performance of the DL model is measured. Indeed, a key element for any Deep Learning model is the use of an *error function*, also called *loss function*, which quantitatively measures the quality of the predictions made by the model. In a *supervised* DL model, where the *correct output* (also called *ground truth*) is provided for a set of inputs, this can be carried out computing some mathematical distance function, which would compute a single value that express some average difference between the ground truth and the output provided by the model once it has been trained. This computation is usually called the *validation* of the model. One of the most simple error functions that can be used for this purpose is the mean absolute error (MAE):

$$\mathrm{MAE} = \frac{\sum_{i=1}^{N} |y_i - x_i|}{N} \tag{1}$$

where x_i are the output value, y_i are the ground truth value, and N is the total number of values used for the MAE computation. Intuitively, this definition can be seen as the average distance between all the outputs and their corresponding correct outputs. For most of the advanced DL models, this loss function is too simple, and some more sophisticated loss functions are used, which were designed to compensate for some specific needs of each model.

To make the validation (quality assessment of the model) meaningful, this can not be computed on the data that were used for the training.

So, the usual practice when training supervised DL models is to divide the set of annotated inputs into two subsets, called the *training set* and the *validation set*: the training set is used to actually train the model, and the validation set is used to assess the quality of the model once it has been trained.

In both training and validation, the loss function (i.e., error function) can be the same, although it is used for different purposes.

In the training process, the loss function is the main tool driving the whole process. This process consists of finding the values for the weights $\vec{W}$, which characterize the internal state of the DL model, such that the set of all the prediction values $\left\{\hat{x}_i^{(n)}\right\}_i$ are as close as possible to all the ground truth values $\left\{y_i^{(n)}\right\}_i$, for all the pixels in all the images, where the index i runs on all the pixels of an image and the index n runs on all the images in the training set [41,42]. So, the training process consists of the minimization of the *loss function* that quantitatively measures the average distance between (all the) sets of predictions and (all the) sets of ground truths. On the other hand, in the validation process, the loss function is used to assess the quality of the training. The two processes are usually applied in sequence: a training session (called *epoch*) using the whole training set is performed once, optimizing the weights and neuron parameters of the model on each element of the training set, and then validation is performed once on the validation set, using the frozen values of the weights and neuron parameters. Those steps (epochs) are repeated in sequence until a certain desired value for the validation is reached, which provides for the desired global quality for the performance of the model. In Figure 3, the value of the loss function outcome is reported for both the training and the validation process, as a function of the epoch number. In DL, the tasks of image segmentation and edge detection can be described as tasks of classification (assigning the input to the correct class, choosing from a list of possible classes) applied to each pixel of an image. In edge detection, the model is trained against two classes: the class of edge pixels and the class of non-edge pixels. However, those two classes are usually highly unbalanced: the number of pixels in the first class is much lower than in the second class. So, during the training process, the impact on the overall training process, from one type of pixel to the other, is also very unbalanced, and this leads to much less effective training. To alleviate this problem, it is possible to design loss functions that compensate for this unbalance. In [38], a *class-balanced* version of the binary cross entropy loss function is used. Since the application in the present work also uses two classes in its segmentation ("water" and "not water" pixels), following [36], the same loss function is used, which is described here in some detail. In this discussion we have used the following notation:

$$\vec{X}_n = \left\{x_i^{(n)}\right\}_{i=1}^{|\vec{X}_n|} \quad \text{and} \quad \vec{Y}_n = \left\{y_i^{(n)}\right\}_{i=1}^{|\vec{Y}_n|} \tag{2}$$

to denote the vector of all the pixel values from the n-th image used in the training set (which has a total number of elements equal to $\left|\vec{X}_n\right|$), and the vector of the values of the corresponding ground truth (which has a total number of elements equal to $\left|\vec{Y}_n\right|$), respectively. The values for the ground truth "pixels" in our case are only two—$\forall i, y_i^{(n)} \in \{0,1\}$—because we have two classes ("water" and "all the rest"). Moreover, the set of all the weights and other parameters of the whole neural network of the model is denoted with $\vec{W}$. Finally, the set of predictions values for the n-th input image, i.e., the $\left|\vec{X}_n\right|$ output values of the neural network, is denoted with $\left\{\hat{p}_i^{(n)}(x_i,\vec{W})\right\}_{i=1}^{|\vec{X}_n|}$, where $\left|\vec{X}_n\right|$ is the total number of pixels in the image. This prediction function $\hat{p}^{(n)}$ is a value $\hat{p}\in[0,1]$, and can be interpreted as the 'probability' of the correct output being equal to 0 or being equal to 1: $\hat{p}(y_i=0|\vec{X},\vec{W})$ is the probability of the correct i-th 'output pixel' being 0, given the values of (all) the input pixels $\vec{X}$, and (all) the weights of the network $\vec{W}$ (for simplicity, we have omitted the index n identifying the image), and similarly for $\hat{p}(y_i=1|\vec{X},\vec{W})$. In this special case where the output can be only 1 or 0, we have $\hat{p}(y_i=0|\vec{X},\vec{W})=[1-\hat{p}(y_i=1|\vec{X},\vec{W})]$; therefore, we can use only one value, which we will denote briefly as $\hat{p}_i^{(n)}$. One initial option for such a loss function in the case where there are only two possible classes is the *Binary Cross Entropy* function:

$$\mathcal{L}_{bce}=\sum_n\sum_i\left[y_i\cdot\log(\hat{p}_i^{(n)})\right]+\left[(1-y_i)\cdot\log(1-\hat{p}_i^{(n)})\right]. \tag{3}$$

In the case where the number of pixels in the two classes is highly unbalanced (water pixels much fewer than non-water pixels), it is possible to compensate, and re-balance it as follows:

$$\tilde{\mathcal{L}}_{bbce}=-\left[\beta\cdot\sum_{n,i\in Y_1}\log[\hat{p}(y_i=1)]\right]-\left[(1-\beta)\cdot\sum_{n,i\in Y_0}\log[\hat{p}(y_i=0)]\right] \tag{4}$$

where the first sum is computed only on the pixels with a ground truth = and the second sum only on those with a ground truth = 0, and where

$$\beta=\frac{|Y_0|}{|Y|} \tag{5}$$

is the class-balancing weight that allows the automatic re-balancing of the two classes, where $Y=Y_1\cup Y_0$; in essence, β is the ratio between the number of pixels in the first class (water) with respect to the total number of pixels. The base for the loss function used in this project for the training was Binary Cross Entropy with Logits Loss from the PyTorch library (version 1.11) (`torch.nn.BCEWithLogitsLoss()`). As reported in Figure 3, the value of the loss function after 320 epochs is 0.004 for the training loss and 0.013 for the validation loss.

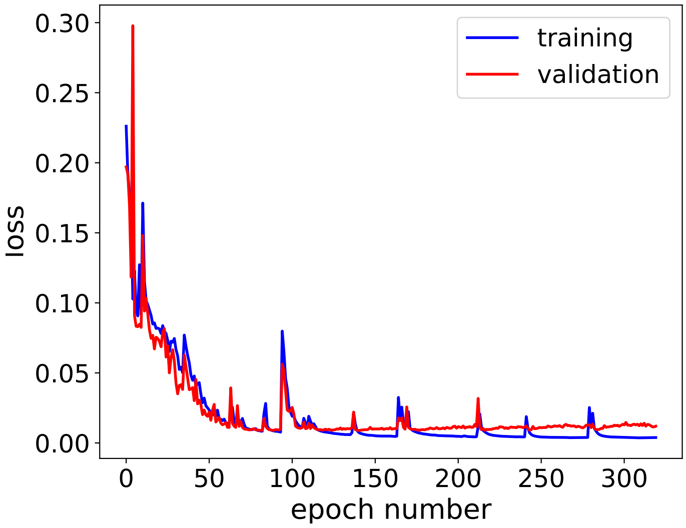

Figure 3. Plots of the training loss and the validation loss as functions of the number of epochs. We can observe a stable loss of around 0.005 for the training loss and of around 0.01 for the validation loss, computed with the `torch.nn.BCEWithLogitsLoss()` loss function explicitly reported in Equation (4).

3.4. Training

We have performed the training of the model in our facilities. In total, 130 images were manually annotated, selecting images as diverse as possible from different videos of different leaves so as to diversify the training and have better predictive performance of the trained model. In total, 80% of those images were used as training set, and 20% of them were used as validation set. For the manual annotation, image editing software (GIMP (version 2.10), free/libre and open source software) and a graphics tablet (Artist 24 Pro, XP-Pen, Japan-China) were used. The images were selected uniformly from each different concentration. For each image in the training set, a superimposing black-and-white mask was created (ground truth), indicating the wet leaf surface pixels and the pixels representing the dry leaf surface or background. Then, some *data augmentation* was used: each image and corresponding mask were rotated in 4 possible angles and mirrored horizontally and vertically, so that for each frame, 6 different training images were generated for a total of $130 \times 6 = 780$ images and masks. The training of the model was performed on a remote server, equipped with one NVIDIA Tesla "V100s" graphics card with 32 Gb of video RAM. The training of 780 images for 320 epochs took around 2 days of computing.

In Figure 3, the plot of the training loss and the validation loss as functions of the epochs are shown. At the end of the training, we have a stable loss of around 0.01 for the validation and around 0.004 for the training.

Once trained, the network is able to perform the segmentation of a single image in around one second.

Image Resizing

In the encoder–decoder module, the HED-UNet model performs several *resizing* (downsampling/upsampling) executions on the input image, halving the number of pixels (for both width and height) for each downsampling and doubling it back for each upsampling. This creates a requirement for the dimension of the input images: they need to have a number of pixels (both in height and width) that can be divided by two several times with an integer result. The root of this requirement is the fact that the number of pixels can only be an integer. Therefore, if a module would halve an odd dimension, it would need to round the result to the next integer; in the up-conversion steps, when the dimensions are doubled and brought back to the initial dimensions, this would result in a mismatch in the number of pixels.

To solve this problem, a pre-processing step is needed, resizing the images and the ground truths to a suitable dimension before they are fed into the neural network. A square dimension, with a side of 1024 pixels was used, where 1024 is a power of 2, so that the division by two always gives an even result.

4. Results and Discussion

4.1. Automatic Image Segmentation

The most important result of this work is the successful use of a DL model for automatic image segmentation of the wet area of plant leaves after comparing it with the other possible approaches to the same task.

The other approaches considered are *manual segmentation*, using a raster graphics editing software such as GIMP, and *algorithmic segmentation*, i.e., the use of numerical algorithms that directly compare the values of the pixels, using either some threshold on some of the pixels values, such as the RGB color components, their hue, saturation and brightness (HSL) values, or some differential rule that computes the gradients of those values while going from a pixel to its neighbours.

The comparison with the manual segmentation is straightforward. Although the manual process is sometimes more accurate, the time to process a single image, especially those with an extended wet surface with rather intricate and fragmented edges, can take up to 40 min, especially if a good level of accuracy needs to be reached. The average processing time can be even longer if we take into account the fact that the process is rather physically and mentally tiring, and it requires some extra time for rest. In comparison, the processing of a single image with the Deep Learning model takes a time in the order of seconds, with an accuracy that we show in Figures 4–7. The accuracy is not as good as the one obtained with the manual segmentation, but it is acceptable, and considering the time needed for the manual segmentation, it is completely impractical when the number of images that need to be processed is in the range of thousands. As discussed in Section 3.3, the data on the loss function reported in Figure 3 and in Table 1 represents a quantitative evaluation of the accuracy of the DL segmentation, computed as a comparison with the manual segmentation. In particular, in the last row of Table 1, we can see how the cumulative error computed on 20% of the total 780 manually annotated images, i.e., the 156 randomly chosen images that constitute the *validation set*, using the model with the configuration obtained after all the 320 training epochs, is equal to 0.0126. The comparison with the algorithmic segmentation is less obvious. The comparison with respect to the processing time gives no significant advantage to either of the two, although the algorithms based on differential computations tend to be slightly slower and more accurate, especially on non-uniformly illuminated images, with respect to the algorithms based on a threshold. However, the accuracy of the DL segmentation is significantly higher. Specifically, the DL segmentation is particularly efficient in distinguishing the "*islands*" of dry surfaces completely surrounded by wet areas, whereas the algorithmic segmentation is usually unable to distinguish them. Similarly, the DL segmentation is efficient in not interpreting the dark *background* as a wet area, whereas the algorithm usually misinterprets it as a wet surface because it is dark and it is difficult to process using some threshold values. In Figure 4, an example of an image with some bright reflection spots is shown, which was correctly identified as wet by the DL model, while in Figure 5, we show another example of an image with the DL prediction with an isolated dry "island" and the presence of dark background, both correctly identified by the DL model.

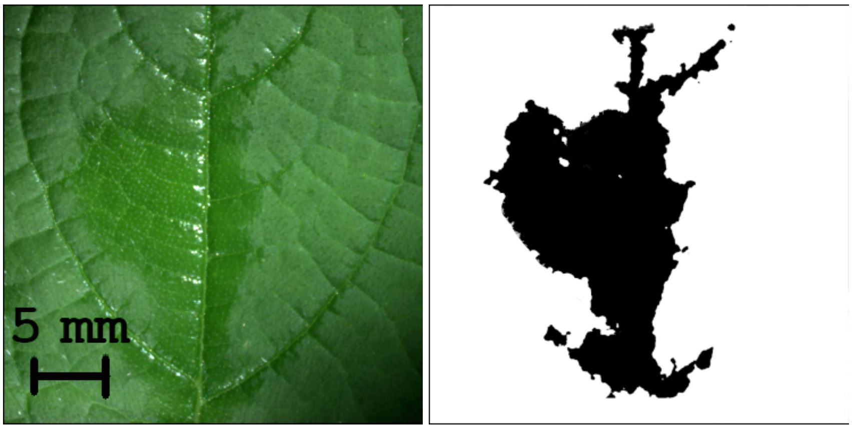

Figure 4. Example of a single frame from a video recording, on the **left**, and the corresponding output of the Deep Learning model segmentation, on the **right**. In this example, in the center, due to the geometry of leaf, a series of several bright reflection spots are present, aligned in a vertical row, along a big part of the image. On the **right**, we can see that the Deep Learning model has correctly identified the reflection bright spots as belonging to the wet area.

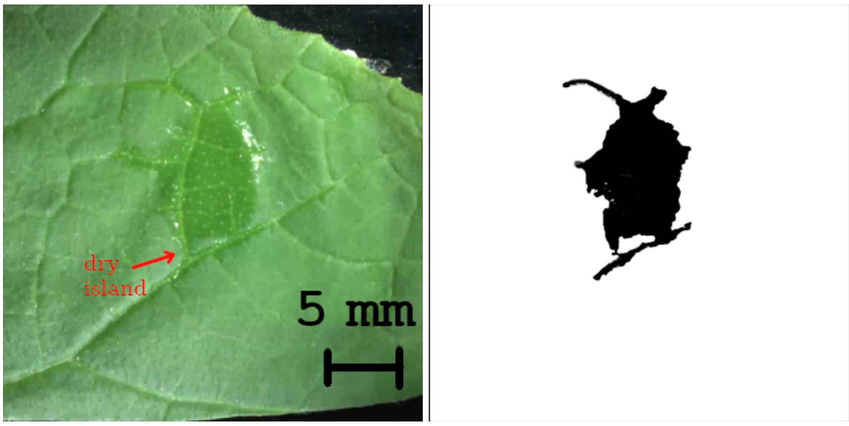

Figure 5. Another example of a single frame from one of the video footage, and the correspondent mask recognized by the Deep Learning model. In this example, we can see how the Deep Learning model was able to recognize as “dry” a spot (dry island) of the dry leaf surface, all encircled by wet surface, at its bottom (pointed by the red arrow). In this image, we can also see how the DL model can correctly neglect the black background present in the image.

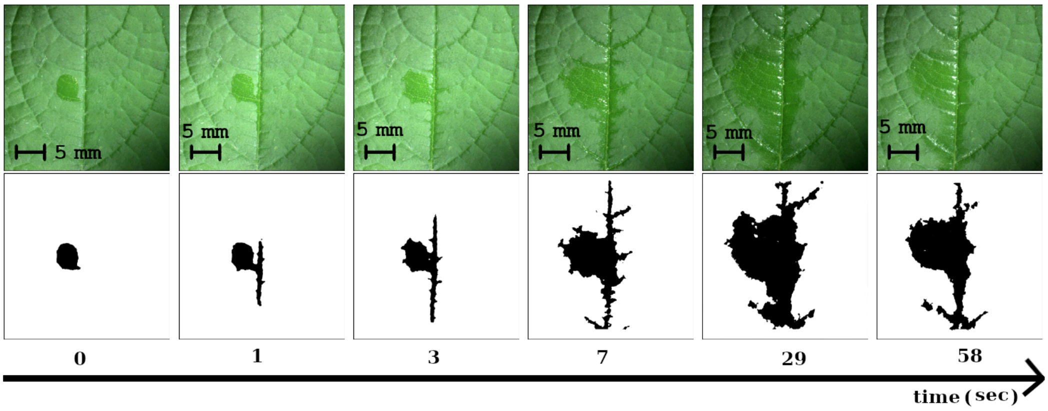

Figure 6. Here, we can follow the evolution over time of the wet area on a leaf, at time t = 0, 1, 3, 7, 29 and 58 s, respectively. From **left** to **right**, we can observe the initial small and round drop, then the bigger area with more fragmented edges, and how the area of the last two images reduces after reaching the maximum extension due to evaporation. For each image in the **top** row, we show the output of the prediction performed by the Deep Learning model.

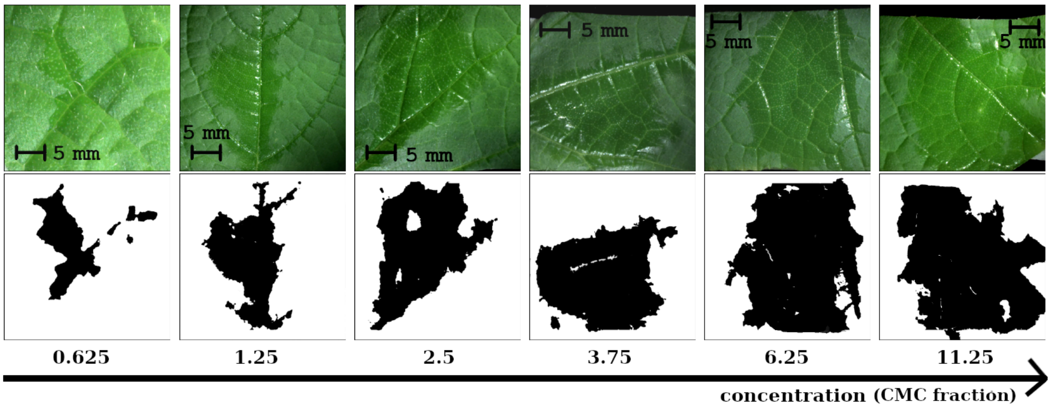

Figure 7. Here, we can observe the different *maximum area extent* as the concentration increases. In particular, from **left** to **right**, the pictures refer to the concentrations of 0.625, 1.25, 2.5, 3.75, 6.25 and 11.25 expressed as fractions of the CMC.

Table 1. Table of the training and validation errors, for intermediate epochs, and for the last epoch.

Epoch	Training	Validation
6	0.0961	0.0813
70	0.0178	0.0126
110	0.0178	0.0136
220	0.0052	0.0104
320	0.00392	0.0126

We have observed that the segmentation mistakes of the DL model that we have trained are almost always "false negative" mistakes, and almost never "false positives". In other words, the pixels representing the dry surface (and possibly the background) are almost always classified as such, whereas the pixels representing the wet surface are sometimes classified as dry. This means that the wet area measurements based on the DL segmentation are on average underestimated. This was observed inspecting single images and the corresponding prediction masks (see the examples in Figures 4–7), and also confirmed by the comparison between the few manual measurements and the DL measurements shown in Figure 8. Indeed, in these two plots, referred to as two concentrations, we can observe how the datapoints representing the manual measurements are positioned at the upper edge of the ribbon of datapoints representing the automatic measurements.

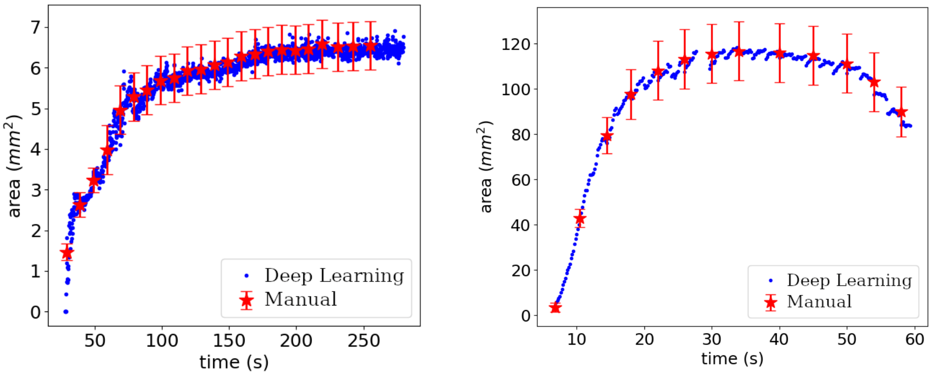

Figure 8. Examples of the progress of wet area over time. On the **left**, we have the data referring to the lowest concentration used in the experiment, 50 µL/L, which is equal to 62% of the CMC. On the **right**, we have the data referring to the highest concentration used, 900 µL/L, which is equal to 11.2 times the CMC. Blue dots represent data measured with the Deep Learning model, and the red stars represent data measured manually.

4.2. Surfactant Concentration and Wetting

The other result of this work is the study of the evolution of the wet area on cucumber leaves, its maximal spread and its lifetime, and how they depend on the concentration of the surfactant used. This has been chosen as an example of the application of DL to precision agriculture. We have explored the range of concentrations starting from about half the CMC up to around 11 times the CMC. In Figure 9, we compare, in one plot, the behavior of the wet area spreading over time for different concentrations. In this and the following figure, the concentrations are expressed as fractions of the CMC, which were measured as 80 ± 5 µL/L (see Section 2.2 and Figure 1). We have observed that for all the concentrations except the lowest, the area reaches a maximum expansion value, stays on that value for a while, and then decreases due to evaporation. A very small and almost constant value of the area for the lowest concentration was observed, which is small compared to the other curves. No reduction in the area in the final part is observed because the solution with a low concentration is so hydrophobic that its layer remains thick above the leaf surface and therefore does not evaporate much, at least within the time duration of this measurement. In Figure 8, we report the evolution of wet area over time for just two concentrations, namely the lowest (below CMC) and the highest. In particular, on left we have the data referring to the lowest concentration used in the experiment, 50 µL/L, which is equal to 62% of the CMC. In this case, the initial drop does not spread much. Therefore, we have a rather small area, and we do not observe the decrease in area although in this case, the time period observed is much longer because the water layer remains thick and evaporates less. On the right, we have the data referring to the highest concentration used, 900 µL/L, which is equal to 11.2 times the CMC. Here, we can observe a fast spread, a plateau of a relatively constant area at the maximum spread, and the area drop due to evaporation. The small (blue) dots represent the area measured with the help of Deep Learning image processing, while the red stars represent the manually measured area. The error bars of the manually measured area were estimated by repetition of the measurement, i.e., manually measuring several times the same image, with different rotations and flipping of the image. The standard deviation of the small sample of measured values was used to infer the standard deviation of the normal distribution of measurements using Student's distribution. In those plots, we also compare the area measured with the DL image segmentation, with the area measured with manual image segmentation (manual measurement).

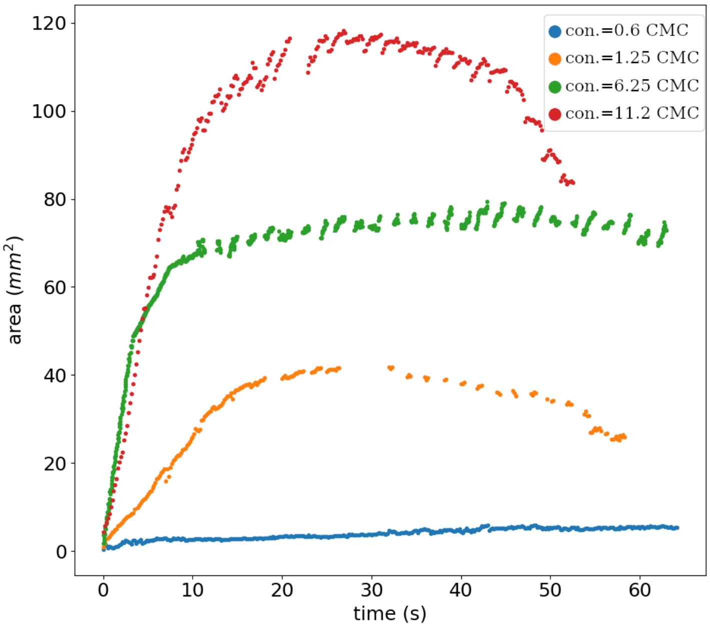

Figure 9. In this plot, we have a summary and a comparison of the behavior of the wet area spreading over time for different concentrations of the surfactant. The concentrations in the legend are expressed in fractions of the CMC.

We have made an estimate of the error affecting the manual measurement by means of performing several manual segmentations of the same image with the same frame taken from a spreading video. To make each of those different manual measurements more meaningful for the estimate of the error, each measured image was rotated by a different angle or mirrored with horizontal or vertical symmetry. We have obtained a small sample of different manual measurements of the same image, measured its standard deviation, and then used it to infer the standard deviation of the normal distribution of measurements using the Student's distribution. The DL-based measurements appear to be *consistent* with the manual measurements.

5. Conclusions

The contributions of this work are twofold. The main contribution of this work is the positive answer to the question as to whether it is possible to successfully apply Artificial Intelligence tools to precision agriculture. In particular, the use of a Deep Learning model, developed for a different purpose, was demonstrated and applied here to efficiently process image segmentation of the high number of images involved. Although the model was developed for a different application for satellite image segmentation, its characteristics that combine a sub-model designed for segmentation and a sub-model designed for edge detection make the idea to apply it to this different task of segmentation of the wet area of leaves effective. This result is represented by the very low value obtained for the loss function which is discussed in Section 3.4 and summarized here in Table 1. In particular, the values of the training and the validation error for the last epoch, of 0.00392 and 0.0126, respectively, demonstrate the good quality of the prediction of the Deep Learning model and success in the task.

This is the more important result of the work: the demonstration of a measurement tool that can be applied to several similar measurements. A limit of the present work in this respect is the relatively small number of images used for the training of the model. Although one of the specific characteristics of the model chosen is indeed the ability to give good results with relatively small training sets, it may be that this has led to some *overfitting* of the model, which makes it less effective if used with a different dataset. It should be noted, in this regard, that the goal of this work was to prove the efficacy of the method, not to develop a software tool readily suitable 'as is' for a large range of different applications, e.g., different leaves or different liquids.

A second result, considered as an example of the possible applications of DL to precision agriculture, is the estimate of the optimal concentration of an organosilicon surfactant in a water-based pesticide formulation (colloidal silver), which takes into account the maximal spreading area, and the lifetime of the wet phase on the leaf of cucumber. The results of these measurements show an optimal concentration close to the CWC of around six times the CMC. In Figure 10, we show a plot of the estimated maximum areas for each surfactant concentration. In this figure, the maximum extension reached by the wet area was estimated when it reached a plateau before it started to decrease due to evaporation. The error bars represent the range of variation in the area values that were estimated within the relatively constant plateau. In this plot, we can observe the behavior of superspreading, which happens at a special concentration called Critical Wetting Concentration (CWC). This phenomenon has been investigated in previous works by one of the authors [43,44]. Indeed, in Figure 10, it is possible to observe that the maximal area stabilizes at around 50 or 60 mm^2 for concentrations of up to five or six times the CMC, and then has a sudden increment. This gives a rough estimate of the CWC. It is also important to notice in Figure 9 that the *lifetime of the wet phase of the formulation* (the horizontal length of the curves before their sudden drop) is much longer for the lower concentrations, including the one at 6.25 CMC, and then becomes rather short for the higher concentrations. In conclusion, the CWC can be considered as the optimal concentration that provides a convenient compromise between a wide spreading area and a long lifetime of the wet phase of the formulation.

More investigation is needed to model the (non linear) function that describes the dependence of the maximum area on the surfactant concentration, and possibly to explore a wider range of concentrations, possibly reaching an asymptotic value for the maximal spreading area.

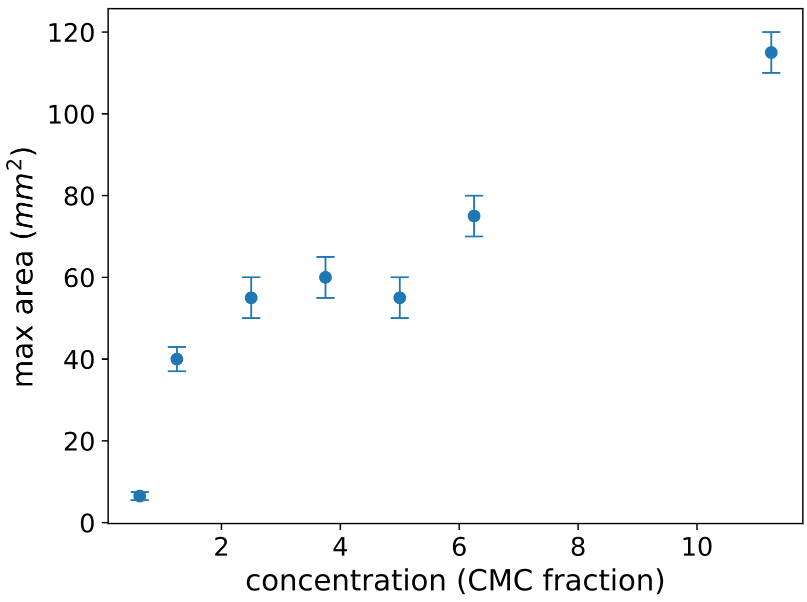

Figure 10. Summary plot of the estimated maximum areas for each surfactant concentration. The values for the concentrations are 50, 100, 200, 300, 400, 500, and 900 µL/L, whereas the CMC is 80 µL/L. So, these concentrations are 0.625, 1.25, 2.5, 3.75, 6.25 and 11.25, expressed as a fraction of the CMC.

Author Contributions: Conceptualization, N.A.I. and F.G.; methodology, N.A.I.; software, F.G.; validation, N.A.I. and F.G.; formal analysis, N.A.I. and F.G.; investigation, A.A.A. and G.L.O.O.; resources, N.A.I. and F.G.; data curation, F.G.; writing—original draft preparation, F.G.; writing—review and editing, F.G.; visualization, F.G.; supervision, N.A.I.; project administration, N.A.I.; funding acquisition, N.A.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Science and Higher Education of the Russian Federation grant number FEWZ-2023-0005.

Data Availability Statement: Data available on request due to restrictions. The data presented in this study are available on request from the corresponding authors. The data are not publicly available due to restrictions.

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

References

1. Cisternas, I.; Velásquez, I.; Caro, A.; Rodríguez, A. Systematic literature review of implementations of precision agriculture. *Comput. Electron. Agric.* **2020**, *176*, 105626. [CrossRef]
2. da Silveira, F.; Lermen, F.H.; Amaral, F.G. An overview of agriculture 4.0 development: Systematic review of descriptions, technologies, barriers, advantages, and disadvantages. *Comput. Electron. Agric.* **2021**, *189*, 106405. [CrossRef]
3. Wang, T.; Chen, B.; Zhang, Z.; Li, H.; Zhang, M. Applications of machine vision in agricultural robot navigation: A review. *Comput. Electron. Agric.* **2022**, *198*, 107085. [CrossRef]
4. Rai, N.; Zhang, Y.; Ram, B.G.; Schumacher, L.; Yellavajjala, R.K.; Bajwa, S.; Sun, X. Applications of deep learning in precision weed management: A review. *Comput. Electron. Agric.* **2023**, *206*, 107698. [CrossRef]
5. Wang, P.; Yu, W.; Ou, M.; Gong, C.; Jia, W. Monitoring of the Pesticide Droplet Deposition with a Novel Capacitance Sensor. *Sensors* **2019**, *19*, 537. [CrossRef]
6. Stevens, P.J.G. Organosilicone surfactants as adjuvants for agrochemicals. *Pestic. Sci.* **1993**, *38*, 103–122. [CrossRef]
7. Taylor, P. The wetting of leaf surfaces. *Curr. Opin. Colloid Interface Sci.* **2011**, *16*, 326–334. [CrossRef]
8. Zhang, C.; Zhao, X.; Lei, J.; Ma, Y.; Du, F. The wetting behavior of aqueous surfactant solutions on wheat (*Triticum aestivum*) leaf surfaces. *Soft Matter* **2017**, *13*, 503–513. [CrossRef]
9. Castro, M.J.L.; Ojeda, C.; Cirelli, A.F. Surfactants in Agriculture. In *Green Materials for Energy, Products and Depollution*; Lichtfouse, E., Schwarzbauer, J., Robert, D., Eds.; Springer: Dordrecht, The Netherlands, 2013; pp. 287–334. [CrossRef]
10. Wang, R.; Xu, X.; Shi, X.; Kou, J.; Song, H.; Liu, Y.; Zhang, J.; Wang, Q. Promoting Efficacy and Environmental Safety of Pesticide Synergists via Non-Ionic Gemini Surfactants with Short Fluorocarbon Chains. *Molecules* **2022**, *27*, 6753. [CrossRef]

11. Jibrin, M.O.; Liu, Q.; Jones, J.B.; Zhang, S. Surfactants in plant disease management: A brief review and case studies. *Plant Pathol.* **2021**, *70*, 495–510.
12. Liu, Z. Effects of surfactants on foliar uptake of herbicides—A complex scenario. *Colloids Surfaces B Biointerfaces* **2004**, *35*, 149–153. [CrossRef]
13. Song, Y.; Huang, Q.; Huang, G.; Liu, M.; Cao, L.; Li, F.; Zhao, P.; Cao, C. The Effects of Adjuvants on the Wetting and Deposition of Insecticide Solutions on Hydrophobic Wheat Leaves. *Agronomy* **2022**, *12*, 2148. [CrossRef]
14. Lin, H.; Zhou, H.; Xu, L.; Zhu, H.; Huang, H. Effect of surfactant concentration on the spreading properties of pesticide droplets on Eucalyptus leaves. *Biosyst. Eng.* **2016**, *143*, 42–49. [CrossRef]
15. Xu, L.; Zhu, H.; Ozkan, H.E.; Thistle, H.W. Evaporation rate and development of wetted area of water droplets with and without surfactant at different locations on waxy leaf surfaces. *Biosyst. Eng.* **2010**, *106*, 58–67. [CrossRef]
16. Pierce, S.M.; Chan, K.B.; Zhu, H. Residual Patterns of Alkyl Polyoxyethylene Surfactant Droplets after Water Evaporation. *J. Agric. Food Chem.* **2008**, *56*, 213–219. [CrossRef]
17. Wang, F.; Hu, Z.; Abarca, C.; Fefer, M.; Liu, J.; Brook, M.A.; Pelton, R. Factors influencing agricultural spray deposit structures on hydrophobic surfaces. *Colloids Surfaces A Physicochem. Eng. Asp.* **2018**, *553*, 288–294. [CrossRef]
18. Fine, J.D.; Cox-Foster, D.L.; Mullin, C.A. An Inert Pesticide Adjuvant Synergizes Viral Pathogenicity and Mortality in Honey Bee Larvae. *Sci. Rep.* **2017**, *7*, 40499. [CrossRef]
19. Ciarlo, T.J.; Mullin, C.A.; Frazier, J.L.; Schmehl, D.R. Learning Impairment in Honey Bees Caused by Agricultural Spray Adjuvants. *PLoS ONE* **2012**, *7*, e40848. [CrossRef]
20. Xu, L.; Zhu, H.; Ozkan, H.E.; Bagley, W.E.; Krause, C.R. Droplet evaporation and spread on waxy and hairy leaves associated with type and concentration of adjuvants. *Pest Manag. Sci.* **2011**, *67*, 842–851. [CrossRef]
21. Yu, Y.; Zhu, H.; Ozkan, H.E.; Derksen, R.C.; Krause, C.R. Evaporation and Deposition Coverage Area of Droplets Containing Insecticides and Spray Additives on Hydrophilic, Hydrophobic, and Crabapple Leaf Surfaces. *Trans. ASABE* **2009**, *52*, 39–49. [CrossRef]
22. Yu, Y.; Zhu, H.; Frantz, J.; Reding, M.; Chan, K.; Ozkan, H. Evaporation and coverage area of pesticide droplets on hairy and waxy leaves. *Biosyst. Eng.* **2009**, *104*, 324–334. [CrossRef]
23. Ivanova, N.; Starov, V. Wetting of low free energy surfaces by aqueous surfactant solutions. *Curr. Opin. Colloid Interface Sci.* **2011**, *16*, 285–291. [CrossRef]
24. Ivanova, N.; Zhantenova, Z.; Starov, V. Wetting dynamics of polyoxyethylene alkyl ethers and trisiloxanes in respect of polyoxyethylene chains and properties of substrates. *Colloids Surf. A Physicochem. Eng. Asp.* **2012**, *413*, 307–313. [CrossRef]
25. Ivanova, N.; Kubochkin, N.; Starov, V. Wetting of hydrophobic substrates by pure surfactants at continuously increasing humidity. *Colloids Surfaces A Physicochem. Eng. Asp.* **2017**, *519*, 71–77. [CrossRef]
26. Ivanova, N.; Esenbaev, T. Wetting and dewetting behaviour of hygroscopic liquids: Recent advancements. *Curr. Opin. Colloid Interface Sci.* **2021**, *51*, 101399. [CrossRef]
27. Grazioso, F.; Fliagin, V.M.; Ivanova, N.A. Measurement of geometrical parameters of the crude-oil/water interface propagating in microfluidic channels using deep learning tools. *Interfacial Phenom. Heat Transf.* **2022**, *10*, 57–74. [CrossRef]
28. Liakos, K.; Busato, P.; Moshou, D.; Pearson, S.; Bochtis, D. Machine Learning in Agriculture: A Review. *Sensors* **2018**, *18*, 2674. [CrossRef]
29. Garcia-Garcia, A.; Orts-Escolano, S.; Oprea, S.; Villena-Martinez, V.; Martinez-Gonzalez, P.; Garcia-Rodriguez, J. A survey on deep learning techniques for image and video semantic segmentation. *Appl. Soft Comput.* **2018**, *70*, 41–65. [CrossRef]
30. Zhu, H.; Salyani, M.; Fox, R.D. A portable scanning system for evaluation of spray deposit distribution. *Comput. Electron. Agric.* **2011**, *76*, 38–43. [CrossRef]
31. Li, H.; Travlos, I.; Qi, L.; Kanatas, P.; Wang, P. Optimization of Herbicide Use: Study on Spreading and Evaporation Characteristics of Glyphosate-Organic Silicone Mixture Droplets on Weed Leaves. *Agronomy* **2019**, *9*, 547. [CrossRef]
32. Zhu, N.; Liu, X.; Liu, Z.; Hu, K.; Wang, Y.; Tan, J.; Huang, M.; Zhu, Q.; Ji, X.; Jiang, Y.; et al. Deep learning for smart agriculture: Concepts, tools, applications, and opportunities. *Int. J. Agric. Biol. Eng.* **2018**, *11*, 32–44. [CrossRef]
33. Hammad Saleem, M.; Potgieter, J.; Mahmood Arif, K. Automation in Agriculture by Machine and Deep Learning Techniques: A Review of Recent Developments. *Precis. Agric.* **2021**, *22*, 2053–2091. [CrossRef]
34. van Rossum, G.; Drake, F.L. *Python Reference Manual*; PythonLabs: Reston, VA, USA, 2001. Available online: http://www.python.org (accessed on 17 September 2023).
35. Bradski, G. The OpenCV Library. In *Dr. Dobb's Journal of Software Tools*; M&T Pub.: Stamford, UK, 2000.
36. Heidler, K.; Mou, L.; Baumhoer, C.; Dietz, A.; Zhu, X. HED-UNet: Combined Segmentation and Edge Detection for Monitoring the Antarctic Coastline. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 4300514. [CrossRef]
37. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; Curran Associates, Inc.: Red Hook, NY, USA, 2019; pp. 8024–8035. Available online: https://pytorch.org (accessed on 17 September 2023).
38. Xie, S.; Tu, Z. Holistically-Nested edge Detection. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1395–1403.

39. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv* **2015**, arXiv:1505.04597.
40. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
41. Nielsen, M.A. *Neural Networks and Deep Learning*; Determination Press: Hoboken, NJ, USA, 2015.
42. Grazioso, F. *An Introduction to Artificial Intelligence and Deep Learning*; Mercury Learning and Information: Duxbury, MA, USA, 2022. [CrossRef]
43. Ivanova, N.A.; Starov, V.; Rubio, R.; Ritacco, H.; Hilal, N.; Johnson, D. Critical wetting concentrations of trisiloxane surfactants. *Colloids Surfaces A Physicochem. Eng. Asp.* **2010**, *354*, 143–148. [CrossRef]
44. Ivanova , N.A.; Kovalchuk, N.M.; Sobolev, N.M.; Starov, V.M. Wetting films of aqueous solutions of Silwet L-77 on a hydrophobic surface. *Soft Matter* **2016**, *12*, 26–30. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.